AUGMENTATION OF HUMAN SKILL IN MICROSURGERY

by
Marcin Arkadiusz Balicki

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland
February, 2014

Abstract

Surgeons performing highly skilled microsurgery tasks can benefit from information and manual assistance to overcome technological and physiological limitations to make surgery safer, efficient, and more successful. Vitreoretinal surgery is particularly difficult due to inherent microscale and fragility of human eye anatomy. Additionally, surgeons are challenged by physiological hand tremor, poor visualization, lack of force sensing, and significant cognitive load while executing high-risk procedures inside the eye, such as epiretinal membrane peeling. This dissertation presents the architecture and the design principles for a surgical augmentation environment which is used to develop innovative functionality to address the fundamental limitations in vitreoretinal surgery. It is an inherently information driven modular system incorporating robotics, sensors, and multimedia components. The integrated nature of the system is leveraged to create intuitive and relevant human-machine interfaces and generate a particular system behavior to provide active physical assistance and present relevant sensory information to the surgeon. These include basic manipulation assistance, audio-visual and haptic feedback, intraoperative imaging and force sensing. The resulting functionality, and the proposed architecture and design methods generalize to other microsurgical procedures. The system's performance is demonstrated and evaluated using phantoms and in vivo experiments.

ABSTRACT

Advisor:

Russell H. Taylor PhD, Professor

Department of Computer Science, The Johns Hopkins University

Readers:

Peter Kazanzides PhD, Associate Research Professor

Department of Computer Science, The Johns Hopkins University

James T. Handa MD, Professor of Ophthalmology

Wilmer Eye Institute, The Johns Hopkins University School of Medicine

Peter L. Gehlbach MD PhD, Associate Professor of Ophthalmology

Wilmer Eye Institute, The Johns Hopkins University School of Medicine

Acknowledgments

In the process of researching and writing my dissertation, I have been very grateful to work with the Microsurgery BRP (*eyeBRP*) team: an extremely competent, creative, and interdisciplinary group of students, scientists, engineers, professors, and surgeons. Without their generous contributions, this work would not have materialized; it benefits tremendously from the time and expertise of this remarkable team. Foremost, I would like to thank my advisor, Dr. Russell H. Taylor, for his steadfast support, encouragement, patience, guidance and (surprising) availability. His endless creativity, deep and broad technical knowledge, enthusiasm, motivation and dedication to research have left a significant imprint on my approach to scientific research and engineering design. In every way, he was instrumental in a wonderful PhD experience; none of the advancements would have been possible without him. Also, thanks to Dr. Taylor, now I know how to fly-cast a Woolly Bugger, and maybe even catch a few fish.

Besides benefiting from privileged access to the world class robotics facility at the LCSR, I have been fortunate to learn from top-notch professors at the university, including Dr. Peter Kazanzides, Dr. Iulian Iordachita, Dr. Greg Hager, Dr. Jin Kang, and the exceptional surgeons Dr. James Handa, and Dr. Peter Gehlbach, who all generously shared their true expertise in their fields.

ACKNOWLEDGMENTS

Many thanks to Anton Deguet and Balazs Vagvolgyi, for quickly engineering solutions to many unforeseen software and hardware problems, helping me debug numerous lines of code, and sharing their broad wisdom in software engineering and their appreciation of fine coffee. Thank you to my thesis committee members, Prof. Kazanzides, Dr. Handa, and Dr. Gehlbach for their insightful comments and suggestions.

A special thanks to many incredibly kind and gifted teachers that have educated, inspired, and guided me in engineering, science, and art, and also in life. Thank you Jean Le Mee, Adrianne Wortzel, Stan Wei, Mr. Morrow, Rich Lullo, Joe Dimartino, Paul Maye, and many others. Dziękuję bardzo to my early-childhood teachers in Poland who fostered my desire to learn and explore.

I would also like to thank my gifted student colleagues whose openness, friendship, expertise, and intelligence made my time at CISST-ERC/LCSR a memorable and very pleasant experience. They include: Xuan, Rogerio, Ali, Xingchi, Tian, Kevin, Topher, Min Yang, Berk, Raphael, Ankur, Topher, Carol, Omar, Kel, Laura, Eric, Amrita, Tamas, Tricia, Robert, and many others who contributed to the *eyeBRP* project through their participation in the extremely cordial and dynamic lab environment.

My friends from New York City have been more than encouraging in my PhD pursuit, and I owe them many thanks: John, Zach, Megan, James, Rob, Rachel, Wojtek, Aaron, Bob, and many others. Likewise, my friends in Baltimore have helped me balance work and life by sharing their love for the city, arts, culture, and nature. Thank you Stephanie, Chad, Jowita, Vahid, Nathan, Alfredo, Ed, Mike, Mary, Karen, Selin, and others.

I would like to acknowledge the support I received from National Institute of Health, which provided the major funding for this work through the NIH BRP grant R01 EB 007969, and the National Science Foundation, for systems infrastructure support through the CISST-ERC NSF

grant EEC9731748. I am also grateful for equipment support from Carl Zeiss Meditec and from The Johns Hopkins University internal funds. Additionally, the ARCS foundation has been very generous in their support throughout my doctoral studies.

Finally, I would like to thank my family (both consanguineal and in-law), especially my lovely wife Hilary, for their boundless love, encouragement, and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Modern patient health care involves maintenance and restoration of health by medication or surgical treatment of illness. The following work focuses solely on microsurgical procedures where surgeons perform high-risk but necessary surgical treatments and face significant technical and human limitations in an extremely constrained environment. Vitreoretinal surgery, in particular, poses enormous challenges due to tissue delicacy, surgical inaccessibility, suboptimal visualization, lack of tactile feedback, and the potential for irreversible tissue damage resulting from unintentional collisions. These factors collectively not only make vitreoretinal surgery the most technically demanding ophthalmic surgery, but also apply similarly to other microsurgical disciplines such as anterior segment ophthalmic surgery, otolaryngology, vascular surgery, and neurosurgery.

The technological innovations of the last two decades have pushed computing and "smart" devices into general surgical practice [1]. Recent developments and miniaturization of sensors, along with real-time computer processing and miniature robotic assistants have the potential to augment the way microsurgery is performed. This dissertation presents the architecture and design principles for a surgical augmentation environment to work in partnership with surgeons to address these fundamental limitations. The ultimate goal is improving clinical care and enabling novel therapies. It is inherently an information-driven modular system integrating robotics, sensors, and visualizations. The integrated system is used to design intuitive and relevant human-machine interferences and generate a particular system behavior to provide active physical assistance and present sensory information to the surgeon. These include manipulation assistance, haptic feedback, intraoperative imaging and force sensing. The performance of these capabilities is demonstrated and evaluated using phantoms and in vivo experiments.

The development of the following concepts and the resulting surgical system is part of a large interdisciplinary project: A Microsurgical Assistant System funded by NIH BRP Grant 1 R01 EB 007969 (PI: R. Taylor). Its overall goal is to develop novel core technology and microsurgical tools with the unique capability to significantly “enhance” the ability of surgeons to perform microsurgical tasks.

1.1 Thesis Statement

A surgical augmentation environment with specific distributed architecture and design methods enables the development of innovative aids and capabilities to address fundamental limitations in microsurgery, especially vitreoretinal surgery.

Here are the elements comprising the argument:

- A computer-integrated environment can augment surgical skill, addressing major challenges in vitreoretinal tasks.

- For a given task, a particular capability is desired based on surgeon preference and available system functionality.

- A modular, information-driven system design can be used by developers to develop such capability.
- By following a design pattern, it is possible to develop an organized system and dynamically reconfigure it to provide new capabilities.
- An integrated system can provide relevant assistance for tasks that are extremely difficult or currently infeasible.
- It can be used to train residents by providing information in a safe setting that would otherwise be gathered by manual trial and error methods.
- Ultimately, such a system is not only clinically relevant, but also supports further research.

1.2 Ophthalmology

Eyesight is considered to be one of the most essential of human senses. It enables us to navigate the complexities of our world by providing most of the information we take in from the environment[1]. Hence, partial visual impairment or a complete loss of vision are serious impediments to one's quality of life, resulting in severe personal, social, and economic consequences [2–6]. According to the World Health Organization there are estimated 285 million individuals in the world who are visually impaired, and 39 million who are blind [7]. Furthermore, 82% of the blind are 50 years old and older. As reported by Pascolini and Mariotti, retinal diseases "are a major cause of visual impairment worldwide, and are likely to become more and more important with the rapid growth of the aging population.". These noncommunicable eye diseases are also indirectly

[1] Around 60% of the cerebral cortex is involved in the perception and interpretation of visual stimuli.

influenced by the modern sedentary lifestyle and poor diets common to the developed world, and increasingly adopted by the developing countries [8,9]. This increase of patients seeking vision-related treatments will require our healthcare systems to provide comprehensive, efficient and high-quality ophthalmological services.

Ophthalmology is the branch of medicine that deals with the anatomy, physiology, and diseases of the eye. Ophthalmologists diagnose pathologies, and suggest and administer appropriate treatments. One specialization is vitreoretinal surgery that deals with surgical management of retinal and posterior segment diseases and disorders. It is considered one of the most challenging microsurgical disciplines, requiring exceptional manual dexterity and years of intense medical training. Surgeons strive to overcome many human and technological limitations, including physiological hand tremor, poor visualization of surgical targets, and lack of tactile feedback in tool-to-tissue interactions. These challenges limit the number and success rates of available treatments, as well as hinder the development of new sight-restoring procedures. The goal of the work presented here is to study and develop technology and methods that address these challenges.

1.3 Human Eye Anatomy

The eye is a spherical organ, 24 mm in diameter, that is both delicate and complex. It is comprised of several structures shown in Figure 1.1. The outside layer of the eyeball includes the sclera, a tough white protective tissue; and the cornea, a transparent layer on the anterior of the eye. The inside of the sclera is coated by a middle layer that consists of the choroid, ciliary body, and iris. The choroid provides some of the vascular circulation for the retina which is the innermost layer coating most of the inside of the eye. The retina is a sensory tissue located on the innermost surface of the eye. It is responsible for light perception and vision, including color differentiation

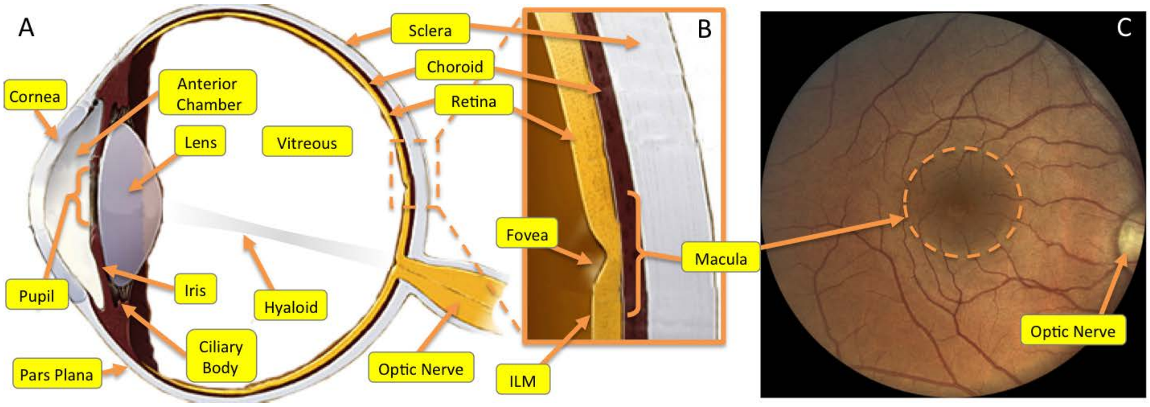


Figure 1.1: A) Cross-sectional diagram of the human eye. (Image credit: BrightFocus Foundation, Inc.) B) Diagram of the macula region. C) Fundus image of the retina.

and the perception of depth. The retina is approximately 100–570 μm thick [10], and contains variable-sized vessels that are 10–100 μm in diameter. The internal limiting membrane (ILM) is a thin layer of tissue (2–3.4 μm [11]) that coats the retina and separates it from the vitreous humor, a jelly-like transparent substance that fills the eye.

The process of human vision involves light entering the eye through the cornea, passing though the pupil in the iris, and then through the lens that focuses it on the retina in the back of the eyeball. The light excites the sensory retina, which generates corresponding electrical pulses. These are preprocessed locally and then relayed to the brain via the optic nerve, where they are interpreted as an image. The optic nerve appears as a circular to oval white disc on the retina, and measures about 1.5–2.0 mm across. The pupil controls the amount of light that enters the eye which is required to create an image, while also minimizing that amount to prevent permanent retinal damage from overexposure. Prevention of such overexposure is particularly important for the macula region, which is a 6 mm oval-shaped spot near the center of the retina, which is specialized for high-acuity vision. A small pit near the center of the macula is the fovea (1.5 mm diameter). It contains the largest concentration of cone cells in the eye and is responsible for central, high-resolution vision. The visual input from this region contributes a substantial portion of the brain's visual capacity. Therefore, damage to the macula results in loss of central vision, and severe impairment to a person's quality of life.

1.4 Vitreoretinal Conditions and Treatments

Vision may be affected by a variety of conditions of the vitreous and retina. These include diabetic retinopathy, macular degeneration (AMD), retinal tears and detachments, macular pucker (epiretinal membrane), retinoblastoma, eye cancer, uveitis, retinitis pigmentosa, etc. Many of these are currently not curable, e.g., AMD, and require treatment to prevent further deterioration. Although orally administered pharmaceuticals would be the ideal solutions to the above diseases many still require high-risk surgical interventions, and in many cases a combination of both. The most prevalent diseases are summarized here:

Diabetic Retinopathy (DR) is one of the leading causes of blindness [12]. Retinopathy is an ocular manifestation of a systemic disease such as diabetes. It is caused by microvascular changes of the retina. In some people, existing blood vessels may swell and leak fluid. In others, the retina develops new, distorted blood vessels in an effort to increase the retinal oxygen supply. Over time, the vessels leak, develop uncontrollably, and obstruct the vision path to the fovea. Patients experience blurred vision from macular edema, and darkening or distorted images with progressively more blurred vision associated with vitreous hemorrhages. Another complication is the formation of scar tissue from the damaged blood vessels, which causes traction on the retina and can lead to retinal detachment.

Currently DR is not curable but there exist a number of treatments that are successful in halting the progression of vision loss. A common treatment for retinopathy is targeted laser photocoagulation, where the problematic (leaking) vessels are cauterized with a laser before the disease progresses. For more advanced cases, panretinal photocoagulation is applied, which generates scattered burns all over the retina, which destroy retinal cells that produce vascular endothelial growth factor (VEGF). By reducing VGEF, the regrowth of neovascular tissue is decreased. However, in many cases such treatments result in high recurrence rates [13]. An alternative is direct injection of anti-VEGF into the eye, which reduces the extent of vessel outgrowth [14]. In cases of severe hemorrhaging into the vitreous, a vitrectomy (removal of the vitreous) is required to restore visual acuity. This step is also necessary in treating retinal detachments.

Age-related macular degeneration (AMD) is currently a major cause of severe vision loss in people over the age of 65 years [15]. AMD affects 8 million Americans and is expected to increase by more than 50% by the year 2020 [16]. The “wet” form of AMD begins with yellow/white spots (drusen) forming between the retinal pigment epithelium and the underlying choroid. These areas leak fluid behind the retina and damage the cones in the macula responsible for acute vision. This may cause retinal scarring and detachment. The typical symptoms include wavy distortions of vision, blurriness, blind spots and limited central vision. In the more common “dry” form of AMD, the drusen become large and numerous, which leads to spotty, or distorted vision, and diminished or changed color perception. As the disease progresses, the retinal cells in the macula atrophy, causing severe loss of central vision.

Similarly to treatments for DR, “wet” AMD treatments are based on anti-VEGF therapy, where injections into the eye attempt to minimize the effects of specific growth factors that promote the growth of abnormal new blood vessels. Unfortunately, this approach can only slow down vision loss since currently there is no cure for “wet” AMD.

The “dry” form of AMD is occasionally treated similarly to DR, using laser photocoagulation. Photocoagulation is used to treat choroidal neovascularization, the growth of abnormal fragile vessels prone to bleeding. This method must be used away from the fovea, since the treatment itself can create a large retinal scar associated with permanent visual loss [16]. Another treatment is ocular photodynamic therapy, in which an intravenously administered, light-sensitive drug concentrates in new blood vessels and is activated with a laser beam focused over the macula, effectively damaging the abnormal blood vessels. The risk of new blood vessels developing after laser treatment is high. Treatment can control the progress of the disease, but in many cases, vision deterioration is unavoidable. Vitreoretinal surgery procedures have shown some limited success rates but are often avoided due to risk of complications from difficulty of the surgery [16].

Retinal Tears and Detachment are considered serious retinal damage, which can cause partial and temporary loss of vision. The most common cause of retinal tears is from vitreous traction. When the vitreous contracts and peels away from the retina, it sometimes pulls on the retina and creates a tear. The patients with retinal tears may see floaters or flashes of bright light even when their eyes are closed. The full-thickness retina tear in the fovea is called a macular hole, and causes distortion or blurriness to the person’s central vision. Retinal tears may be addressed with laser ablation or with cryotherapy by scarring the retina around the tear that causes it to attach to the choroid. This prevents fluid buildup under the retina by sealing the tear.

This treatment prevents a more serious condition: retinal detachment where the retinal tissue is separated from the choroid[2]. An untreated detachment will cause the loss of vision (dark spots, shadows, warped vision) in the detached area and can be permanent if it is not addressed early. Small detachments are treated like retinal tears with a laser, but larger ones are repaired surgically. One approach is a scleral buckle that alters the shape of the eyeball to encourage re-attachment of the retina. Another one is pneumatic retinopexy, in which the surgeon injects a gas bubble inside the vitreous cavity, after which the laser or freezing treatment is applied to the detached retina. The gas bubble forces the retina against the eye wall so it can re-attach. One of the more common treatments is a vitrectomy followed by the gas bubble treatment or a similar one involving silicone oil. Some surgeons also remove the ILM to improve the visual and anatomic success of macular hole treatment, but results of this technique are still debated [17]. Although full vision is often not fully restored, the above procedures are mostly successful in preventing blindness.

Retinal Vein Occlusion (RVO) occurs when a clot obstructs blood outflow in a central (CRVO) or branch vein (BRVO) of the eye. They are the second most common retinovascular disorders after diabetic retinopathy [18] and have no common effective treatment [17]. Untreated CRVO can lead to severe damage to the retina and eventual blindness due to several factors: the resulting restriction in blood supply (ischemia), and swelling (edema); neovascularization; and neovascular glaucoma. The patient's symptoms include sudden partial or full blurring or vision loss. Similarly, BRVO involves a blockage that damages a part of the retina that is serviced by the branched vein.

Most treatments for RVO deal with management (e.g., laser photocoagulation, pharmacologic therapy with anti-VEGF injections) of the resulting pathological condition and have shown

[2] There are many causes for retinal detachments: diseases, injuries, etc.

limited success in improving visual acuity [17]. These approaches tend to effectively stabilize and address the complications of disease without attempting to reverse the underlying pathology. There have been numerous failed and unadopted treatments for RVO [19], e.g redial optic neurotomy, where the pressure on the vessels is relieved by direct incision near the optic nerve [20, 21].

Direct endovascular procedures are also promising but technically challenging with current instrumentation. These include: retinal vein cannulation with infusion of a tissue plasminogen activator to remove the blood clot [22, 23]; chorioretinal venous anastomosis, which aims to bypass the occluded vein by an alternative route (e.g., a shunt between a retinal vein and the choroids) to improve retinal outflow and relieve the venous obstruction [17]; and microvascular retinal stenting, a potential solution for patients who suffer from recurrent blockage despite pharmacologic treatment or cannulation [24].

Epiretinal Membrane (ERM) also called Macular Pucker, is a common condition where a thin sheet of scar tissue forms on the surface of the retina. ERM develops when retinal cells and other tissues within the eye become liberated into the vitreous gel and eventually settle onto the surface of the macula. Sometimes these cells grow into a prominent membrane that contracts and generates traction on the retina, causing it to “wrinkle,” which compromises retinal function. Visual dysfunction resulting from ERM includes blurred vision, image distortion, and altered image size. ERMs are present in 2-6.4% of people [25].

Vitreoretinal surgery is the only known treatment for ERMs. It involves removing of the vitreous, then peeling the problematic membranes of the retina without causing further injuries. Sometimes an air or gas bubble is used to improve healing and flatten the retina. ERM peeling is high-risk and can result in retinal detachment, which occurs in 1 to 2 out of 100 cases following ERM surgery.

1.5 Vitreoretinal Surgery

In current practice, retinal surgery procedures are carried out in a standard operating room (OR) outfitted with a stereo operating microscope. Typical procedures are 1–3 hours in length and do not require an overnight stay. The patient is prepared on a standard operating room table laying supine with the head located under the microscope (see Figure 1.2A). The lead surgeon sits superior to the patient's head. The surgeon's hands rest on a supporting ring attached to the operating table. According to Galbraith, the most common cause of an unsteady hand during surgery is lack of support for the wrist and the hand. To minimize this, the surgeon also rests his/her hands on the patient's forehead. This allows the surgeon to react rapidly in an emergency when the patient's head moves unexpectedly, especially since in most cases the patient is awake, locally anesthetized[3], and free to move. The assistant surgeon sits beside the patient's head, often using the auxiliary microscope eyepieces to monitor the procedure. The assistant is responsible for irrigation and removal of fluids and placement of various visualization lenses on the cornea.

A single eye is operated on per session, and in most cases, three incisions in the sclera (sclerotomy) are required: one for infusion to control the intra-ocular pressure, one for intraocular illumination (fiber-optic "light pipe"), and one for a surgical instrument (see Figure 1.2B). The tools are inserted through two trocars located in the pars plana region, the meeting point of the iris and the sclera on the ciliary body. The surgeons operate in bimanual fashion with the "light pipe" in one hand and one of the following in the other hand: hook, forceps, laser, vitreous cutter, fragmatome, etc. Typical instruments have a ∼32–35 mm long tubular shaft that is 0.5–0.9 mm in diameter

[3] Full anesthesia is required for anxious patients and children.

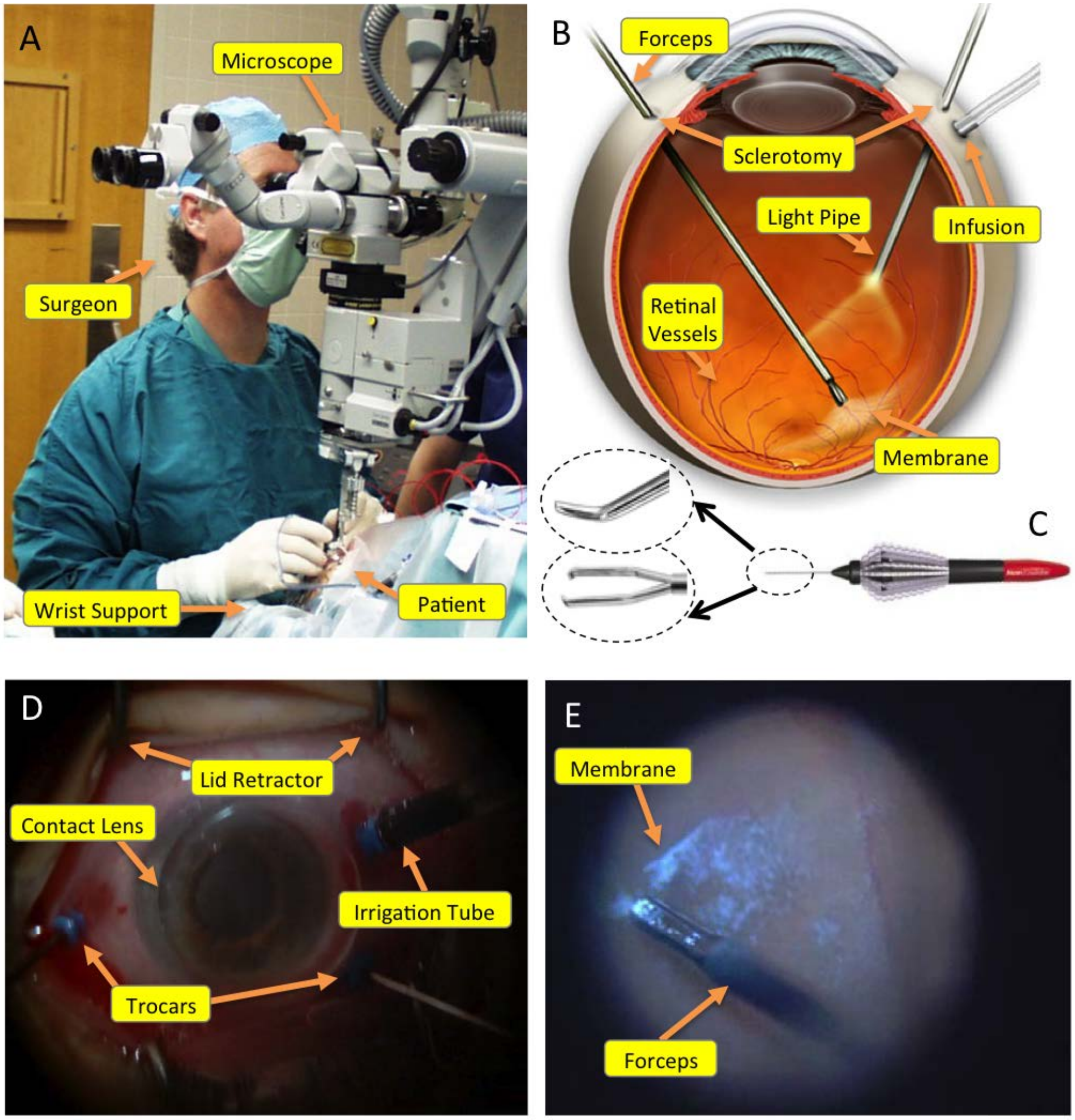


Figure 1.2: Vitreoretinal surgery: A) Position of the patient and lead surgeon in the operating room. B) The layout of surgical instruments in the eye during membrane peeling. (Image credit: www.eyemdlink.com) C) Disposable vitreoretinal surgery instruments: hook and forceps (Alcon, Inc.) D) Surgeon's microscope view of the eye with instruments inserted through the trocars and a magnification contact lens placed on the cornea. E) Surgeon's view of membrane peeling. Microscope view of the retina through magnification contact lens.

(20–25 Gauge) (see Figure 1.2C). The instruments are held like a pen or pencil and the active parts are actuated with a squeeze (forceps) of the thumb and the index finger, or with a foot pedal (e.g., vitrector). Provided that the hands are supported, fine movements of these instruments are made only with these fingers [26].

Once the patient is prepared for surgery: locally anesthetized, face draped, eye lid retracted, etc., the surgeon creates the three access incisions in the pars plana and performs a vitrectomy procedure. It involves removing the natural vitreous fluid of the eye and replacing it with a saline liquid. This often-required step provides unobstructed access to the retina. During the vitrectomy the intraocular pressure drops significantly. The infusion (saline supply line) is constantly monitored and adjusted to prevent the eye from collapsing. In modern ORs this function is performed by a vitrectomy surgical system (e.g., CONSTELLATION Vision System, Alcon Inc.) that also provides other functions and instruments like a laser, fragmatome, irrigation, aspiration, diathermy, etc. The configuration of surgical devices is often done through voice commands to the nurse technician, who adjusts the parameters via a touch-sensitive graphical user interface, or more conventionally, with physical buttons and knobs. Some of the devices produce additional audio feedback to reinforce their state of operation or configuration. For example, the laser system creates a periodic “clicking” when it is in the active state. The newer multifunction systems provide voice confirmations in response to changes in operating parameters. For some functionality, the surgeons directly control these electromechanical devices through a multifunction foot pedal interface (e.g., vitreous cutter on/off.)

Throughout the operation the eye is constantly torqued by the surgeon with the two bimanual instruments that are inserted into the eye. This provides required visual access to the retina, since the pupil tends to be very small. The surgeons have the option of using a wide angle lens that provides a panoramic view of the fundus (up to 120°.) This lens is about 15 mm in diameter, is attached to the microscope and swings down $\sim$ 1 cm above the cornea. It has large aberrations in the peripheral sections and is mostly used for fine work on the retina. In general, microscope zoom is not changed (controlled by foot pedals) very often during the procedure, but the surgeons can use a special contact lens placed on the patient's cornea for higher magnification, although this results in a narrower field of view (e.g., 20°flat contact lens). This is often in place during very fine tasks like membrane peeling.

Once the vitreous is removed, the surgeons can perform a number of surgical treatments for diseases described earlier. The following are the steps for a typical operation that involves the patient arriving early in the morning, and leaving the facility later in the afternoon.

1. Pre-surgery consultation by staff.
2. Place intravenous line.
3. Administer dilatation drops to the eye.
4. Roll patient into the OR.
5. Attach vital monitoring equipment.
6. Administer a five-minute sedation (gas).
7. Inject local anesthetic (block) behind the eye.
8. Patient is slightly sedated but breathing on their own.
9. Clean eye area with iodine.
10. Drape the eye with a sticky plastic.
11. Make an incision in the plastic and place the lid retractor to immobilize the eye.
12. Place microscope over the patient.
13. Coat cornea with gel to keep it moist (often reapplied by assistant during the procedure).
14. Perform first sclerotomy and install trocar.
15. Attach infusion.
16. Place two more trocars.
17. Vitrectomy starting from the anterior towards the retina.
18. Inject markers to enhance visibility of the vitreous remnants.
19. Continue with the vitrectomy.
20. Peel hyaloid with the vitreous cutter.

21. Perform required treatments: membrane peeling, laser treatment, buckle, etc.
22. Replace intraocular fluid with silicone or gas bubble.
23. Remove trocars and suture the sclerotomy sites[4].
24. Remove all draping.
25. Add creams over the eye and add a patch.
26. Remove patient from the OR.

Many of the surgical skills that are discussed in this thesis can be found in two target procedures: retinal membrane peeling, and more experimental retinal vein cannulation.

1.5.1 Membrane Peeling

Epiretinal membrane (ERM, see Section 1.4) peeling is a common vitreoretinal surgery procedure but it remains one of the most technically challenging and highly consequential [27]. The procedure involves the surgical removal of scar-like tissue that is 10–80 μm thick [28] from the surface of the delicate retina, without compromising the retina's structure or function.

Surgical removal of an ERM begins with a vitrectomy that provides clear visual and physical access to the retina. First, the surgeon visually locates the transparent ERM and identifies the outer margin of the membrane as a potential target "edge". For this step, the surgeon relies on a combination of pre-operative fundus and Optical Coherence Tomography (OCT) images, direct visualization often enhanced by coloring dyes, and mechanical perturbation in a trial-and-error technique using a diamond-dusted membrane scraper [27, 29].

The localization of these candidate peeling edges is difficult. Surgeons often rely on inconsistent and inadequate preoperative imaging due to developing pathology, visual occlusion, tissue swelling, and/or other direct effects of the surgical intervention. Furthermore, precision membrane peeling is performed under very high magnification, allowing only a small area of the retina ($\sim$

[4] In some cases small (25 Gauge) self-sealing incisions are used that do not require sutures.

5–15%) to be visualized at any one time. This requires the surgeon to mentally register sparse visual anatomical landmarks relying only on information from pre-operative images, and at the same time, to consider any changes in retinal architecture due to the operation itself. Some ERMs provide clear visual evidence of edges, while others require creation of an edge by the surgeon. This may be performed by incising the membrane surface, by bluntly creating an edge using a microvitreoretinal blade or a 25-gauge needle, or by gently grasping the membrane with a forceps and creating a tear in the ERM.

Once an edge is located or created, various tools such as pick or forceps, may be employed to engage and delaminate the membrane from the retina, while avoiding damage to the retina itself. It is imperative that all of the ERM which can be millimeters in diameter, is removed. This process often requires a number of peels in a single procedure. Membrane peeling and associated surgical challenges can be summarized in the following steps:

1. Identification of ERM: ERM is difficult to locate visually.
2. Creation of target edge if necessary: underlying layers are difficult to assess visually, and imprecise tool control can damage the retina.
3. Lifting/grasping of membrane edge : unstable tool control and lack of force feedback can cause collision or tearing of the retina.
4. Delaminating the membrane: unstable tool control and lack of force feedback can cause retina tears.
5. Assess retina for complete membrane removal: Retina is difficult to completely inspect visually for the remnants of the membrane.

Each of these steps requires excellent visualization, high levels of manual dexterity and micro-instrumentation. Furthermore, this procedure is performed with a comparatively large metal instrument that does not offer tactile sensation. During this time, a patient's involuntary and voluntary movements must also be manually compensated for by the surgeon while the instrument is in direct contact with fragile intraocular tissue. Incorrect micron-magnitude movements can cause retinal tears, retinal detachment, visual field defects, retinal hemorrhage, local retinal edema, nerve fiber layer injury, and macular holes, all of which can contribute to blindness. Other complications become similar to eyes undergoing pars plana vitrectomy, such as cataracts, macular trauma, and light toxicity.

The same peeling technique may also be used in the peeling of the internal limiting membrane (ILM), which is thought to improve the results in ERM and macular hole surgery [11, 30, 31]. The benefit of ILM peeling is that it ensures complete removal of the posterior hyaloid or any overlying ERMs, since both grow over the ILM. However, ILM is friable, transparent, and only a few microns thick, making it very difficult to visualize, grasp, and remove safely. Its difficult removal may increase the risk of trauma to the retina. Additionally, the dye used to enhance its visualization (indocyanine green (ICG)) has been shown to be toxic to the retina. ILM peels are more difficult than ERM peels, and should only be performed by expert surgeons [32].

1.5.2 Vitreoretinal Cannulation

A retinal vein occlusion results when a clot forms in the central retinal vein or one of its branches, often leading to blindness if not treated correctly (see Section 1.4). Current laser-based treatments have proven ineffective in treating this occlusion, resulting in high recurrences or complications in patients [33]. The most promising treatment is intravenous injection of clot-dissolving drugs (e.g., recombinant tissue plasminogen activator (t-PA)) directly upstream of the occlusion [17, 22, 34]. However, due to the small vessel size, fragility of the surrounding tissue, and difficulty in reaching the site of occlusion, the free-hand intravenous injections have proven clinically challenging and are still considered a very experimental procedure.

The surgery begins with standard pars plana vitrectomy and the removal of the posterior hyaloid. Once the retina is accessible, the location of the occlusion is identified visually. The surgeon brings the injection instrumentation into the eye through the trocar. This could include a very small needle (40 gauge) or a glass pipette with 10 μm ID lumen. The distal end of the needle is then aligned in parallel with the problematic blood vessel, which may be very challenging due to the constrained geometry comprising eye shape, visualization angle, trocar placement, and vessel location and orientation. The needles or pipettes need to be very small in order to cannulate the extremely delicate vessels that are themselves between 10 and 130 μm in diameter. This makes visualization of the injection elements difficult. The cannulation step involves the surgeon translating the pipette down onto the vessel, then along it to puncture the vessel wall. Once the cannulation is visually verified, a bolus of 200 μg/ml t–PA [17] or other drug is fully injected for several minutes. During this time the vein is very vulnerable to rupture from excessive forces applied by the injection instrument. After drug delivery, the pipette or needle is slowly retracted and removed from the eye.

Besides difficulties in confidently determining the state of cannulation from visual cues, the principal clinical barrier is that the target blood vessel diameter is smaller than the typical 108 μm amplitude of physiological tremor [35]. This makes the safe free-hand introduction of the tool into the vessel nearly impossible, and the maintaining of the cannulation for an extended period even more challenging. Besides managing hand tremor, the surgeon has to compensate for inadvertent patient movement (e.g., from respiratory motion) to prevent damage to the vessel or the retina. The individual steps and challenges of the cannulation procedure are summarized below:

1. Identification of retinal vein: difficult to incorporate preoperative angiogram information into microscope view of the retina.
2. Positioning of the injection instrument at the cannulation site : difficult to locate instrument visually; geometry of the procedure is constraining.
3. Cannulation of the vessel: unstable tool control and lack of force feedback can cause vessel and retina damage.
4. Inject the medication into the vessel: long injection time, unstable tool control and lack of force feedback can cause vessel and retina damage.

5. Removal of cannulation instrument from vessel: unstable tool control and lack of force feedback can cause vessel and retina damage.

This direct injection of t-PA via retinal vein cannulation requires a stabilization system for reliable and safe intravenous administration of the drug without destroying the vessel. Without such assistance this technique is still unproven and can lead to serious ocular complications, including vitreous hemorrhage, formation of retinal tears, retinal detachment, glaucoma, endophthalmitis and phthisis bulbi [17].

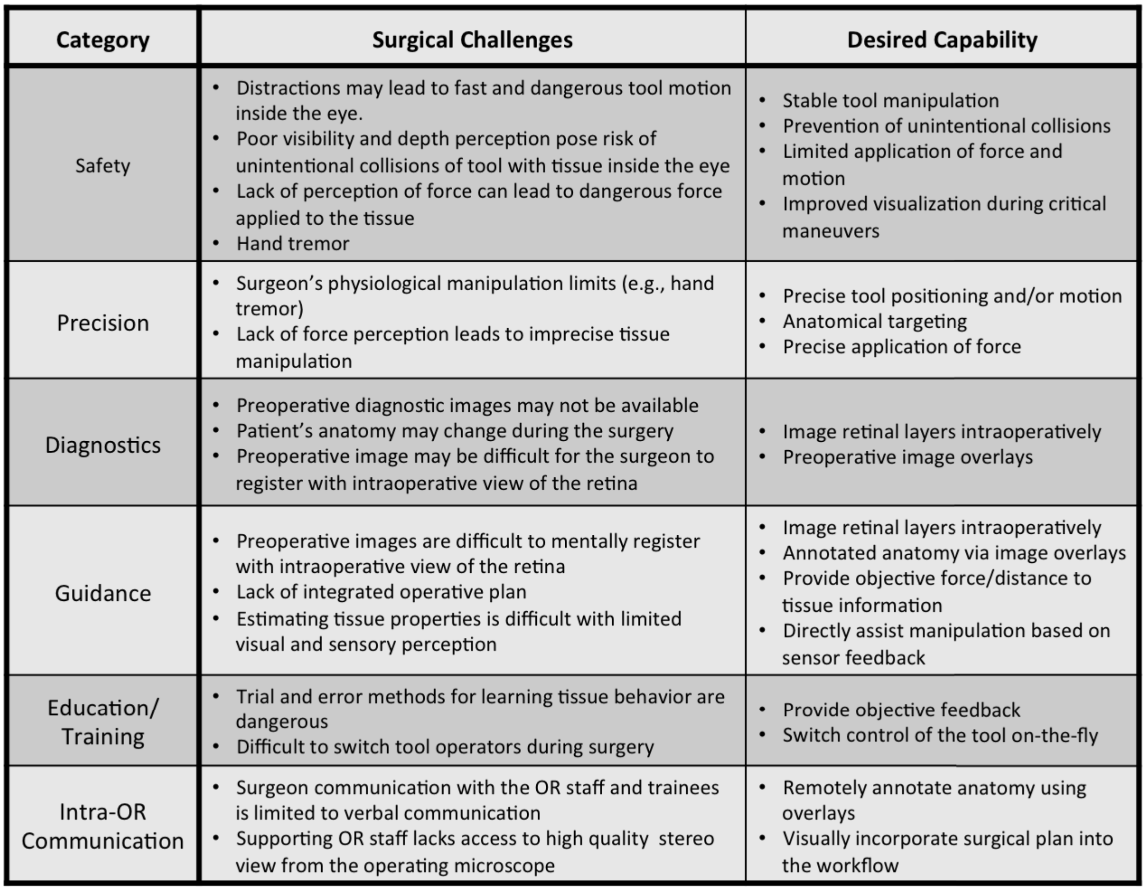
1.5.3 Technical Challenges in Retinal Microsurgery

The technical demands placed on the surgeon by vitreoretinal procedures are extreme. Surgeons are challenged by many human-sensor-motor limitations in an environment where targets could move at any moment (from involuntary patient movement) and are severely constrained by the geometry of the anatomy and the required miniaturization of the instruments. A significant example is physiological hand tremor, which comes from involuntary motion, with a maximum amplitude of about 100 μm at a frequency range from 7 to 17 Hz [35]. Hand tremor is a large contributing factor to the difficulty of micromanipulating delicate 1–100 μm retinal structures, and only becomes more pronounced with aging of the surgeon. Another challenge is lack of tactile feedback in tool-to-tissue interactions where the associated forces are well below human perception ($<$ 7 mN) [36]. This is further complicated by force attenuation from tool–trocar interaction, making any unassisted force sensing in retinal tissues manipulation impossible.

Surgeons are also limited by their ability to clearly visualize surgical targets. Even though today's state-of-the-art operating microscopes are thought to be very good and make vitreoretinal surgery possible, their effectiveness can be severely affected by developing pathology and visual occlusions. Furthermore, visualization microscopes are adjusted for high magnification to provide high resolution, and show only a small area of the retina at any one time. Navigating through this "keyhole" is difficult, requiring the surgeon to mentally register sparse visual anatomical landmarks with a mental map of the retina. This is further complicated by the need to mentally integrate often inadequate pre-operative images and surgical plan information "on-the-fly". Such information may evolve during the operation due to the developing pathology, tissue swelling, and other direct effects of the surgical intervention. Other cognitive stress factors on the surgeon include: constant multitasking of visuomotor control of the instruments; managing operating room staff via verbal communication; patient status monitoring; surgical navigation; and integration of patient-specific preoperative information.

Surgeon fatigue is another factor that should be considered since the procedures are relatively long, usually 1–3 hours, and in some instances 5–6 hours. The poor ergonomics of the standard operating microscope requires rigid positioning by the surgeon for prolonged periods, which is the major cause for back and neck ailments and is the most common cause of disability among vitreoretinal surgeons [37, 38]. Beyond its toll on the surgeon, an extended operation time may be detrimental to the patient because of common surgical factors, but also because the patient's eye may be exposed to toxic levels of intense direct light on the retina. Furthermore, surgeons are under great pressure from hospitals to lower operating costs by shortening procedures.

In addition, tool limitations, such as lack of proximity sensing, force sensing, or other "smart" functions, are important factors that contribute to surgical risk and reduce the likelihood of achieving surgical goals. Current instruments do not provide physiological or even basic interpretive information, e.g., the distance of the instrument from the retinal surface, the depth of instrument penetration into the retina, or an indication of the force exerted by the instrument on the retinal tissues. Surgical outcomes (both successes and failures) are limited, in part, by these technical



Category	Surgical Challenges	Desired Capability
Safety	• Distractions may lead to fast and dangerous tool motion inside the eye. • Poor visibility and depth perception pose risk of unintentional collisions of tool with tissue inside the eye • Lack of perception of force can lead to dangerous force applied to the tissue • Hand tremor	• Stable tool manipulation • Prevention of unintentional collisions • Limited application of force and motion • Improved visualization during critical maneuvers
Precision	• Surgeon's physiological manipulation limits (e.g., hand tremor) • Lack of force perception leads to imprecise tissue manipulation	• Precise tool positioning and/or motion • Anatomical targeting • Precise application of force
Diagnostics	• Preoperative diagnostic images may not be available • Patient's anatomy may change during the surgery • Preoperative image may be difficult for the surgeon to register with intraoperative view of the retina	• Image retinal layers intraoperatively • Preoperative image overlays
Guidance	• Preoperative images are difficult to mentally register with intraoperative view of the retina • Lack of integrated operative plan • Estimating tissue properties is difficult with limited visual and sensory perception	• Image retinal layers intraoperatively • Annotated anatomy via image overlays • Provide objective force/distance to tissue information • Directly assist manipulation based on sensor feedback
Education/ Training	• Trial and error methods for learning tissue behavior are dangerous • Difficult to switch tool operators during surgery	• Provide objective feedback • Switch control of the tool on-the-fly
Intra-OR Communication	• Surgeon communication with the OR staff and trainees is limited to verbal communication • Supporting OR staff lacks access to high quality stereo view from the operating microscope	• Remotely annotate anatomy using overlays • Visually incorporate surgical plan into the workflow

Figure 1.3: Technical challenges faced by vitreoretinal surgeons and corresponding capabilities to address them categorized by common functions.

hurdles that cannot be overcome by current instrumentation. For example, in the most technically demanding cases, there may not be a set of tools that allows the “typical” vitreoretinal surgeon to safely remove epiretinal tissue to ensure surgical success.

The vireoretinal challenges are summarized by category (safety, precision, diagnostics, guidance, education, training, and OR communication) in Figure 1.3. The figure also lists the corresponding capabilities that augment surgical skills to addresses these challenges.

1.5.4 Acceptance Considerations

Altering conventional surgical practice is a great challenge, both for surgical system designers and for the surgeons. Before use on humans, any new medical device or new procedure requires thorough analysis and testing to prove that it is beneficial to the patient and at least as safe as the current treatment alternatives. Besides addressing a specific surgical need, surgical innovations need to show shorter-term efficacy and follow strict safety requirements. Additional benefits could include minimizing blood loss, shortening the patient’s hospital stay, decreasing pain in rehabilitation, and reducing overall recovery time. Of course, for a solution to be realistically viable it has to also be cost effective. A case could be made for very expensive or complex surgical systems that save lives or significantly improve the patients’ quality of life.

There exist many reasons why surgeons may be reluctant to adopt new technologies [39]. The accountability for surgical outcomes is a big concern when using new and radically different surgical technologies or protocols. Ultimately, it is the surgeon’s responsibility to make the “safe” decision for a patient’s course of treatment. Naturally, most surgeons are inclined to choose familiar instrumentation and procedures, and to rely on their current skillset with a predictable level of success. The resistance to change is often great with the introduction of a new technology that requires adequate training and often comes with a steep learning curve. Since surgeons are busy professionals, they are reluctant to devote significant non-OR time to additional education. Furthermore, they are often pressured by profit-driven institutions to perform as many procedures as possible in the shortest amount of time, instead of training, or employing new procedures or technologies that may increase operation times.

Any new medical technology (e.g., device, system, service, or procedure) has to be transparent in its function and provide intuitive control and feedback for the surgeon. This is especially important in the initial stages of adoption, becoming less so when the surgeon builds enough trust and comfort in its use to allow for some level of semi-autonomous assistance. For example, surgeons are much more likely to take up hands-on “smart” tools than to use an autonomous robotic device that performs a task without any direct input. Additionally, such technology needs to address a clear surgical problem, and not generate additional problems for the surgeon. Even though a technology may provide an essential capability on its own, it still needs to fit within the overall surgical workflow, which includes setup, maintenance and actual use. The design of such technology has to be centered on the needs of the surgeon and the application while also considering the well-being of the patient. This includes user interfaces as well as requisite training.

Many of these factors are considered in the development of the following surgical system, while some are omitted for the sake of the development of the first proof of concept[5]. Many of the factors that are examined can be addressed with sufficient engineering effort and standard solutions that are common to commercial surgical devices.

[5] E.g., the EyeRobot is made from off-the-shelf components and in its current form is difficult to position near a patient. However, it is sufficient for rabbit and desktop experiments.

1.6 Proposed Solution

Vitreoretinal surgery pushes many natural human abilities to their limits, including dexterity, hand-eye coordination, sharp stereo vision and fine sense of touch. These factors affect surgical performance in varying amounts based on the surgical task, the pathology, and the surgeon's experience and natural physical ability. To facilitate the development of solutions that overcome these potential shortcomings, common surgical procedures, tasks, and maneuvers were deconstructed and their individual challenges were considered independently (Figure 1.3) during the initial development and testing. This divide-and-conquer approach leads to the development of an integrated system that includes robotic devices, innovative real-time sensing, preoperative imaging, intuitive interfaces, visualization, and the facilitation of information exchange between these components. The system is called the Surgical Assistant Workstation for eye surgery (*eyeSAW*).

Figure 1.4 shows example hardware devices and software components that comprise *eyeSAW*. These include robotic assistants: the cooperatively-controlled EyeRobot (see Section 4.3) that provides stable control and precise positioning of the surgical tool held simultaneously by the robot and the surgeon; the Micron device (see Section 4.2) is handheld and is designed to reduce hand tremor; a force sensor technology (see Section 5.1) that is built with optical fibers and is incorporated into surgical instruments to sense sub-milliNewton forces inside the eye; an optical coherence tomography (OCT) imaging system that uses optical fibers embedded in surgical instruments to provide cross-sectional intraocular tissue imaging and range sensing; a stereo-video microscopy system with a 3D flat panel display and associated computer vision functionality including tool and retina trackers and augmented reality video overlays; and graphical, aural and haptic user interfaces. Figure 1.5 shows a photo of the physical system prototype used for the bench-top development and testing.

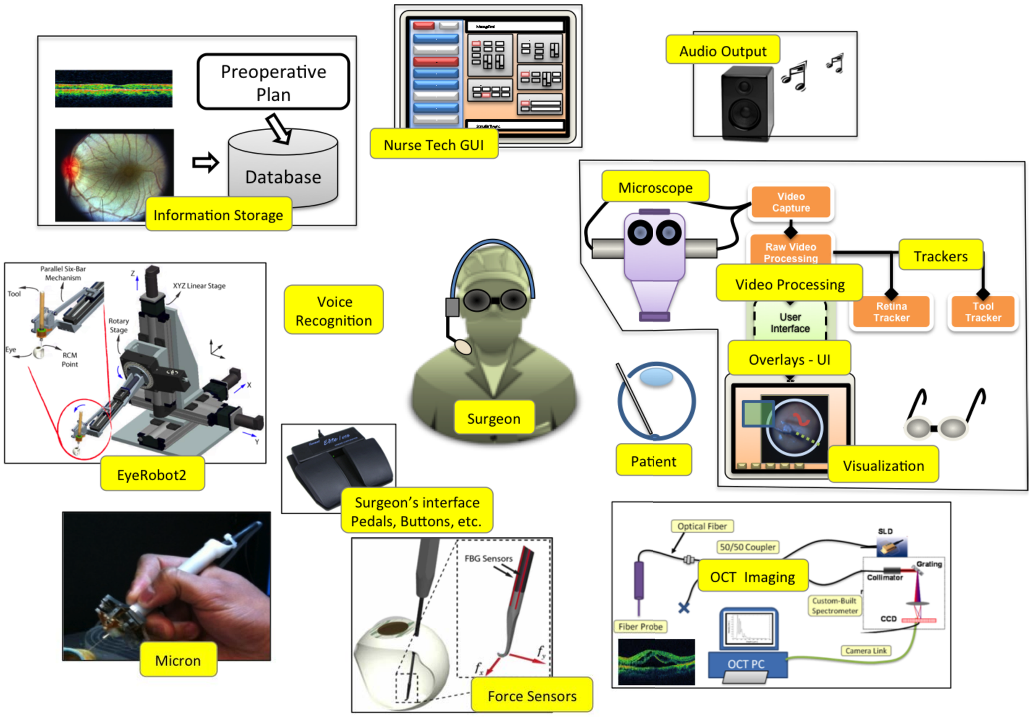


Figure 1.4: Examples of devices in the *eyeSAW* system.

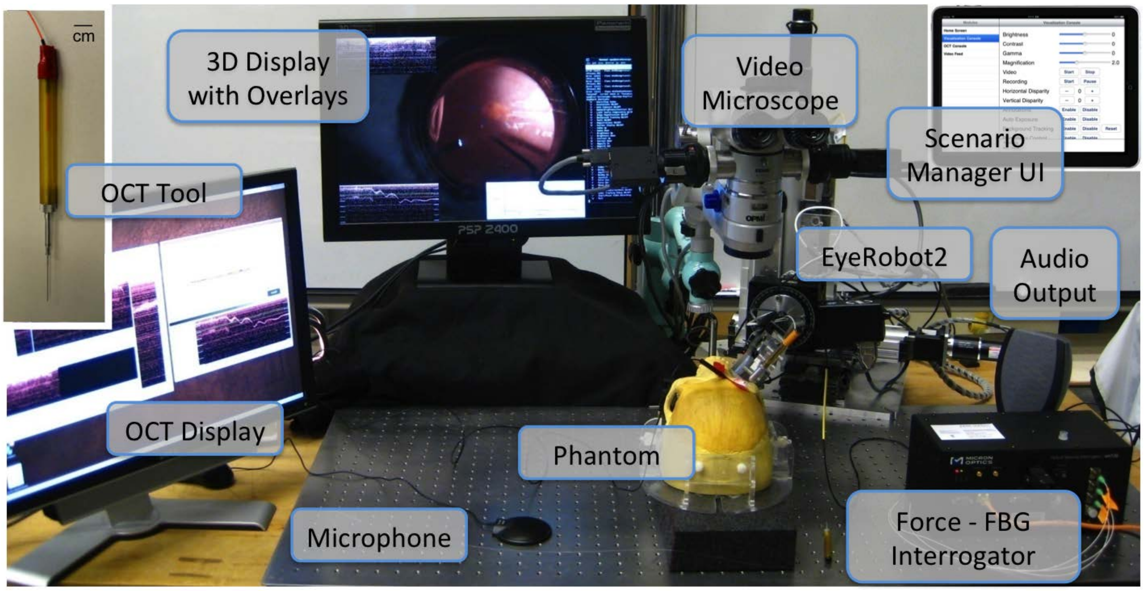
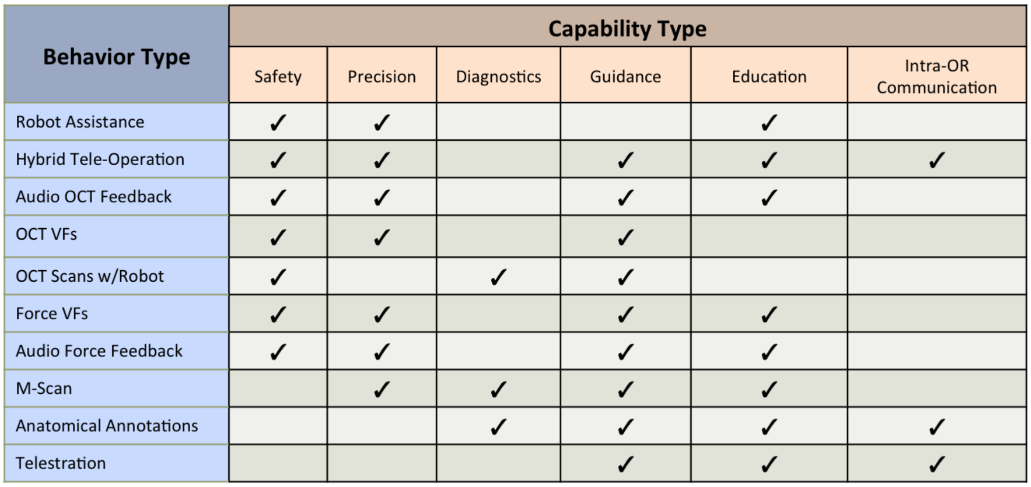


Figure 1.5: Photo of *eyeSAW* devices in an experimental environment.

Throughout the development process it became apparent that many of the desired surgical capabilities were provided only by a tight interaction between the *eyeSAW* components, where each component's information and functionality could be easily shared, and the surgeon was the focus of the system's behavior. This led to the development of the notion called *Behavior*[6], which is the function and configuration of particular system resources to provide specific capability (function or information) to the surgeon. A simple example is Audio Force Feedback Behavior (see Section 5.3), where tool-to-tissue force exerted by the surgeon is measured by the micro-force sensor and is communicated to the surgeon using corresponding auditory alarms. This capability augments the surgeon's ability to sense forces near the human-perception threshold by providing immediate force guidance and potentially reducing risk (improving safety) without significantly changing the surgeon's workflow. Furthermore, the quantitative information can be used to educate novice

[6] Specific Behaviors are described in full detail in later chapters.

surgeons to associate other cues (e.g., visible tissue deflections) with the forces that are being applied to the tissue.

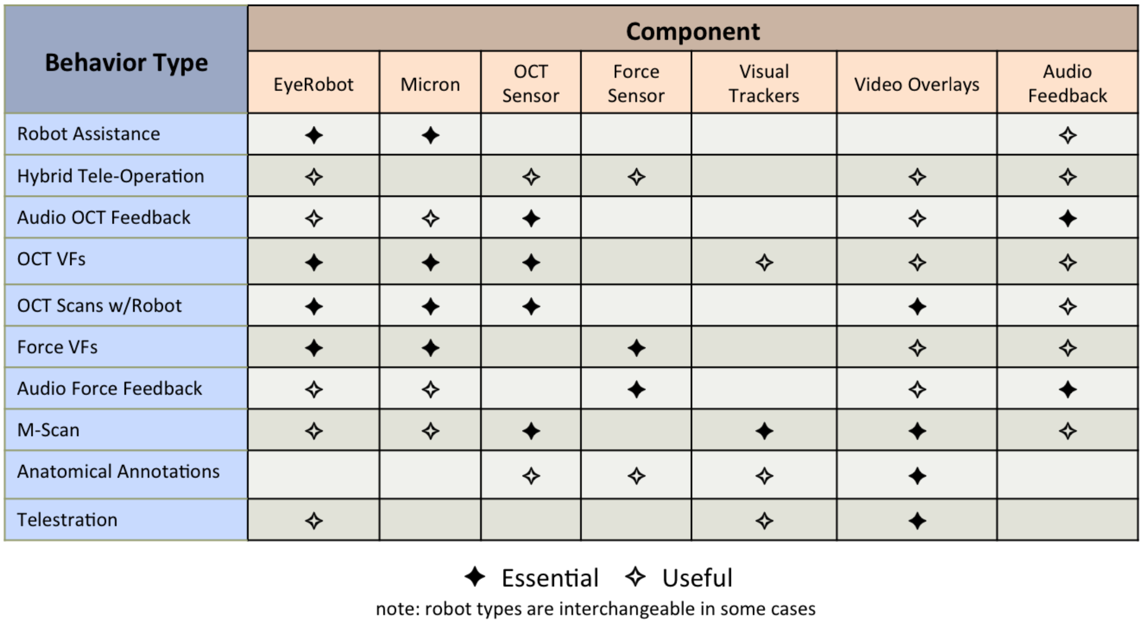


Behavior Type	Capability Type					
	Safety	Precision	Diagnostics	Guidance	Education	Intra-OR Communication
Robot Assistance	✓	✓			✓	
Hybrid Tele-Operation	✓	✓		✓	✓	✓
Audio OCT Feedback	✓	✓		✓	✓	
OCT VFs	✓	✓		✓		
OCT Scans w/Robot	✓		✓	✓		
Force VFs	✓	✓		✓	✓	
Audio Force Feedback	✓	✓		✓	✓	
M-Scan		✓	✓	✓	✓	
Anatomical Annotations			✓	✓	✓	✓
Telestration				✓	✓	✓

Figure 1.6: Surgical capabilities provided by Behaviors.

The Force-scaling Behavior is a more complex example that combines the function of the micro-force sensor with the functionality of the EyeRobot, providing precision positioning and stable manipulation. By using the robot's virtual fixtures framework (see Section 4.3.4), the force sensed at the tool tip can be scaled up (e.g., by 500 times) and rendered at the tool handle during cooperative control. The surgeon is provided with direct haptic guidance, improving tissue manipulation and potentially improving the safety of the surgical task. This Behavior can simultaneously coexist with the Audio Force Feedback Behavior and other robot Behaviors, such as the Virtual RCM (see Section 4.4), which constrains robot motion so that the eye does not freely translate, but allows the surgical instrument to rotate and translate inside the eye. Naturally, Behaviors complement each other, and the surgeon may elect to use a combination of Behaviors for particular tasks.

Other Behaviors can facilitate communication between an expert surgeon and a surgical trainee or the OR staff. Through the Telestration Behavior (see Section 7.13), a surgeon sitting at a remote console is able to virtually mark up patients' anatomy using drawing primitives (e.g., line segments) which are displayed on the surgeon's patient-side video-microscope. These augmented reality annotations are registered to the anatomical features and appear to travel with them whenever the microscope view changes. For diagnostics purposes, the surgeon can use intraocular OCT imaging Behaviors. For example, the Visual M-Scan Behavior (see Section 7.10) gives the surgeon the ability to create intraoperative cross-sectional images of the retina by translating the OCT probe across a region of interest. The scan trajectory is displayed on the view of the retina along with the corresponding OCT image. Figure 1.6 shows the types of Behaviors and the example surgical capabilities that they can provide to the surgeon. Many of these Behaviors derive their function from multiple system components as shown in Figure 1.7.



Behavior Type	Component						
	EyeRobot	Micron	OCT Sensor	Force Sensor	Visual Trackers	Video Overlays	Audio Feedback
Robot Assistance	✦	✦					✧
Hybrid Tele-Operation	✧		✧	✧		✧	✧
Audio OCT Feedback	✧	✧	✦			✧	✦
OCT VFs	✦	✦	✦		✧	✧	✧
OCT Scans w/Robot	✦	✦	✦			✦	✧
Force VFs	✦	✦		✦		✧	✧
Audio Force Feedback	✧	✧		✦		✧	✦
M-Scan	✧	✧	✦		✦	✦	✧
Anatomical Annotations			✧	✧	✧	✦	
Telestration	✧				✧	✦	

✦ Essential ✧ Useful

note: robot types are interchangeable in some cases

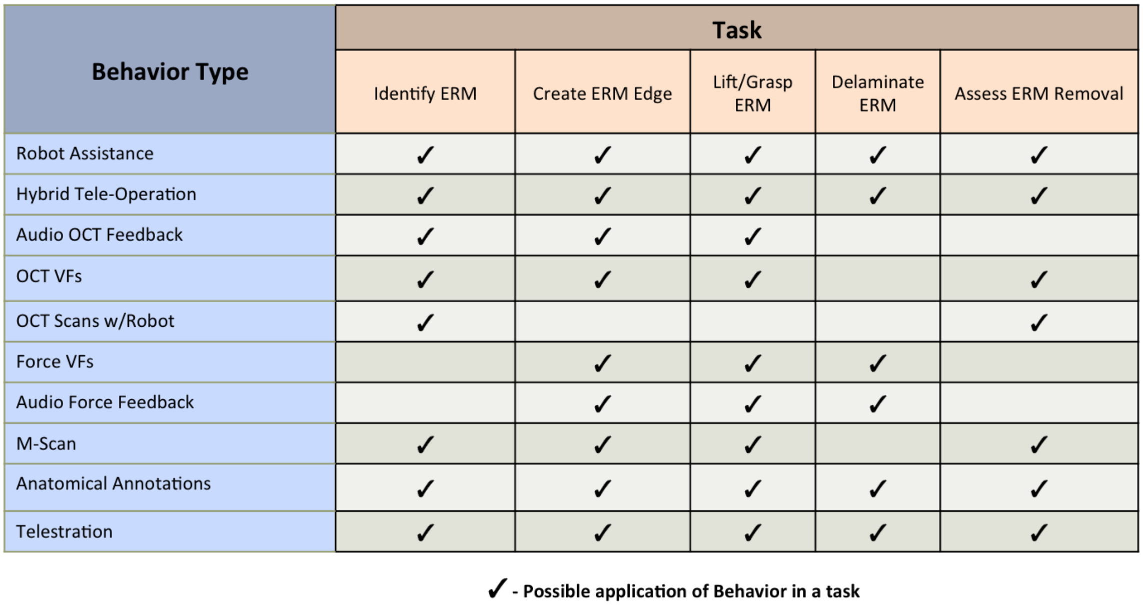
Figure 1.7: Components of the system involved in various Behaviors. Behaviors require specific functionality to operate, but can be enhanced by additional components or Behaviors.

The experience of developing many of these Behaviors and associated hardware and software infrastructure led to the development of a microsurgical system design methodology, which is a systematic way to specify desired surgical capabilities and create relevant technological solutions that consider the surgeon, the patient and the available resources in a complex hardware and software system (see Section 2.2). Furthermore, a complementary software architecture is proposed in Section 2.4 that facilitates the development and real-time management of a human-centric, distributed, and interactive systems. This is especially pertinent when multiple Behaviors are used simultaneously with contextually-modifiable settings based on the task at hand and surgeon preference. An example of this would be movement from one step of a procedure to the next where the system configuration needs to change, which can include enabling/disabling Behaviors, changing Behavior's working parameters such as gains or tool specifications, or "rewiring" the function of a user interface (e.g., pedal input). The proposed architecture framework addresses these challenges.

1.6.1 Use Case - Assisted Membrane Peeling

The general research approach has been to analyze and address the clinical challenges of common but challenging vitreoretinal procedures, one of which is membrane peeling, described in Section 1.5.1. The goal of the procedure is to carefully delaminate a thin fibrous tissue layer from the delicate retina surface by using either a bent blade or a hook. The procedure is associated with high risks of irreversible retinal damage that may result in vision loss. Figure 1.8 shows various Behavior types that can be incorporated into the steps of the membrane peeling workflow. The goal is to provide system assistance while considering the task requirements and surgeon preference.

For example, in the first task of the peeling procedure (after the vitrectomy and any associated preparations), the surgeon needs to identify the ideal ERM peeling location. The assistant



Behavior Type	Task				
	Identify ERM	Create ERM Edge	Lift/Grasp ERM	Delaminate ERM	Assess ERM Removal
Robot Assistance	✓	✓	✓	✓	✓
Hybrid Tele-Operation	✓	✓	✓	✓	✓
Audio OCT Feedback	✓	✓	✓		
OCT VFs	✓	✓	✓		✓
OCT Scans w/Robot	✓				✓
Force VFs		✓	✓	✓	
Audio Force Feedback		✓	✓	✓	
M-Scan	✓	✓	✓		✓
Anatomical Annotations	✓	✓	✓	✓	✓
Telestration	✓	✓	✓	✓	✓

✓ - Possible application of Behavior in a task

Figure 1.8: Behaviors for retinal membrane peeling.

uses a graphical user interface to switch between different task settings, or if available, the surgeon can use a voice recognition Behavior to interact with the system directly. For membrane diagnostics the surgeon can use the Visual M-Scan Behavior (see Section 7.10) to image cross-sections of the retinal structures and inspect them for an ideal candidate peeling edge. This also requires the 3D Video-Microscope Display Behavior (see Section 7.1) that provides direct visualization throughout the whole procedure. To assist with the scans, the surgeon may want to specify the scan location and have the robot perform an automated OCT scan (Automated B-Scan Robot Behavior, see Section 6.5.4), while simultaneously keeping the OCT probe at a constant distance relative the surface to improve the OCT image quality (Surface Tracking Behavior - see Section 6.5.2).

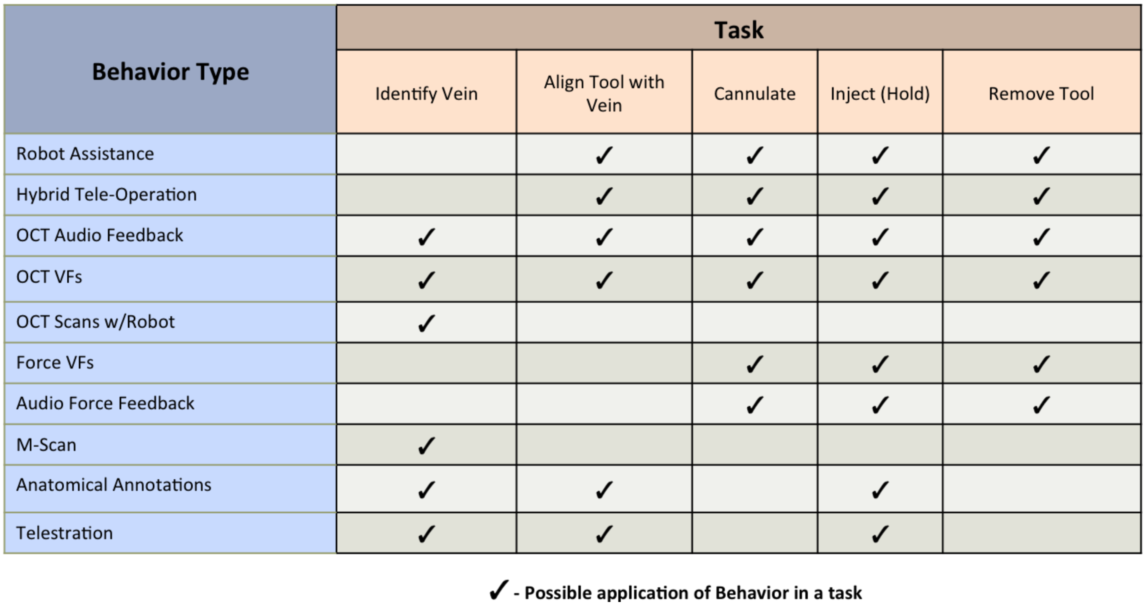
There may be cases where the surgeons needs to create an edge in the membrane. For this step, the surgeon may choose to use an instrument with integrated OCT imaging and to use the

Audio OCT Feedback Behavior (see Section 6.4), which uses the OCT's range-sensing measurements to aurally communicate the distance of the tool to the surface of the retina. Once the target edge is ready, the force-sensing forceps are introduced to assist the surgeon in engaging the membrane. The Force-limiting Behavior (see Section 5.7) is also enabled. It limits the velocity with which the surgeon grasps and lifts the membrane, by using a robot that incorporates the micro-force sensor readings in feedback control. This step can also be supplemented with Audio Force Feedback. The same set of assistance can be used during the delaminating stage. The surgeon may also dynamically lower the robot control gains using a foot pedal to further stabilize the tool motion and minimize the risk of retina tears. Once all the clearly visible membrane sections are removed, the surgeon can reintroduce the OCT instrument into the eye and enable the M-Scan Behavior to assess the retina structures for complete membrane removal. During this procedure, the Logging Behavior, could be enabled to manage the archiving of data generated by the various system components (video, OCT, force, OR audio, etc) and specified by the surgeon prior to, or during the procedure. This data can be used later by the surgeon for educational or research purposes.

1.6.2 Use Case - Retinal Vein Cannulation

The other target procedure considered during development is retinal vein cannulation (RVC), which has the potential to address retinal vein occlusion (see Section 1.5.2). Any surgical procedure carries a risk of complications but RVC is particularly difficult due to the small vessel diameter, the target's proximity to delicate retinal structures, the constraining environment, and the inherent human physiological limitations, such as hand tremor that is on the order of the vessel diameter. It is rarely, if ever, performed by surgeons, because the high risk of trauma to the retinal structures can cause irreversible blindness. Similarly to the membrane peeling use case (see Section

1.6.1), the capabilities provided by the *eyeSAW* system can significantly augment surgical skill, potentially enabling surgeons to perform this difficult procedure successfully. Figure 1.9 shows example Behavior types which can assist in the typical steps of retinal vein cannulation.



Behavior Type	Task				
	Identify Vein	Align Tool with Vein	Cannulate	Inject (Hold)	Remove Tool
Robot Assistance		✓	✓	✓	✓
Hybrid Tele-Operation		✓	✓	✓	✓
OCT Audio Feedback	✓	✓	✓	✓	✓
OCT VFs	✓	✓	✓	✓	✓
OCT Scans w/Robot	✓				
Force VFs			✓	✓	✓
Audio Force Feedback			✓	✓	✓
M-Scan	✓				
Anatomical Annotations	✓	✓		✓	
Telestration	✓	✓		✓	

✓ - Possible application of Behavior in a task

Figure 1.9: Behaviors in retinal vein cannulation.

For example[7], for the identification step, a Behavior based on anatomical annotations concept that incorporates surgical planning and preoperative (diagnostic) imaging information could be used to "pinpoint" the location of the problematic vessel in the microscope view (see Section 7.7). Once the injection instrument (e.g., a pipette) is introduced into the eye with aid from a robotic assistant, the surgeon aligns the pipette with the target vessel. For the cannulation step, the surgeon approaches the vessel with the pipette tip, which is constrained by the robot to move in a specific trajectory relative to the vessel using computer vision-based Virtual Fixtures (VF, see

[7] Some of the Behaviors for cannulation are hypothetical and show how new capabilities could be built with the existing infrastructure.

Section 4.3.4). Simultaneously, a Force-Scaling Behavior (see Section 5.6) could be enabled, which amplifies the vessel manipulation forces. During the semi-autonomous injection step, the robot is visually stabilized relative to the retina, and the Audio Force Feedback (see Section 5.3) is enabled to warn the surgeon about any excessive forces exerted on the tissue. During the removal of the instrument from the vessel, the robot is again in the VF mode to only allow motion parallel to the vessel. During these maneuvers, the surgeon is provided with tool-force information using visual overlays (see Section 7.8). Also, the amount of medication that has been injected is displayed as a numerical overlay near the instrument tip. If a complication arises, the surgeon may request additional Behaviors to be enabled, which provide specific capabilities to address the emergency situation.

1.7 Dissertation Contributions

The dissertation presents a new paradigm for a distributed surgical system for augmenting highly skilled tasks. It addresses multiple surgical challenges by leveraging the power of an integrating software environment to create relevant surgical assistance capabilities by facilitating the exchange of information, and by coupling individual hardware and software functionality. Major lessons learned through the design and development process of the whole system were incorporated into the proposed design, resulting in a generalized method for developing a collaborative human-machine system to assist the surgeon through specific, highly demanding surgical scenarios. Many of the contributions were developed in collaboration with the members of the *eyeBRP* team, and much deserved credit is clearly acknowledged in the contribution section of each chapter. The major contributions reported in this dissertation are summarized here:

- Introduction of a design methodology and a software architecture for a distributed microsurgical system. This architecture facilitates the design and management of an integrated multi-functional system.

- A vitreoretinal surgery system and its demonstration in simulated and in vivo environments. This system includes the following novel hardware:
  - Prototypes of two cooperatively controlled robots (EyeRobot2 and EyeRobot2.1).
  - Force-sensing instruments for measuring force at the tool tip inside the eye.
  - OCT instruments for intraocular imaging and interventions.
- Novel technical capabilities (Behaviors) that can be used to address specific surgical challenges and demonstration of their performance in simulated prototypical surgical procedures, e.g., membrane peeling and vein cannulation. These include:
  - Cooperative robot interaction methods: Pedal Gain Control, Pseudo RCM, Bimanual RCM, Automatic Tool Retraction, and Virtual Joystick.
  - OCT-based control with the EyeRobot: Safety Barrier, Surface Tracking, Servo-To-Target, and Automatic B-Scans.
  - Force-based control with the EyeRobot: Force limiting, Force-guided Peeling, Tissue Characterization, and Tool Compliance Correction.
  - Audio sensory substitution to present micro-forces and OCT-based range sensing information.
  - Volumetric OCT scanning with cooperatively-controlled EyeRobots and also with hand-held Micron robots.
  - Visual tool-to-tissue proximity detection based on video-microscopy.

  - OCT-based intraocular navigation that combines visual overlays, OCT imaging and visual tracking of the tools and the retina.
  - Multi-view tele-visualization for eye surgery.
  - Augmented reality with visual overlays to display real-time intraocular sensor information and methods to visually annotate the retina structures during an operation (telestration).

- A cooperative teleoperation prototype for vitreoretinal surgery that combines the EyeRobot and the da Vinci master console.
- Bimanual cooperative control with two EyeRobots demonstrated on a human eye phantom.
- Artificial phantoms and testbeds for development, demonstration and evaluation of the overall system, and task-specific functionality.
- A software framework and application to facilitate analysis of voluminous multi-media data collected during subject and rabbit experiments.
- A novel automatic OCT spectrometer calibration method using the EyeRobot.
- A video-microscope latency testing device.
- Basic science experiments to assess human depth perception in video-microscopy, vessel cannulation forces, and membrane peeling forces.

1.8 Outline

This chapter has presented the motivation for this work, and introduced the approach to addressing the challenges in vitreoretinal surgery. Chapter 2 presents a philosophy in designing a complex microsurgery system, and demonstrates a software architecture and design paradigm that facilitates implementations of a fully integrated distributed surgical system. Chapter 3 presents two experimental system setups and the variety of testbeds that were developed for development, demonstration, and validations of the resulting system. Chapter 4 presents the two robotic platforms used in manipulation assistance, along with control methods developed for microsurgery specific behaviors. The following chapters 5 (Force), 6 (OCT) describe surgical assistance methods based on feedback from robot-compatible smart surgical instruments with embedded micro-sensors. Chapter 7 presents the visualization system. These subsystems are integrated into a distributed system and used to augment surgical skill by providing aural, visual, and haptic real-time feedback for specific tasks. Each chapter includes relevant experiments and discussions which are summarized in the conclusion chapter 8.

Chapter 2

Integrated Microsurgery System

Effective surgical care requires expert human judgment and fine motor skill to handle the diversity and complexity of human anatomy and physiology, while considering the particular aspects of the disease. In conventional practice, surgeons perform highly dexterous maneuvers in extremely restricted cavities, often with suboptimal feedback. The localization of target sites such as tumors or lesions is difficult due to their inherent location deep within the tissue, or obstruction by the geometry of the anatomical structures, especially in minimally invasive surgery (MIS). The targets are often only visible in pre-operative images (e.g., MRI, CT, Ultrasound, OCT, etc.), requiring the surgeon to virtually collocate the pre-operative information with barely visible anatomical landmarks in situ which is a very difficult task. Conventional surgical instruments are designed as a compromise between desirable function, manual dexterity, and requisite accessibility to the surgical site. These instruments inherently tend to magnify the effects of natural hand-tremor. Furthermore, they obfuscate natural tactile sensations, which are traditionally used in diagnostics and navigation (e.g., palpation). The surgeons are also affected by poor ergonomics, which results in increased fatigue and potential occupational disability, and early retirement. The surgeons face these barriers while simultaneously interacting with the operating room staff and closely monitoring the progress of the surgery. Overall physical and cognitive load on the surgeon is great. As described in the first chapter, all these challenges are especially acute in vitreoretinal surgery.

In the last two decades, the advancements in robotics, computing, and sensing technologies have allowed researchers to develop computer-integrated systems to address many of the above challenges to make existing procedures more accurate, faster, or less invasive; and also by enabling otherwise infeasible interventions. Most systems have focused on navigation to improve visualization and targeting, and robotic assistants to improve precision and reduce invasiveness. In microsurgery, the surgeons are required to overcome numerous clinical and technical challenges and a single device can only aid in small subset of these in a given task. Through the development of assistive technologies for vitreoretinal surgery as part of the *eyeBRP* it became apparent that a systems approach to create a fully integrated surgical assistant workstation (SAW) to augment many facets of surgical skill in a variety of task is necessary. Such system combines human expertise with robotics, sensors, haptic, aural and visual displays, and incorporates information and computing to fundamentally enhance surgical performance. The result is a partnership that transcends human limitations and creates super-surgeons to perform highly skilled tasks.

This chapter introduces the design methodology of a microsurgical system, associated engineering challenges, and proposes a system architecture that facilitates the design and management of an integrated multi-functional surgical system.

2.1 Surgical Assistance Systems

One of the earliest robotic systems is the ROBODOC [40] which is designed for orthopedic implant surgery, see Figure 2.1A. The system includes pre-operative planning where the surgeon interactively defines the location of an implant in a 3D CT model of the patient's femur. This plan is then registered with the patient and the robot autonomously drives a pneumatic or electric drill to mill a precise cavity for a hip or knee implant. In this paradigm, the surgeon relinquishes hands-on tool control for significant improvement in bone surface and precise location of the surgical cut. A more recent and widely commercialized da Vinci robotic surgery system [1] is designed for soft tissue surgery and provides no autonomous behavior. It is a bi-manual tele-robotic platform where a surgeon sits at a console, away from the patient, and manipulates two robotic "joysticks". The patient side robotic manipulators have dexterous end-effectors that are positioned inside the body cavity and mimic the exact motions of the surgeon. The surgeon console displays endoscopic stereo video and is the main form of feedback during the operation. This platform offers good ergonomics, intuitive control, improved in situ dexterity and reduces hand-tremor effects from motion scaling

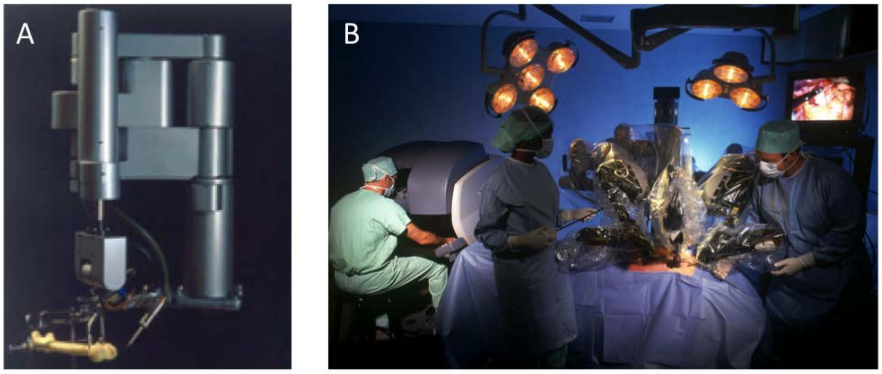


Figure 2.1: Early surgical robots: A) Robodoc system (Credit: www.robodoc.com), B) da Vinci surgical system (Credit: www.intuitivesurgical.com).

and filtering. For these benefits the surgeons trade-off the conventional hands-on operation and require a scrubbed-in backup surgeon in case of emergencies.

These examples of MIS robotic systems have shown successful assistance in surgical practice on the millimeter scale, but are not designed to cover the challenges of microsurgical procedures that often involve manipulation of tissue on the scale of tens-of-microns. In the case of vitreoretinal surgery, there has been very little surgical system development that actively augments the skills required to perform multi-step vitreoretinal procedures. Most research projects focus on basic tool-manipulation assistance in the μm to mm scale using various robot paradigms described in Section 4.1. Other groups worked on visualization, diagnostic imaging, and force sensing but there has not been any significant development that combines these technologies in a unified, integrated system. Currently, the most pertinent and widely used clinical vitreoretinal surgery system is a vitrectomy machine that provides independent surgical functions including a vitrectomy cutter, laser ablation, illumination, vacuum, infusion, etc. Figure 2.2 shows a popular vitrectomy system made by Alcon, Inc. Although essential, these are mostly passive, mechanical tools without real-time sensing

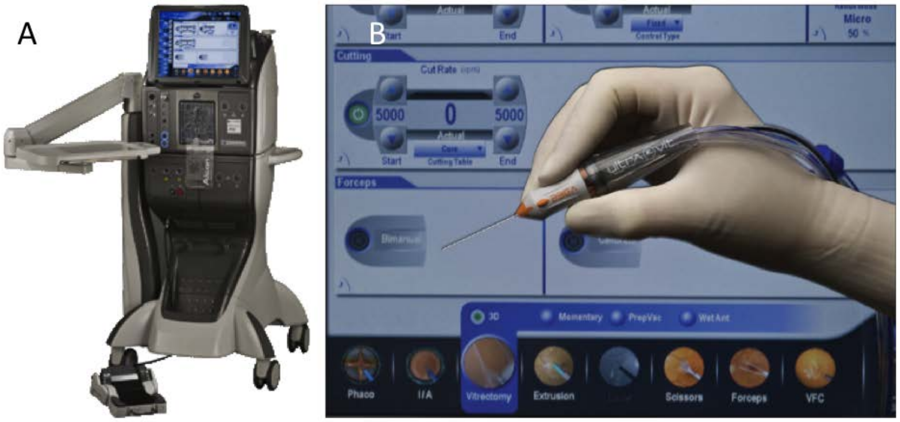


Figure 2.2: A) Constellation vitrectomy system (Credit: www.alconsurgical.com). B) Vitreous-cutting probe [41].

or built-in intelligence that adapts to the particular needs of a surgical scenario. The vitrectomy systems provide minimal, or no feedback to the surgeon during the procedure beyond the state of the device settings or errors. The surgeons may adjust some of the tools' functional parameters on-the-fly (e.g., cutter speed) with foot pedals, but many of the adjustments are communicated verbally to an assistant who enters them into the system using a touch panel. The surgeon needs to constantly monitor the instruments' parameters (e.g., infusion pressure) and adjust each one independently for a given task. This additional responsibility adds to the cognitive load of the surgeon, who is already under severe pressure to make no mistakes.

2.2 System Design Methodology

Surgical assistance systems should be integral components of a tightly integrated health-care process that is individualized for the patient. According to Taylor et al. [42], a surgical assistance system is designed to assist a surgeon in carrying out a surgical procedure. The system may include preoperative planning, intraoperative registration to pre-surgical plans, use a combination of robotic assistants and intraoperative sensors, and manual tools for carrying out the plan, and postoperative verification and follow-up. Although all of these phases are important, the work presented here focuses on the intraoperative aspect of the surgical assistance system and considers future expansion that includes preoperative planning, post-operative data analysis, etc. The overall goal is to enhance the effectiveness and safety of a procedure by coupling technology and information to action in the operating room or interventional suite [43].

The *eyeSAW* system development followed two parallel efforts. One is the development of software infrastructure to enable rapid development of new capabilities through standardized software interfaces and communication methods between various software modules. The other is the

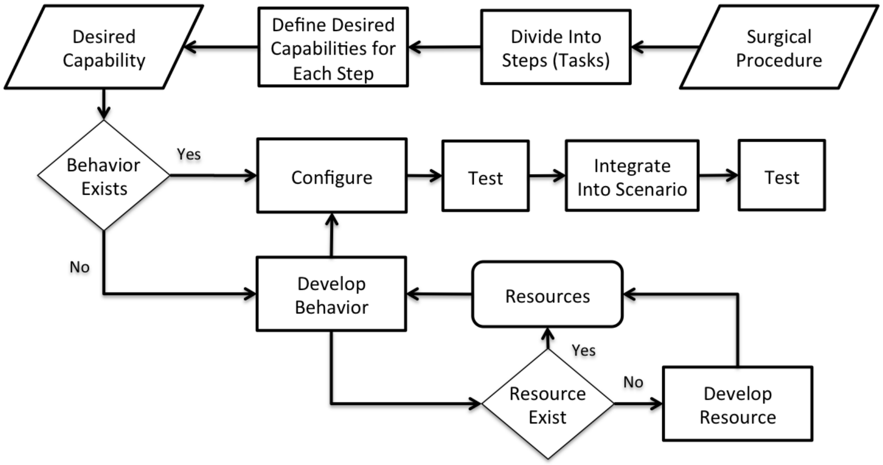


Figure 2.3: Design methodology for surgical system capabilities.

general system design methodology[1] for surgical capabilities, which is shown in Figure 2.3. The process starts with choosing a target procedure or a treatment. This procedure could be experimental, showing promising results but has high associated risk, or lacks adequate instrumentation. Other candidate procedures could improve efficiency to reduce intraocular operation time, or involve many repetitive tasks that could benefit from automation. Initially, the target surgical procedure (also called a *Scenario* here) is decomposed into discrete steps (also called *Tasks*), see Membrane Peeling in Section 1.5.1 for an example. These steps are further analyzed for technical challenges faced by the surgeon to extract the required surgical capabilities, and also the desired capabilities that could augment the surgeon's performance, and generally address the technical and surgical limitations described in Sections 1.5.3. These define the requirements that a system designer uses to develop a new *Behavior*. Behavior is a well-defined capability provided by the system specifically designed to assist the surgeon in a Task. Some of these Behaviors may already be found in the system, or,

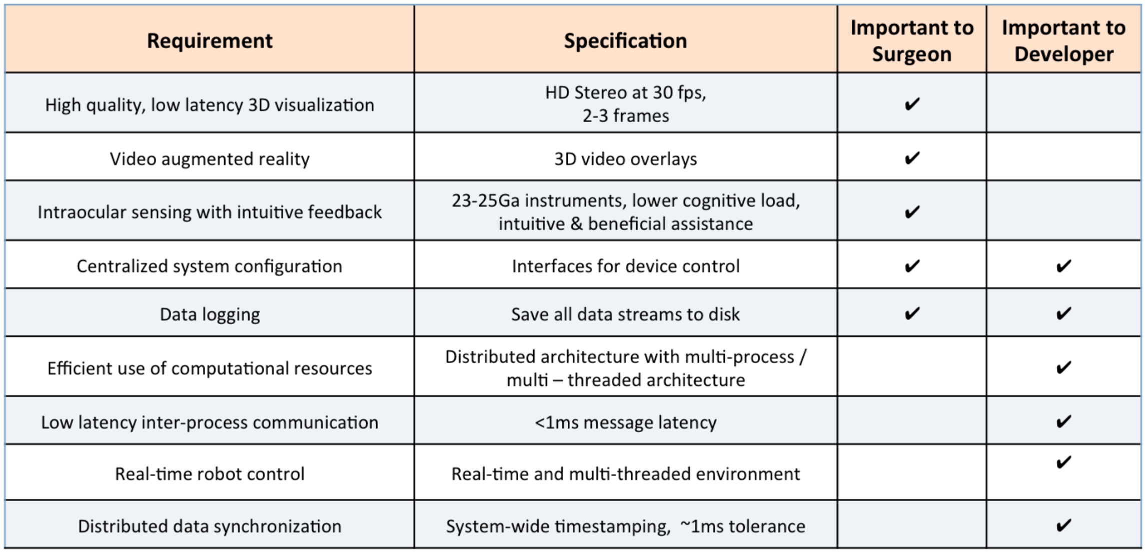
[1] Initial idea for this approach is inspired by the *eyeBRP* proposal by Russell H. Taylor, et. al.

as in many cases, need to be developed from the available resources. The system *Resources* include various hardware technologies such as robotic manipulators, imaging sensors, as well as software algorithms that process data. Each of these *Resources* is encapsulated using a Component software design pattern, discussed later in the Chapter, which exposes the *Resource's* specific functionality (control, retrieve data, etc.) using a standard software interface. Sometimes, new technologies need to be developed to provide functionality, such as force sensing, that is then incorporated into a Behavior, e.g., audio sensory substitutions (see Section 5.3). Each Behavior in a task is configured to suit the surgeon's preference and the task requirements. It is then tested independently and incorporated into the surgical workflow alongside other Behaviors. The whole system is then tested to check for any adverse effects of the newly introduced Behavior. As with any mission-critical design, multi-level iterative design and testing are required.

The process of design, deployment and operation of complex smart instruments to meet the clinical objectives has to be holistic and include a balanced view of the clinical and engineering (hardware and software) components in the overall solution. In this case the process heavily engages specialists from different disciplines, but most importantly it requires the participation of surgeons with focus on surgical assistance to improve the outcomes, at least in the long run, i.e., the surgeons are part of the process all the way through the design, development and testing.

2.2.1 System Engineering Challenges

A designer of a surgical system is faced with common operating room (OR) challenges such as safety, robustness, efficacy, precision, sterilization, ergonomics, and OR workflow integration. In the case of vitreoretinal surgery, there exist additional challenging aspects, including device miniaturization, extremely high sensor sensitivity requirements, high risks associated with tissue contact, unusual work volume, poor visualization, and an aqueous environment. Surgeons require minimal latency, high quality visual feedback; easy and accessible system interfaces, and intuitive communication of real-time sensor information, such as visual or audio sensory substitution. These are driving design factors but do not necessarily overlap with the system developers' considerations which include: requirements of existing hardware, computational efficiency to process video or sensor data in real-time, communication and distribution of computation over many machines, and ease of development of new functionality. Figure 2.4 summarizes a set of example factors to be considered by a system engineer. Some of these are also pertinent to software engineers and are explained further in the following section.



Requirement	Specification	Important to Surgeon	Important to Developer
High quality, low latency 3D visualization	HD Stereo at 30 fps, 2-3 frames	✔	
Video augmented reality	3D video overlays	✔	
Intraocular sensing with intuitive feedback	23-25Ga instruments, lower cognitive load, intuitive & beneficial assistance	✔	
Centralized system configuration	Interfaces for device control	✔	✔
Data logging	Save all data streams to disk	✔	✔
Efficient use of computational resources	Distributed architecture with multi-process / multi – threaded architecture		✔
Low latency inter-process communication	<1ms message latency		✔
Real-time robot control	Real-time and multi-threaded environment		✔
Distributed data synchronization	System-wide timestamping, ~1ms tolerance		✔

Figure 2.4: Example considerations and resulting specifications used in the system engineering design processes of the *eyeSAW* system.

Furthermore, validation of the system functionality is extremely difficult due to the very small scale (1 μm – 1 mm) of the anatomy and the associated surgical maneuvers, and to the lack of adequate metrology instruments.

The *eyeSAW* system has evolved to include many subsystems or independent modules which are shown in Figure 2.5. These include robots for manipulation assistance; real-time sensing such as OCT imaging and force sensing; a surgical visualization console that captures, manipulates, and displays 3D stereo video from the surgical microscope; a technician console application that is responsible for monitoring and configuring the whole system; hardware interfaces such as pedals, audio output, external data storage; and various software modules such as computer vision trackers, data loggers, etc.

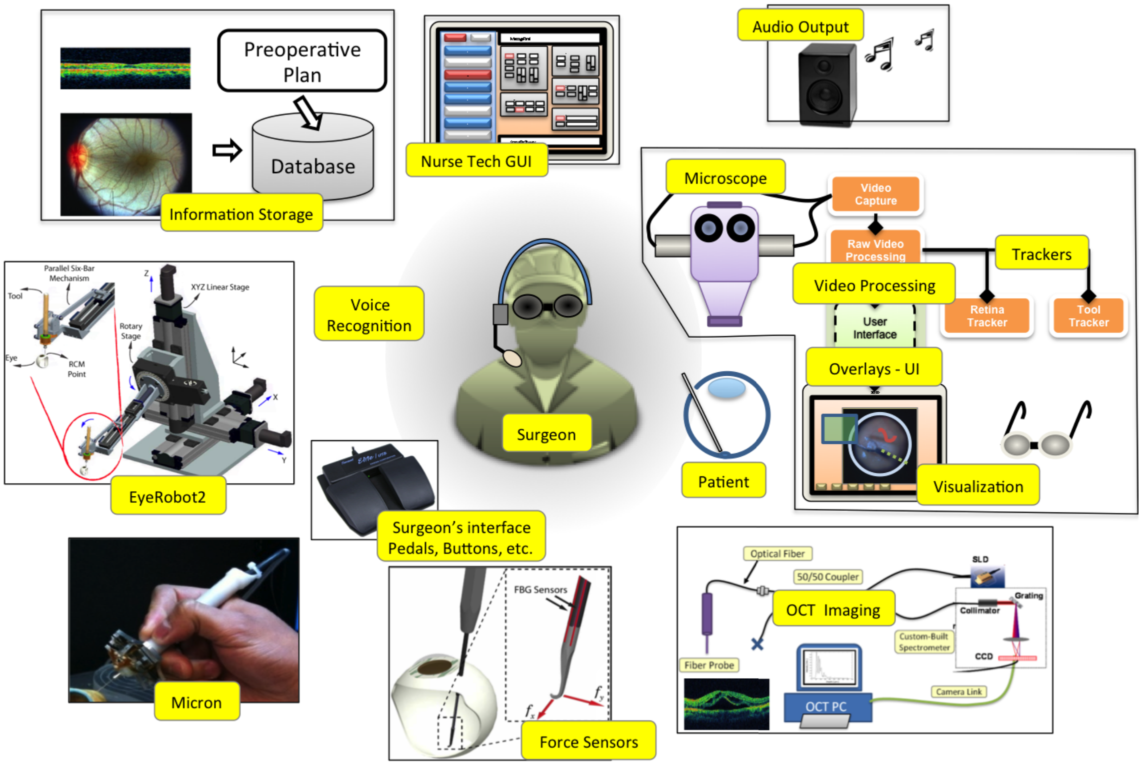


Figure 2.5: Various hardware and software modules in *eyeSAW*.

Since many of the surgical and environmental challenges can simultaneously affect surgical performance, a system that provides a wide range of capabilities concurrently such that they can be combined into a modular, synergistic, and extendable system is desirable. Additionally, this enables computer-interfaced technology and information processing to work in partnership with the surgeon to provide an intelligent and refined real-time assistance. The various components can leverage their functionality and information to provide new capabilities. However, the complexity of the system caused by such a large number of hardware and software modules and their interconnections that need to be designed and managed during operation poses additional system design challenges.

The *eyeSAW* system is built with a mix of custom and off-the-shelf devices that are often located on dedicated computers due to hardware (e.g., high performance graphics card) and software dependencies (e.g., device driver availability), cost, specific processing needs, different operating systems, and also for convenience and portability reasons. The system needs to be easily maintainable, which can be facilitated by modularity where each subsystem's basic functionality can be tested and verified independently of the whole *eyeSAW* system. Also, the physical robustness and size should be sufficient as to fit in a standard vehicle and be transported without much effort.

For example, the OCT is a self-contained module that acquires raw data using a special acquisition data board, then processes and generates large (hundreds of MBytes) 2D and 3D OCT images in real time. The OCT module is developed and tested in another lab, where it is used for many purposes such as basic OCT device research and other surgical applications. The surgical microscope visualization runs on a separate machine and captures video using a Firewire interface, processes up a half a gigabyte of video data (30+ FPS), displays it, and saves it disk, also in real-time. Although it would be feasible to build a single machine executing a single application that combines OCT processing and the visualization pipeline, the contention for processor resources and bus bandwidth would significantly affect video frame latency and frame rate, while degrading real-time OCT performance.

Since the*eyeSAW* system is inherently a human-machine collaborative system, its user interfaces are especially important. Surgeons are already interacting with many devices in the OR via multiple pedals and touch screens. While they are comfortable with these interfaces, introducing more of these devices should be kept to a minimum. Furthermore, the ergonomics and intuitive nature of these interfaces plays a major role in adoption of the system, especially since surgeons have only a limited time for training and are already performing difficult tasks under significant cognitive load. The surgeon's attention is also divided between the task, interaction with the OR team, and feedback from the devices in the OR, mostly in the form of audio beeps. It is essential to intuitively communicate system information, not only to the surgeon, but to other members of the surgical team. Of course, an overload of such information can be detrimental, so the design of the feedback systems should consider the current context, i.e., the state of surgical procedure and the state of the system.

2.2.2 Software Engineering Challenges

Architectural complexity, design flexibility, run-time system management, high performance, and safety are challenges among many that need to be considered when developing software for a surgical system with many interoperable components. These systems require the integration of an increasing number of devices, including robots, displays, sensors, tracking devices, user interfaces, etc. The resulting system complexity makes applications challenging to develop, more difficult to debug and the resulting accumulation of ad hoc interfaces reduces the overall portability.

The software infrastructure should provide a basic framework that allows the system developers to quickly implement surgically relevant application prototypes for which they may need various components such as real-time imaging, robot control, registration algorithms, etc. It also needs to provide standard methods to facilitate interoperability and exchange of information between the software components, whether they are located in the same or different threads of execution, processes, or on remote computing platforms. One example is real-time robot control that incorporates various sensor data (e.g., force, or OCT range) that is generated by hardware devices located on remote computers. Variability in software APIs, hardware requirements and software dependencies, such as a particular operating system or a computer type, are a real issue and require well defined software interfaces for creating device wrappers to ensure smooth integration into the system. These are particularly challenging in the cases where subsystem modularity and configuration capability for similar devices are highly desirable, e.g., effortlessly swapping a robotic manipulator for a similar one in an existing control algorithm.

The performance considerations in real-time feedback systems are significant, hence the data exchange between the system components should be low latency, and use minimal CPU processing and network transport resources to minimize resource contention with other modules. For example, the robot control module needs to collect feedback force measurements and OCT range data from other processes that may be located on other machines. These signals are remotely preprocessed, significantly reduced in size and then sent to the robot control machine. Therefore, whenever possible, the processing modules and the data should be located "near" each other to avoid excessive copies or serialization. This is very challenging considering that data generated by a single hardware device may need to be accessed by multiple CPUs (e.g., video stream) to be processed as fast as possible, while simultaneously be available to other modules in the system, possibly located on other machines.

The collection of the data generated by various subsystems is very important for experimentation and especially pertinent in surgical systems that are used in animal or human clinical experiments. These experiments are often performed under severe time constraints due to anesthesia and possible rapid degradation of tissue quality, e.g., optical and mechanical properties of the retina. The experiments are costly requiring a large support staff, specialized equipment, and biological samples. Furthermore, it is imperative to minimize the number of experiments that require animals which are often sacrificed after each experiment. This limits the number of possible experiments and increases the need and reliability requirements of the data collection methods.

Simultaneously, data collection should not impede the performance of the vital system components that produce the data itself, and also consider the performance of other components. For example, a typical servo-level robot control Component processes small data of a few tens of bytes at high frequencies in the kHz range; whereas a real-time imaging module with vision processing Components may handle large data, such as a couple hundred MB/sec, at relatively low frequencies (e.g., 20-30 FPS). The types of data produced by the various system Components is quite different, e.g., video images, scalar forces, robot kinematics, volumetric OCT, etc. These need to be time-synchronized across the system and stored in a simple format for future analysis. Collecting such diverse data efficiently at run-time is not trivial and becomes more challenging with the increasing number of interacting components, and custom data types. Data collection functionality needs to be relatively easy to implement for the developer and be centrally controlled during the experiments. Additionally, the large volumes of multi-media datasets, many of which have proprietary formats, require a simple and intuitive tool for post-experiment analysis. This tool should be easily extensible to accept new data types.

There is continuous demand for changes, enhancements, and new features that the developers and surgeons find necessary once the system is established. It may be due to unforeseen ways to use the system for existing procedures, or for new procedures that require new capabilities from

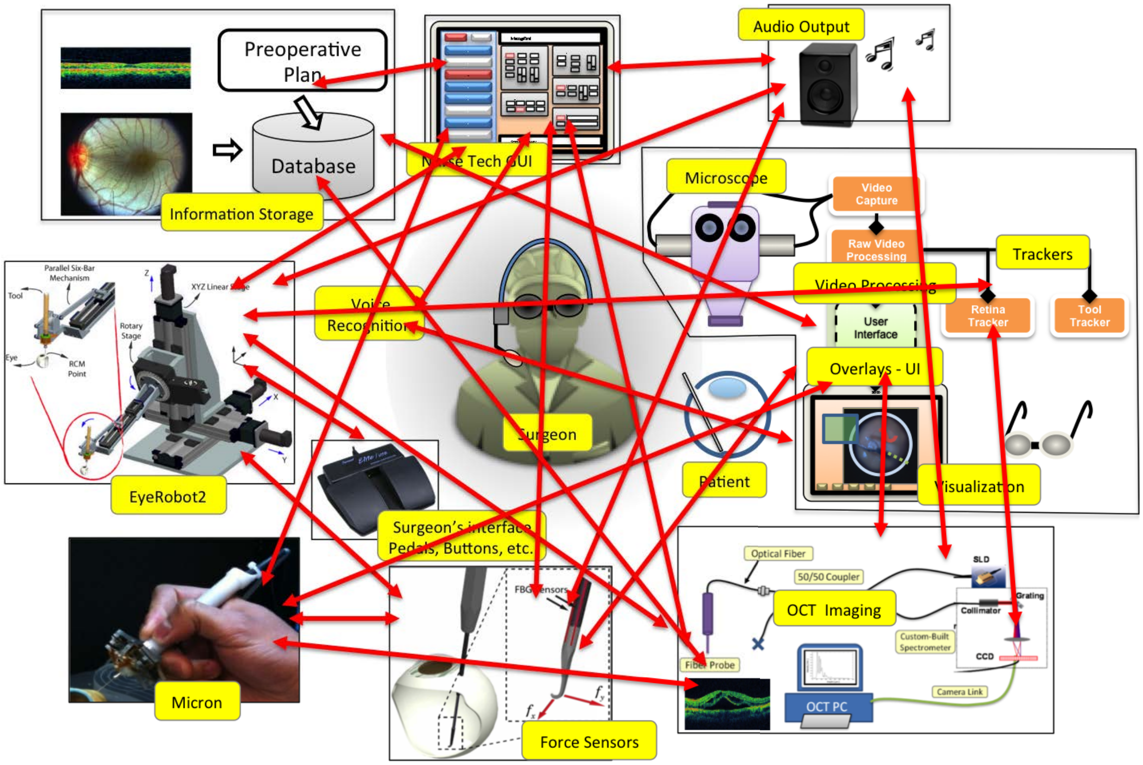
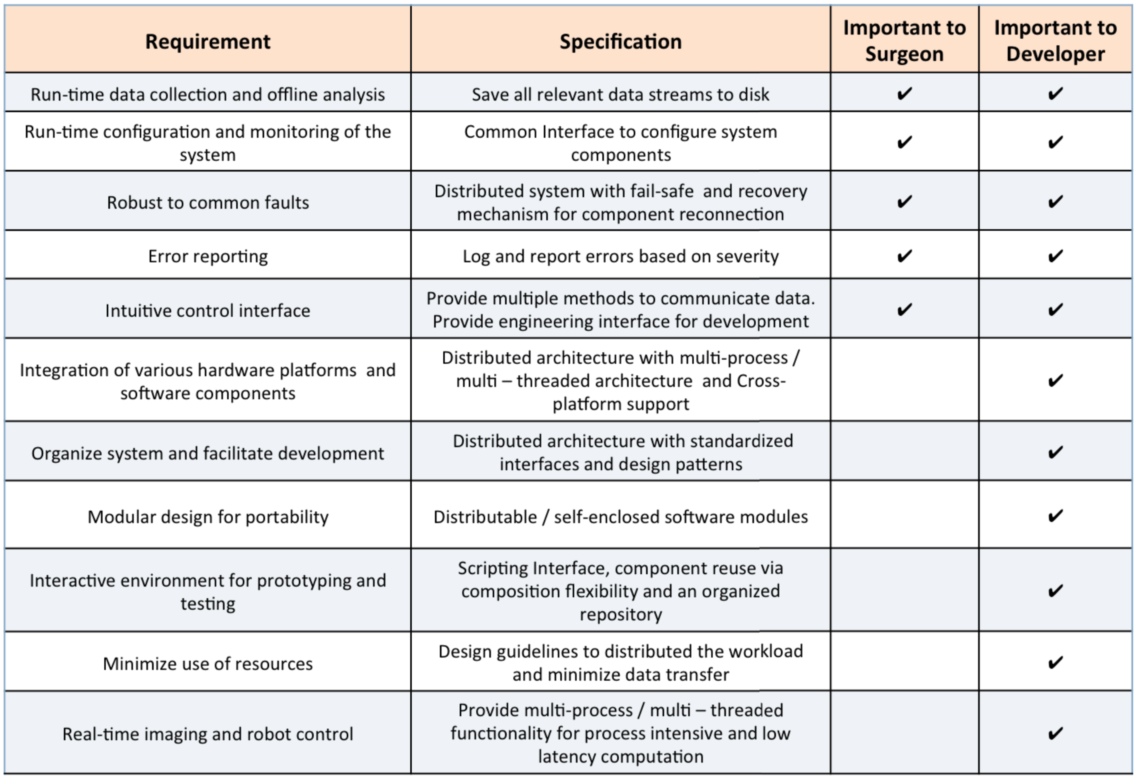


Figure 2.6: Web of interconnections in the *eyeSAW* system.

the system. For this, the reuse of the software and hardware is important. One approach is to create many independent modules that function on their own and can be "wired" together by the developer. Although this solves many of the hardware and software dependencies, it creates many new software engineering challenges, one of which is the natural complexity of a highly interconnected and modular system. The complexity of the *EyeSAW* system is clearly evident in Figure 2.6, which includes a large number of independent components and many interconnections that make applications difficult to develop and maintain. Besides the need for an inter-component communication software infrastructure, there is a great need for an architecture abstraction for assembling the system together so it can be developed efficiently and run as expected. This architecture needs to prevent overly complicated implementations and minimize human design errors that are often found in situations where inter-dependability of multiple components is required.

Many of the components in the system require individual configuration, as well as specific configuration of their interconnections. Often these are dynamic configurations, based on the context, and require a centralized management scheme. An associated challenge is ensuring system integrity during runtime, i.e., the system cannot be in a conflicting configuration. Finally, the proposed system includes many components working independently and together, in cooperation with a human, therefore intuitive user interfaces should be a priority, considering traditional (graphical) and also alternative (audio, haptic) user interfaces.

Figure 2.7 shows example considerations and resulting specifications for developing the *eyeSAW* surgical system.



Requirement	Specification	Important to Surgeon	Important to Developer
Run-time data collection and offline analysis	Save all relevant data streams to disk	✔	✔
Run-time configuration and monitoring of the system	Common Interface to configure system components	✔	✔
Robust to common faults	Distributed system with fail-safe and recovery mechanism for component reconnection	✔	✔
Error reporting	Log and report errors based on severity	✔	✔
Intuitive control interface	Provide multiple methods to communicate data. Provide engineering interface for development	✔	✔
Integration of various hardware platforms and software components	Distributed architecture with multi-process / multi – threaded architecture and Cross-platform support		✔
Organize system and facilitate development	Distributed architecture with standardized interfaces and design patterns		✔
Modular design for portability	Distributable / self-enclosed software modules		✔
Interactive environment for prototyping and testing	Scripting Interface, component reuse via composition flexibility and an organized repository		✔
Minimize use of resources	Design guidelines to distributed the workload and minimize data transfer		✔
Real-time imaging and robot control	Provide multi-process / multi – threaded functionality for process intensive and low latency computation		✔

Figure 2.7: Example considerations and resulting specifications used in the software engineering design processes.

2.3 Background

Many of the existing surgical assistance platforms are similar to the da Vinci tele-surgery system [1], which is inherently a closed system[2] that is structured for a single purpose with a very limited set of capabilities, mostly in improving manipulation precision in difficult to access regions of the body. Although the da Vinci has many hardware components: slave arms, master controller arms, buttons, pedals, etc.; the actual behavior of the system during a typical procedure does not change very much. The operating modes include the adjustment of endoscope position, clutching

[2] There is some development to open up limited da Vinci robot control functionality for research purposes [44].

to change master position, enable/disable arm, etc. In other words, the da Vinci provides a basic tele-surgery paradigm (including visualization) with some additional customization for instrument swapping. Furthermore, the da Vinci, and most of the surgical systems, include a computer in the loop between the surgeon and the patient, which can facilitate the incorporation of additional information to augment the surgeon's abilities to carry out the desired surgical task. The open-source *cisst* Surgical Assistant Workstation (*cisst-SAW*) framework presented by Kazanzides and Taylor et al. [45–47] is a good example of an effort towards this goal.

2.3.1 *cisst-SAW* Framework

The *cisst-SAW* framework [48] is a set of software libraries developed at The Johns Hopkins University to address some of the problems encountered when integrating devices for computer assisted interventions (CAI). The *cisst* libraries include basic foundation classes (data types such as vectors, matrices, transformations, and tools such as class and object registries, logging, etc.). It also contains the *cisstMultiTask (MTS)* library, which is a component-based software framework for cross-platform, multi-threaded, multi-process systems. It supports different execution models, such as periodic threads, callbacks, and event-based programming. MTS adopts the Component-based Software Engineering (CBSE) [49, 50] principles to 1) facilitate the development of software from pre-produced parts, 2) improve the ability to reuse those parts in other applications, and 3) simplify maintenance and customization of those parts to produce new functions and features. CBSE is analogous to electronic design, where integrated circuit components with standard interfaces (input/output pins) and well defined functionality are connected together to create a complex system. In MTS, a *Component* is a class object that encapsulates some services (particular functionality); it can run on its own thread; and has a *Provided Interface* that can be connected with a *Required*

Interface from another *Component*. This connection mechanism for data exchange can be dynamically executed at run-time from anywhere in the system, where the individual components can reside on the same or different process and/or machine.

The *SAW* architecture provides component-based software modules specifically to support rapid prototyping of medical robotics and computer-assisted surgery systems. It is based on earlier work by the same group at The Johns Hopkins University that looked into methods for distributing data objects in distributed modular computer-integrated surgical robot systems (Schorr et al. [51]); as well as associated software module organization and basic network architectures (Bzostek et al. [52]).

A key aspect is the definition of interface standards within the *SAW* framework, that enable "plug-and-play" configuration of systems. It differs from other component based frameworks in its intended application and in the extent of surgically relevant functionality and device interfaces that it provides. In fact, *SAW* is underlying technology enabling the development of the system presented in this dissertation. However, it does not provide a convenient application level architecture framework or guidelines to design and manage the behavior of the resulting system.

2.3.2 Related Frameworks

Aside from *cisst-SAW*, the other well-known open source efforts to facilitate the development of CAI systems include 3-D Slicer [53], Image-Guided Surgery Toolkit (IGSTK) [54], and Medical Imaging Interaction Toolkit (MITK) [55]. These are very similar toolkits and primarily focus on medical image visualization and processing for intraoperative image-guided navigation. The closest work to concepts developed for the *eyeSAW* system is the IGSTK. IGSTK consists of libraries for reading, visualizing, and interacting with medical images, and supports several tracking systems. Most applications built using IGSTK are small in comparison to *eyeSAW* and generally involve a medical imager, and/or positional tracker, display and in some cases a robotic manipulator. Typically, these applications have a static overall system configuration and often provide only a single navigation aid, and/or robotically assist in precision targeting. Additionally, IGSTK has a component-based architecture and uses a state machine software design methodology to improve the reliability and safety of the individual system components [56].

Despite being a useful tool, the IGSTK does not provide a wide infrastructure support that is required for a distributed surgical system such as *eyeSAW*. This includes efficient and flexible interprocess communication, built-in device support for a variety of robots and sensors, stereo-video processing and visualization, and more importantly a convenient, high-level monitoring and control of a large set of distributed system components that are used for a variety of tasks.

In the robotics domain, a variety of component-based software packages and middle-wares have been developed and released to facilitate robotics research, see the survey by Elkady and Sobh [57]. The two most widely used packages are Open Robot Control Software (Orocos) [58] and a newer framework called Robot Operating System (ROS) [59]. Although these could be used to implement the *eyeSAW* system, they are not designed with medical applications in mind. Orocos is a set of low-level C++ libraries for machine and robot control and consists of the Kinematics and Dynamics Library (KDL), the Bayesian Filtering Library (BFL), and the Orocos Real-Time Toolkit (RTT). ROS provides software modules for performing common, but higher-level, robotics functions such as motion planning, object recognition, and physical manipulation. It also includes hardware abstraction interfaces, many device drivers, wrappers for external software libraries (e.g., OpenCV [60]), visualizers, package management, simulators, and a convenient message-passing scheme. Similarly to *cisst-SAW*, ROS enables any number of independent modules (components) to run at a given time. These modules can be connected for testing, disconnected for debugging, and reinstated without destabilizing the distributed system as a whole.

2.4 System Architecture

The system architecture design evolved to address the surgical system needs and software engineering challenges (as described Sections 2.2.1 – 2.2.2) that have been encountered during the development of the *eyeSAW* system. The design is specific to a context where a human or a small team of humans require computer-integrated assistance to perform a set of difficult tasks that involve an expert level of skill. This assistance can be in the form of information presented visually, aurally, or haptically, and direct or indirect manipulation enhancement. The system not only has to cooperate smoothly with the surgeon to provide these capabilities, but also internally, the system's components need to exchange information and collaborate functionally, as shown in Figure 2.8. The creation of these interconnections grows in complexity with each additional capability provided by the system and the resulting topology changes throughout the procedure to address the different needs of the surgeon and the requirements of the surgical task.

A general surgical system design methodology was described in Section 2.2. It proposes a systematic way to design surgical system capabilities that are derived from combining functionality from various software and hardware *Resources* available to the developers. A *Resource* is the smallest unit of composition in the system, and is a Component as defined by CBSE design principles [49] with each interface providing a well defined set of functions.

Furthermore, the terminology (*Behavior, Task, Scenario*) used in the design process is directly translated into structural and functional elements of the system architecture. Each significant capability is then realized through a proposed software architecture design pattern called

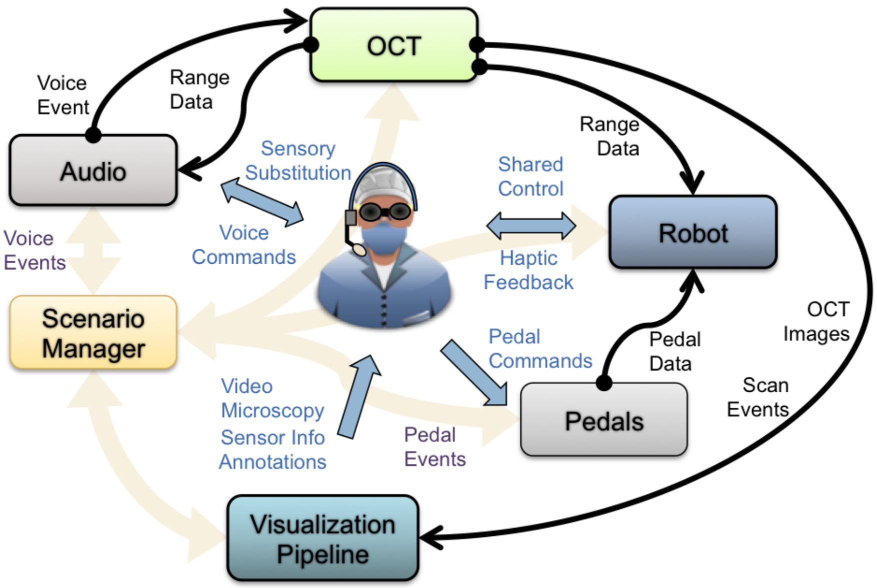


Figure 2.8: Information flow in a simplified *eyeSAW* system.

the *Behavior*[3]. Behavior is defined as a group of Resources, their configuration, and some specific logic encapsulated in a software component that makes them work together to provide a specific capability in a surgical task. These Behaviors are considered the building blocks of the surgical application. The main idea is to break down the application level design and run-time control of the system into manageable pieces (Behaviors) that can be combined, reused, and reconfigured (via substitution) to compose a functional system application (see principles of CBSE [49]). Furthermore, it is much easier to thoroughly test individual Behaviors with a minimal set of capabilities than to test medium or larger monolithic applications with a multitude of features and convoluted interfaces. This approach aligns with the main concept of CBSE where isolation of function can lead to a more manageable large scale distributed system development. However, this only works

[3] Note: there are many uses of the term “Behavior” in the software engineering literature. It was adopted here purely for intuitively conveying the concept of capabilities provided by the system to the users.

in a well defined software architecture, as proposed in this section.

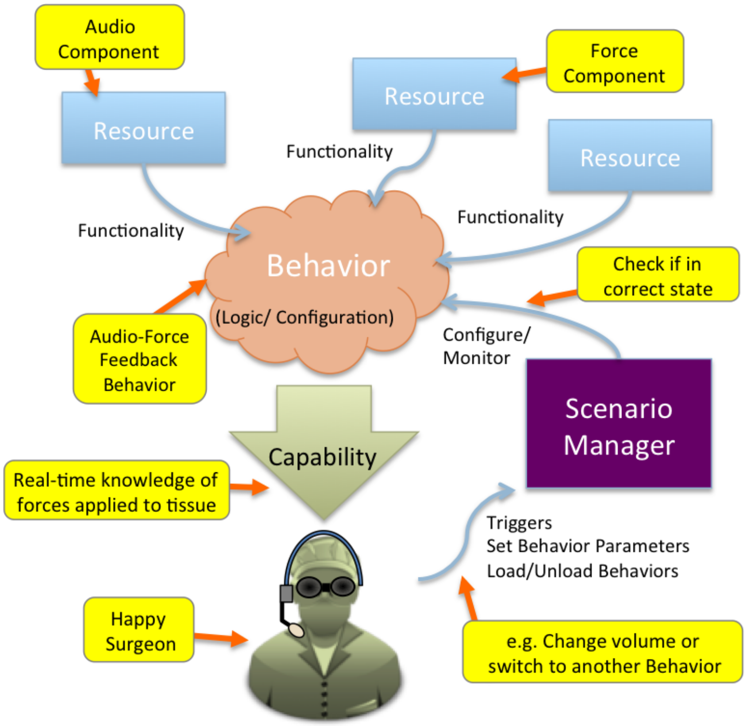


Figure 2.9: Concept schematic for a typical *Behavior*, in this case auditory sensory substitution for force sensor data.

An example of a Behavior is shown in Figure 2.9, where the functionalities of two resources are used by the Audio Force Feedback Behavior (AFFB) to provide audio-force sensory substitution capability to the surgeon (see Section 5.3). The AFFB is itself a Component that connects to a force sensing Component via a functional interface to access force sensor values and configure its operating parameters. The AFFB is also connected to an Audio Component that has a functional interface to play/pause/adjust various audio clips. Once the connections are established and the Resource Components are configured, both performed by the AFFB, a desired AFFB Operating Mode is set (e.g., “Warning Mode”). The AFFB then continuously collects data from the force sensor and triggers audio clips on the Audio Component that correspond to particular force values.

The AFFB can operate on its own but it is much more useful when other Behaviors are enabled along with it, such as the Robot-assisted Force Scaling (see Section 5.6). Furthermore, AFFB can be dynamically configured to use another force sensor Component that possesses a compatible functional software interface, e.g., a 3 DOF force sensor instead of 2 DOF. Also, since surgical procedures involve many steps that may require different Behavior operating parameters and surgeon preferences, these AFFB parameters can be adjusted using a GUI. However, it should be noted that manually adjusting these settings for many Behaviors simultaneously throughout a surgical procedure is just not practical.

As described earlier, a surgical procedure is composed of a set of tasks required to complete the procedure. Based on technical and specific human limitations faced by the surgeon in a particular task, a set of system Behaviors can be incorporated to provide the desired capabilities to augment the surgeon's abilities in executing that task. To provide an organizing principle, a notion of a *Task* is introduced which is a configuration (e.g., contained in a markup language) of the system that will provide those capabilities. The Task configuration includes the set of required Behaviors (e.g., Microscope Display Behavior and AFFB), their Resource definitions (e.g., AFFB should use 2 DOF force sensor), and particular Behavior settings (e.g., AFFB volume) specific to the task at hand.

The specific Tasks are grouped into a *Scenario* that corresponds to a surgical procedure. The Scenario construct enables the system developer to preplan the required Tasks and their specific Behaviors and parameters, so these settings can be "switched on/off" together. The Task parameters are preplanned in bundles of *Behavior Configurations* based on the requirements of surgical step, surgeon's preference, or patient-specific needs. Although a single Task is active at any moment

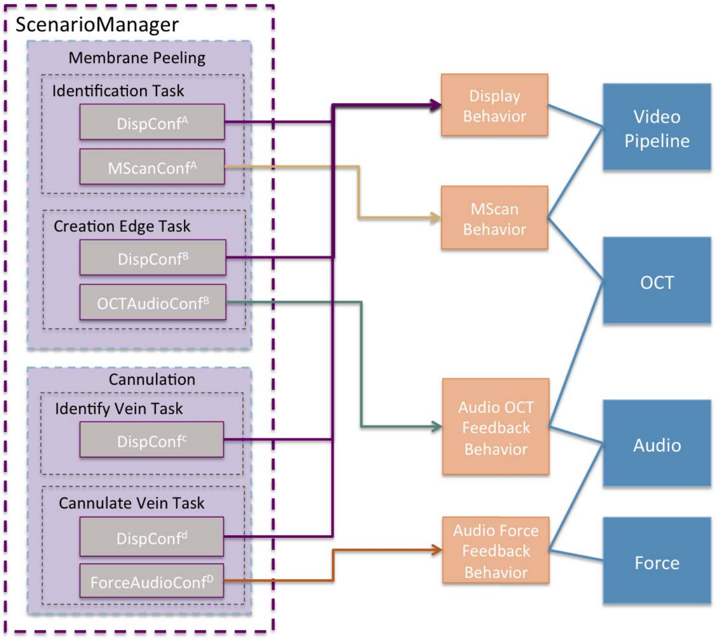


Figure 2.10: Schematic of an instance of the *Scenario Manager* showing two *Scenarios* with *Tasks* and corresponding *Behaviors* used in those *Tasks*.

during the procedure, the Behaviors can be reused for other Tasks in a procedure. This is done by applying a new set of Behavior Configurations. The overall management of Scenarios and Tasks is performed centrally by the *Scenario Manager* application. Figure 2.10 shows a simplified schematic of the Scenario Manager showing two available Scenarios, the Task with Behavior Configurations, and corresponding Behaviors used for a particular Task. Figure 2.10 also shows how an instance of a Behavior can be incorporated into another task (e.g., Display Behavior is used in all Tasks), and that Behaviors can reuse Resources (e.g., Audio Resource is used by two different Behaviors).

2.4.1 Behavior - Software Architecture Pattern

The overall system design architecture can be categorized into three hierarchical layers, as shown in Figure 2.11. The lowest level (Resource Layer) consisting of a collection of basic Resource Components (e.g., Audio component). The middle tier (Behavior Layer), which contains the Behaviors; and a higher tier (Scenario Layer) that is composed of a system-wide application called the Scenario Manager (SM) that includes system control logic, task configurations, and GUI objects that correspond to the Behaviors. This layout shows the top down design of the system, with the SM at the top level with direct access and control over Behaviors, and Behaviors directly interacting with relevant Resources.

In general, a Behavior involves composition of functionality and/or information from different Resources to provide a specific derived capability to the end user, in this case the surgeon. Software developers can create multiple self-contained applications that contain one or two Behaviors each, mostly based on the key technology involved (e.g., robot, visualization, OCT). Anecdotally, this was the approach for the initial versions of the *eyeSAW* system. However, during full system tests, each Behavior (initially called Module) had to be manned by a different operator,

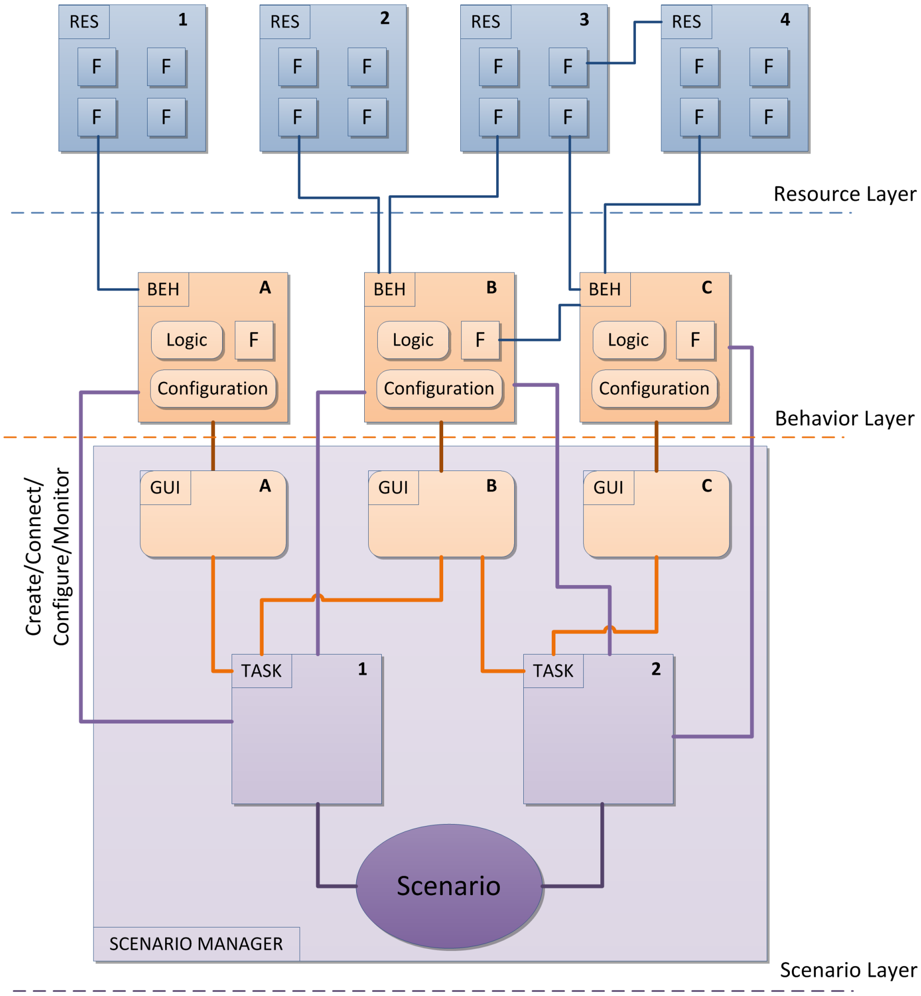


Figure 2.11: Schematic of the Behavior architecture showing three distinct layers and the interconnections between Resources, Behaviors, Scenario Manager and GUI components. Note: only a single Task is active at any one time.

sometimes requiring four individuals – one per user interface, to operate the whole system. To improve the logistics, a simple centralized control application (GUI with buttons) was developed that allowed the operator to trigger events and set parameters on the remote Behaviors. Each Behavior required additional interface "wrappers" (e.g., using the *adapter pattern* [61]) for integration into this centralized control application.

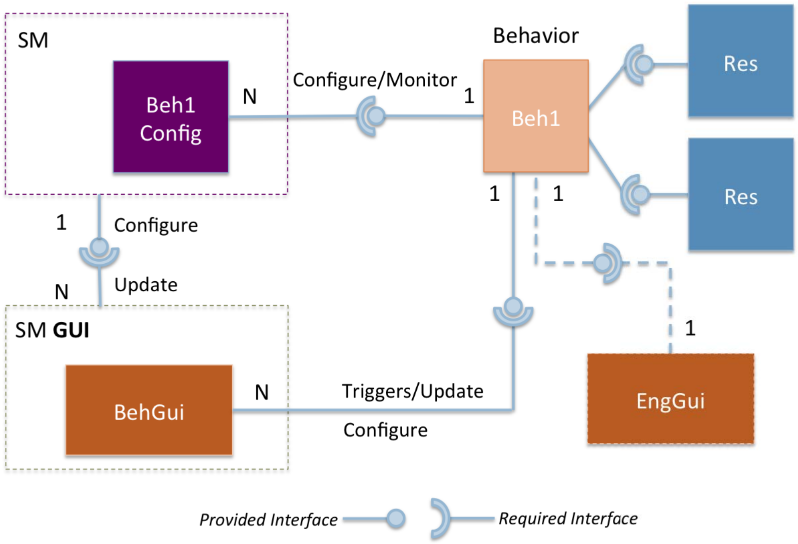


Figure 2.12: Abstraction schematic of the Behavior architectural software design pattern. Includes Required and Provided interfaces for each Component.

With the rapid growth of the system it became clear that a cohesive framework and software abstractions were necessary to facilitate system development, architectural organization, and run-time management of the Behaviors. Considering these issues the Behavior architectural software design pattern was developed, as shown in Figure 2.12.

The Behavior design pattern aims to establish a separation of concerns (SoC) so that a Behavior can encapsulate a complex service that involves details of configuration (itself and its Resource components) and Behavior-specific control logic from a higher level application (such as the SM). The Behavior Class object is a CBSE Component and also a specialization of the *mediator pattern* [61], which defines an object that encapsulates how a set of objects interact in order to reduce the direct coupling between lower and higher level objects.

Additionally, the Behavior pattern provides SoC in regards to the user interface (View) for state feedback and adjustment of specific run-time parameters of the Behavior (not to be confused with display Behaviors). This resembles the *Model–View–Controller* (MVC) architectural pattern [61] where the *View* is a Behavior-specific GUI (e.g., AFFB widget containing volume slider, warning zone range, etc.); and the *Model* is the state and parameters of the Behavior Object and also the data and parameters of the Behavior's Resources; while the *Controller* is internal logic located in the Behavior Object that interprets events from the *View* (e.g., change volume, forward to audio component) and updates its internal model based on the state of the Behavior and its Resources[4]. In a related work by Zheng et al., CBSE methods have been incorporated in MVC-based applications to "guarantee the usability and stability of the big amount of MVC-based softwares in the new environment" [62].

At the center of the Behavior pattern is the Behavior Class abstraction that is a specialized version of a CBSE component that has the following features:

- **Configuration of Self/Resources** - A Behavior is highly parametrized. Once it is instantiated, the Behavior is able to configure itself with the default parameters, and connect and configure any Resources that are needed for running the default configuration of the Behavior. The configuration also accepts external configurations (e.g., in a markup format) from the SM.

- **Control Logic** - The core application logic for the Behavior. (e.g., interprets data from two

[4] This type of Controller (in MVC) is sometimes called Mediator Controller, or a Controller that creates View Models that package the Model specifically for the View.

Resources and sends the processed data to another Resource to be rendered for the user.) It also is responsible for monitoring the status of relevant Resources and packages the internal state of the Behavior for use by the SM.

- **Operating Mode** - Specialized control logic for the Behavior (e.g., Audio Force Feedback: "Warning" Mode). Modes allow the Behavior to have a set of slightly different functionality without creating a completely new Behavior. This is useful when the topology of the Behavior does not change (same Resources), and only the control logic and some of the parameter settings are different. Idle Mode is the default operating mode where the Behavior is providing minimal functionality as deemed by the system designer to provide continuity of service (e.g., microscope visualization is enabled in the Display Behavior). It could also be used as a transition state (e.g., during configuration).

- **Component Interfaces** By following the CBSE design paradigm the major responsibilities of the Behavior are encapsulated and accessed through the following *Provided Interfaces.* Most Behaviors also have multiple *Required Interfaces* that connect to *Provided Interfaces* of the Resource Components used by the Behavior. See Section 2.5.2 for more information on inter-Component communication.

  - **Behavior Control** - The standard interface used by the SM Controller to configure, monitor, and control the state of the Behavior. E.g., the SM Controller continuously collects the configuration of the Behavior, so it can be restored in case of a crash.

  - **GUI Interface** - The standard interface used by the SM GUI to provide relevant feedback and control input for the Behavior. I.e., a way for the nurse technician to adjust Behavior parameters and trigger Behavior events "on the fly".

  - **Engineering Interface** - An optional, specialized *Provided* interface that is mainly used for development to allow a deeper access to the Behavior than the other interfaces. It can be connected to a corresponding engineering GUI or used via a scripting language interface.
  - **Functional Interface** - A *Provided* interface that exposes particular functionality (or data) making this Behavior a Resource that may be used by other Behaviors. (e.g., volumetric OCT C-Scans from the Scanner Behavior can be accessed by a virtual fixture Robot Behavior to provide collision avoidance in a targeting task.). The Behavior can have multiple interfaces for each significant functionality it provides. (Note: these are not shown in Figure 2.12.)

- **Behavior States** - A Behavior has three internal states that describe the status of the Behavior.
  - **Operating Mode** - The currently active control logic.
  - **Disabled/Enabled** - A way to disable and enable Operating Modes. Disabling can be used to configure Behaviors before one of the Operating Modes is active, especially when reconfiguring connections to Resources. When Enabled, one of the Operating Modes is active, and full configured functionality is available unless there is an Error.
  - **Error** - The general flag signaling the state of the Behavior for a given Mode. It is set if the Resources are not available, or other internal control logic error occurs, as defined by the developer. A human-readable message describes the error.

It should be noted that the system has a single instance of a particular Behavior, which is only connected to by a single instance of the SM. In some cases, multiple instances of the

Behavior can be created by instantiating the Behavior with a unique name in the system. However, such solution has to consider the way the Behavior uses Resources so that other instances of the same Behavior do not cause conflicts (see design guidelines, Section 2.4.3). A Behavior can be physically located anywhere in the *eyeSAW* system, but is likely in the same process as the Resource Component that it is using, so to minimize the use of overall computing and data transfer resources. However, multiple GUIs can connect to the Behavior simultaneously. In most cases a system will only have a single instance of a particular Behavior type (e.g., Microscope Display Behavior) since many Behaviors are tied to particular hardware Resource and can be reused for different tasks through the configuration process.

2.4.2 Scenario Manager

The Scenario Manager (SM) is a framework that guides the composition of a distributed dynamic application using Behaviors as the building blocks. An SM application is responsible for marshaling the Behaviors for a particular Task and providing standard methods for interacting with the system and its Behaviors. The SM layer is an extension of the MVC architecture for implementing user interfaces, as shown in Figure 2.13. The following are the definitions of the MVC components:

- **Model** - Procedures are described in terms of Scenarios, Tasks, Behaviors, and operating parameters. The Model is built from a system configuration file. See Figure 2.14 for excerpts from a sample configuration file. The model also contains the current state of the system (e.g., active Task and states of Behaviors).

- The **Controller** is the logic that interprets the Model, processes Scenario and Tasks transitions, and monitors the active Behaviors and updates the system state. The SM Controller is

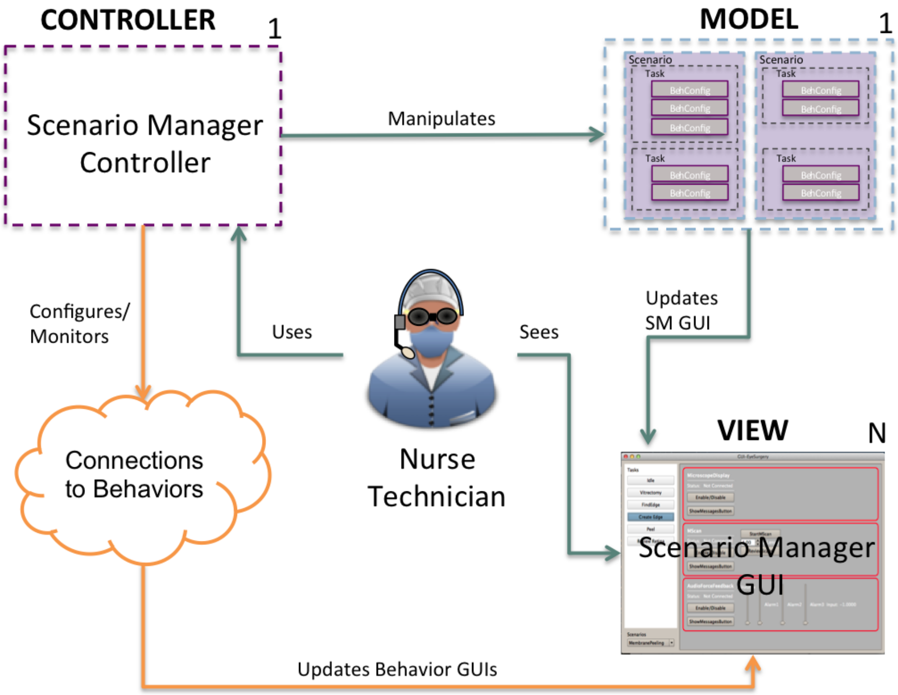
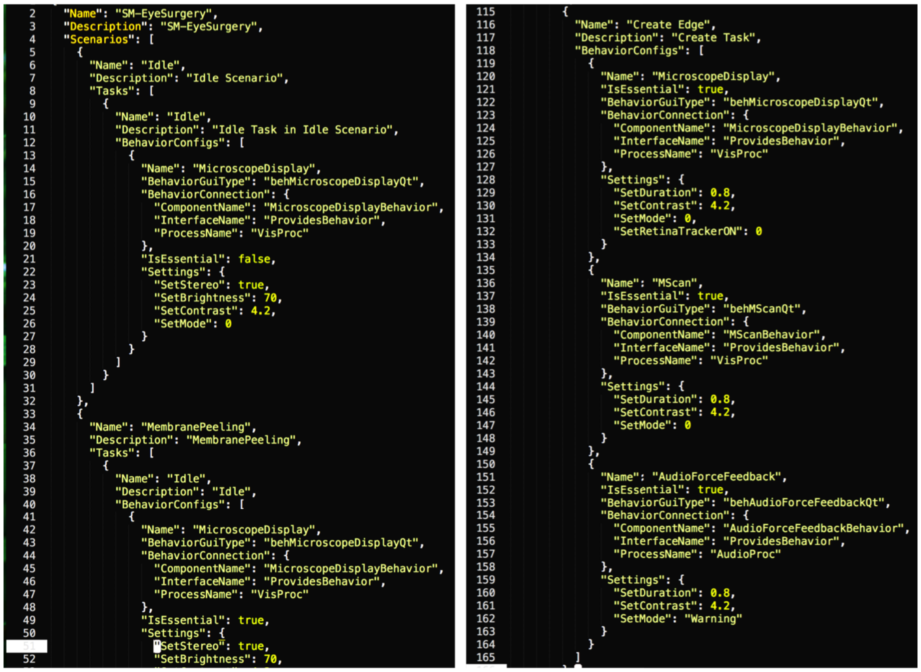


Figure 2.13: Scenario Manager Architecture is an extension of the Model-View-Controller design pattern.



```
  "Name": "SM-EyeSurgery",
  "Description": "SM-EyeSurgery",
  "Scenarios": [
    {
      "Name": "Idle",
      "Description": "Idle Scenario",
      "Tasks": [
        {
          "Name": "Idle",
          "Description": "Idle Task in Idle Scenario",
          "BehaviorConfigs": [
            {
              "Name": "MicroscopeDisplay",
              "BehaviorGuiType": "behMicroscopeDisplayQt",
              "BehaviorConnection": {
                "ComponentName": "MicroscopeDisplayBehavior",
                "InterfaceName": "ProvidesBehavior",
                "ProcessName": "VisProc"
              },
              "IsEssential": false,
              "Settings": {
                "SetStereo": true,
                "SetBrightness": 70,
                "SetContrast": 4.2,
                "SetMode": 0
              }
            }
          ]
        }
      ]
    },
    {
      "Name": "MembranePeeling",
      "Description": "MembranePeeling",
      "Tasks": [
        {
          "Name": "Idle",
          "Description": "Idle",
          "BehaviorConfigs": [
            {
              "Name": "MicroscopeDisplay",
              "BehaviorGuiType": "behMicroscopeDisplayQt",
              "BehaviorConnection": {
                "ComponentName": "MicroscopeDisplayBehavior",
                "InterfaceName": "ProvidesBehavior",
                "ProcessName": "VisProc"
              },
              "IsEssential": true,
              "Settings": {
                "SetStereo": true,
                "SetBrightness": 70,
```

```
        {
          "Name": "Create Edge",
          "Description": "Create Task",
          "BehaviorConfigs": [
            {
              "Name": "MicroscopeDisplay",
              "IsEssential": true,
              "BehaviorGuiType": "behMicroscopeDisplayQt",
              "BehaviorConnection": {
                "ComponentName": "MicroscopeDisplayBehavior",
                "InterfaceName": "ProvidesBehavior",
                "ProcessName": "VisProc"
              },
              "Settings": {
                "SetDuration": 0.8,
                "SetContrast": 4.2,
                "SetMode": 0,
                "SetRetinaTrackerON": 0
              }
            },
            {
              "Name": "MScan",
              "IsEssential": true,
              "BehaviorGuiType": "behMScanQt",
              "BehaviorConnection": {
                "ComponentName": "MScanBehavior",
                "InterfaceName": "ProvidesBehavior",
                "ProcessName": "VisProc"
              },
              "Settings": {
                "SetDuration": 0.8,
                "SetContrast": 4.2,
                "SetMode": 0
              }
            },
            {
              "Name": "AudioForceFeedback",
              "IsEssential": true,
              "BehaviorGuiType": "behAudioForceFeedbackQt",
              "BehaviorConnection": {
                "ComponentName": "AudioForceFeedbackBehavior",
                "InterfaceName": "ProvidesBehavior",
                "ProcessName": "AudioProc"
              },
              "Settings": {
                "SetDuration": 0.8,
                "SetContrast": 4.2,
                "SetMode": "Warning"
              }
            }
          ]
```

Figure 2.14: Example excerpts from a Scenario configuration file in JSON markup language.

not directly involved with specifics of Resource management but rather delegates that responsibility to the Behaviors. To ensure system consistency, the SM Controller is a system-wide singleton with a single active Model.

The general SM application workflow begins with creating the SM Controller, which involves the following steps:

1. Start the SM Controller application.
2. Load system configuration description.
3. Analyze configuration for parameter conflicts, hierarchy clashes, etc.
4. Set *Idle Scenario* as active.
5. Load *Idle Task*.
   (a) Create and connect to Behaviors for *Idle Task*.
   (b) Set Task-specific configuration to Behaviors.
   (c) Set as active Task.
6. Continuously monitor the Behaviors.

- The **View** provides a nurse technician the ability to view and trigger Scenario/Task state changes (See Figure 2.15), as well as high level control of the active Behaviors. The SM

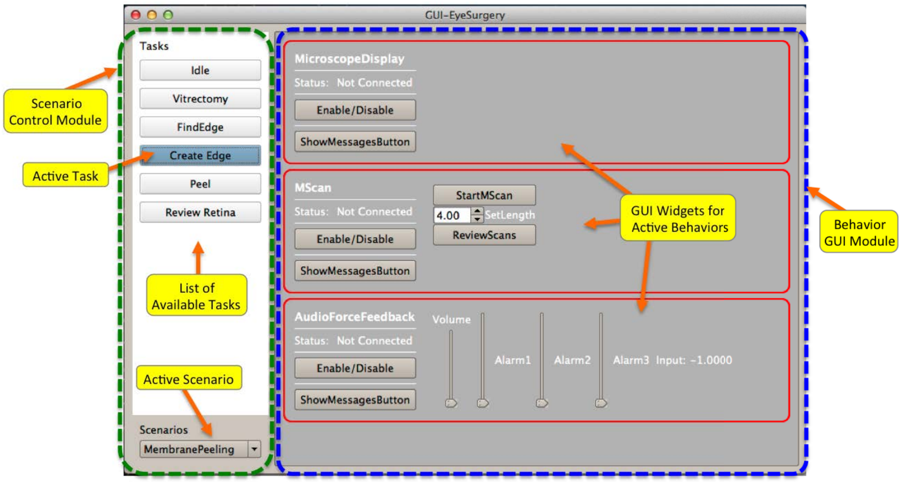

Status: Not Connected
Status: Not Connected
Status: Not Connected
4.00 SetLength
Input: -1.0000
Scenarios
MembranePeeling

Figure 2.15: Screenshot of the Scenario Manager GUI showing two GUI modules: Scenario Control and Behavior GUIs. The Behavior GUIs are dynamically loaded based on the active task.

GUI has two modules, the Scenario Control area that presents standard GUI widgets (e.g., buttons) for switching between the preloaded Scenario/Task model definitions of the system; i.e., the canonical *View* in MVC. The other is the Behavior GUI module that contains the Task-specific Behavior GUIs, which are very loosely related to the SM Controller in that their visibility is affected by the state of the system provided by the SM Controller. In contrast, the Behavior GUI is the *View* component from the Behavior Pattern and is associated with a single Behavior ("Controller"); i.e., for a given active Task, each Behavior GUI instance connects directly to the corresponding Behavior in the system. During the use the nurse technician can change the parameters of a particular Behavior by directly manipulating its corresponding GUI, which is located in the SM GUI. The interaction events are then forwarded to the Behavior (e.g., using the command pattern [61]). The Behavior GUI also fetches the state of the Behavior and updates its widgets periodically. In some cases these widgets are more than just a slider, or a button, but include images, real-time plots, videos, volumetric scans, etc. The Behavior GUI's are dynamically loaded (e.g., using a plug-in architecture) for each Task definition included in the Model that is provided by the SM Controller. The Behavior GUI parameters include the GUI object type, physical Behavior and interface information, and task-specific GUI visualization settings. There can be multiple SM GUI instances which allows for the system to be monitored and controlled from multiple points, e.g., tablet, desktop, smart phone, etc. It is important to reiterate the separation of concerns in the SM GUI: the Scenario/Task state changes are forwarded to the SM Controller while the Behavior specific triggers and parameters are directly sent to the Behavior. Likewise, the Scenario/Task data is updated from the SM Controller, while SM GUIs are directly updated from data fetched from the corresponding Behaviors.

Once the SM Controller is running, the SM GUI is instantiated and configured as follows:

1. Load SM GUI application.
2. SM GUI connects to SM Controller.
3. SM GUI fetches the Scenario Model definition and overall state of the system from the SM Controller.
4. SM GUI loads Behavior GUIs for a given Task based on the active Task and Model definition.
5. Behavior GUIs connect to their respective Behaviors.
6. SM GUI continuously updates the system state from the SM Controller.
7. Behavior GUIs continuously update their corresponding Behavior state from the Behavior directly.

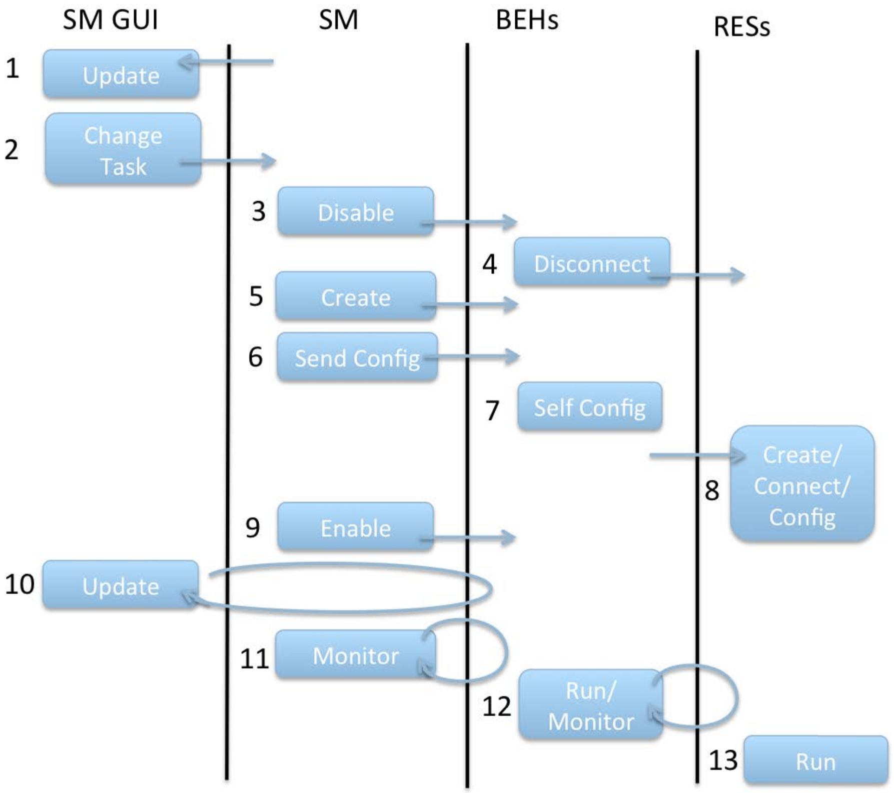


Figure 2.16: Sequence diagram illustrating the transition process between Tasks.

A typical Scenario change is similar to the startup of the SM Controller, however, in the transition process all the Behaviors are first disabled, and then the desired Scenario and its *Idle Task* are loaded. The surgeon can decide to follow a predefined consecutive sequence of the Tasks to complete the procedure or “jump” around as needed. The system sequence diagram in Figure 2.16 illustrates the transition from one Task to another, which includes these steps:

1. SM GUI is updated with the latest state of the SM Controller (e.g., Active Scenario, Active Task, Behavior descriptions).
2. User clicks on a non-active Task button, the SM GUI sends ChangeTask event to SM Controller.
3. SM Controller iterates through previous Behaviors and sends disable messages to the ones that are not required for this Task.
4. Each disabled Behavior disconnects their interfaces to the Resources.
5. SM Controller iterates through the new Task’s Behaviors and remotely invokes creation of a Behavior if it does not exist.
6. SM Controller sends desired configuration to each Behavior to set the Resource definitions (e.g., Component Name, Interface Name), Operating Mode, etc.
7. The Behavior configures itself (e.g., set default parameters if needed).
8. The Behavior creates required Resources, connects to their interfaces, and sets the required parameters for a given Behavior Operating Mode.
9. Once configured each Behavior is Enabled, if not enabled already.
10. SM GUI is updated to display the new task as Active, and corresponding Behavior GUIs are loaded. Both GUI modules (SM GUI, Behavior GUIs) are updated continuously.
11. The SM Controller is also continuously monitoring the Behaviors for errors, or disconnects.
12. Internally each Behavior is executing its control logic while monitoring the state of its Resources and connections. Its state is also updated for the Behavior GUI and Engineering Interfaces.
13. The Resources are also running their own control logic.

It should be noted that, depending on the implementation, some of the procedure calls above may be asynchronous, and in many cases each SM GUI, Behavior, Resources, have independent threads of execution. This allows each of the components to function independently and minimize system wide deadlocks. The Behaviors themselves are not aware of other Behaviors that are running in the system, with the exception of a Behavior that is using another Behavior as a Resource. In case of a severe problem with a Behavior such as no connection, the SM Controller sets its internal status of the Behavior to Disabled, and attempts to reconnect. Since the SM Controller continuously collects the Behavior configuration parameters from the Behavior itself, it is able to resend the most recent known configuration upon reconnection[5]. If the Behavior is not able to function as configured (e.g., Resource is missing), it is in an error state which is reported to the SM Controller. Because the Behavior GUI continuously requests updates from the Behavior, it is also aware of the error and responds accordingly by disabling widgets that are linked to a non-functional part of the Behavior (e.g., "grayout" volume slider linked to the missing Audio Resource), along with an error message.

One of the goals of the Behavior Pattern is to encapsulate functionality to provide robustness. This allows the SM Controller to ensure proper operation of the available Behaviors that are specified in the Task definition, i.e., it does not disable all Behaviors just because a single one went offline. Of course, if the SM Controller crashes the centralized control is no longer available, such as switching between Tasks and Scenarios. In that case, the SM GUI only provides the Behavior control to the Behaviors that were included in the Task definition at the time of the SM Controller crash. This allows the surgeon to finish the current surgical step and safely recover from the crash by restarting the SM Controller.

2.4.3 Design Guidelines

By following the general surgical system design methodology in Section 2.2, a set of requirements for a desired Capability is produced which are translated into functionality specifications.

[5] The ability to reconnect to, and configure, the Behavior to the last known state is very useful during development, e.g., when crashes of various subsystems are common, or when Behavior recompilation and restarts are required.

These are then implemented with the proposed Behavior framework and integrated into the *eyeSAW* system. The typical design workflow is as follows:

1. Select Resources that will provide the required functionality. In some cases new Resources may need to be created or altered.
2. Choose probable physical location of the Behavior (subsystem) considering available resources and likely system configurations.
3. Create a new Behavior (e.g., C++ class).
4. Create Required Interfaces for Resources.
5. Create Control Logic, including Behavior Operating Modes and Resource configuration logic.
6. Create Engineering Control Interface.
7. Create Engineering GUI.
8. If the Behavior generates information or controls Resources that do, it needs to provide a logging control through the Behavior Control interface.
9. Create Behavior Control Interface.
10. Select Parameters that will be used by the GUI.
11. Create Behavior GUI.
12. Incorporate into the Scenario Manager.
13. Test and iterate.

Many of the steps above require the system developer to consider multiple factors such as surgeon's needs, resource allocation, complying with Behavior design abstraction, etc. The following set of guidelines should be considered when implementing a surgical system based on the Behavior Software Architecture Pattern.

Behavior Design - A good Behavior has an intuitive and efficient human-machine interface and presents sensory information visually, aurally, and haptically, or provides active physical assistance (e.g., tissue manipulation assistance). A Behavior should provide the surgeon minimal information and/or manipulation assistance to complement the surgeon's manual skills and support the surgeon's expert knowledge in the most demanding phases of the procedure. Furthermore, an assistive technology has to work in completely predictable ways, whether it is a robotic manipulation assistance, or a visual feedback method. The Behaviors should strive to make the surgeon feel in control. Behavior design should consider ergonomics in all the aspects of the surgery, the patient, the surgeon, the OR equipment, etc. It should be safe, robust and minimize the possibility for unintentional misuse. Additionally, it should consider other Behaviors in the system to provide consistent user interfaces to avoid misinterpretation of function, e.g., pedal #1 is always for the Robot gain[6]. The Behaviors should strive to minimize the surgeon's cognitive load. For example, a Behavior that incorporates a visual overlay should present information minimally and clearly, and provide ways to tailor it for a given Task. A specific example is the Visual Force Feedback Behavior (see Section 7.8) which has two Operating Modes: (A) where the scalar data from the force sensor is overlaid as an intensity bar near the instrument; or (B) where a scrolling x-y plot is positioned near the corner of the Display and presents the history of the scalar force. For some Tasks, Mode (A) would compete with the surgeon's visual focus during a fine manipulation task under high magnification. However, it may be acceptable for a lower-risk task like palpation of tissue. Similar considerations exist for any other sensory substitution methods, as well as for direct manipulation assistance.

Behavior Design Granularity - It is possible to design an extremely configurable audio substitution method that can handle many different source inputs (e.g., scalar data, events), and render the data in a variety of ways. But, in practice, the source data is not always the same quality (noisy); it has varied data types; it may have different frequency of updates; and it may require special processing. In those cases multiple specialized Behaviors that have similar functionality are preferred, e.g., Audio OCT Range Feedback (AOFB in Section 6.4) vs.

[6] Some parameters can be adjusted through the SM configuration to improve interface consistency across Behaviors.

Audio Force Feedback (AFFB in Section 5.3). Furthermore, Behaviors should use different widgets or methods of feedback for two different purposes. For example, AOFB should use distinctly different audio clips than AFFB to avoid confusion.

Behavior Implementation - One of the key features of the Behavior software architecture pattern is the ability of the system to configure, monitor, and control all of the components making up a surgical system in a unified way. To achieve this, each Behavior is required to implement the Behavior abstraction, as outlined in Section 2.4.1, which includes control and GUI interfaces, Behavior configuration and control logic, etc.

Furthermore, the implementation of each Behavior needs to consider the available computing and communication resources, as well as the physical location of the Behavior and its Resources. For example, if a Behavior requires real-time access to the microscope video to perform some image processing or a direct access to a specialized interface (e.g., Visual M-Scan uses overlays; see Section 2.5), it should be located in the visualization process to minimize bandwidth and to reduce latency. Otherwise, the Behavior should reside on a different machine, to allocate maximum CPU and GPU resources to the visualization process, especially if the Behavior logic is computationally intensive. Another example is the Visual H-Scan Behavior (see Section 7.9) that displays a Picture-in-Picture video of rolling OCT A-scans. The captured OCT Data (hundreds of MB/s) is reduced and rendered as an image by a H-Scan Component on the OCT machine. When the image is ready, the H-Scan Component emits an event with the image as the payload, which is received asynchronously by the Visual H-Scan Behavior. This reduces the network load and improves latency since the amount of transmitted data is much smaller for the image than the raw OCT data. Additionally, the Visual H-Scan Behavior receives the image without polling and display it with very little actual processing of the already reduced data.

If a Resource Component that is used by a Behavior is self sufficient, i.e., it does not use special hardware or rely on other Resources, the Behavior may minimize the use of communication resources by instantiating its own version of a Resource in the same process (using a different name) to avoid costly data transfers from another process or machine where the original Resource Component resided.

Likewise, an instance of a particular Behavior may be duplicated with a unique Behavior Name and Configuration. This reduces code duplication and enables quick prototyping and testing of Behavior Configurations. For example, two Audio Force Feedback Behaviors can be simultaneously instantiated, each corresponding to different force sensing instruments and feedback types which are specified through the configuration mechanism. In such case there exists a higher chance of conflicts due to possible sharing of Resources, or of simply confusing the surgeon. Therefore, extra care is required to ensure that the configurations are valid.

A Behavior is required to manage all errors pertaining to its function as well as the function of the Resource Components it is managing. I.e., it is imperative that Behavior's control produces non-blocking response when its Resource malfunctions, is missing, or produces inconsistent data. This minimizes cascading effects of subsystem crashes that could bring down the whole system. Any significant errors should be reflected in the the Behavior State and produce human-readable output. This enables other Behaviors, as well as the SM Controller, to act appropriately. Furthermore, the Behavior GUI should report the Behavior error state, providing a clear visual and/or aural feedback to the user.

Inter-Component Communication - The *cisstMultiTask* (MTS) Component infrastructure (see 2.5.2) used to implement the Behavior framework uses *Required* and *Provided* interfaces in a

CBSE paradigm to connect a set of function calls (*Commands*) between two Components, see Figure 2.21, Section 2.5.2. Once connected the Components can either send *Events* or execute a *Read* or a *Write* Command on the remote component[7]. When establishing connections between the *Required* and *Provided* interfaces (i.e., connecting two components, Component A and Component B), the Commands in the *Required* interface need to exist in the *Provided* interface, otherwise the connection fails. This ensures that the Component (A) with the *Required* interface has access to expected functionality rendered by the Component (B) with the *Provided* interface. Additionally, the connection would succeed if Component (B) contained additional Commands in the *Provided* interface beyond the set that was required in Component A's *Required* interface. In practice, some of the commands in the *Required* interface definition were not always essential, and not always available in the *Provided* interfaces. Instead of building multiple *Required* interfaces with slightly different definitions, which is tedious and impractical, a per Command level of granularity for specifying required Command connections was added with an "IsOptional" parameter for non-essential Commands.

During run-time, a Component can check whether a Connection is active, and whether a particular Command has a binding, i.e., is available. A Behavior GUI Component with a *Required* interface may use this flexibility to expose many widgets corresponding to particular Commands when connecting to a *Provided* GUI interface of the corresponding Behavior. If a Command has a binding, then the GUI will display the corresponding widget. In this way, both GUIs and Behaviors can each provide functionality beyond a required core.

Even though two Components may provide interfaces that have identical definitions, their internal logic may differ and therefore simply interchanging them as a Resource of a Behavior

[7] Other Command execution models are available.

may not result in the expected function. The system developer needs to be aware of this and consider the context in which the Components exist.

The MTS inter-component communication mechanism provides Commands that can be used to "poll" to retrieve data (using a client/server messaging pattern) or "push" data using an *Event* (using publish/subscribe messaging pattern) that broadcasts data to all connected *Event Handlers*, and a *Write* Command that pushes data to a particular interface. It is recommended that Components that use a Behavior's Control and GUI interfaces use the "poll" approach to retrieve the data from the Behavior in a packaged form (e.g., poll all parameters in a single payload), rather than having the Behavior emit an Event for every internal state change, which could be very rapid.

The SM Controller and all the instances of the SM GUIs and the individual Behaviors GUIs are self-updating at a constant rate (e.g., 30 FPS) by retrieving status data from the Behaviors[8]. Such "polling" minimizes network congestion, and reduces the number of specialized Event handlers needed for each event type that would be required to represent the state of the Behavior (e.g., one for each control parameter), thus reducing the development time. User generated trigger events (change task, configure Behavior, set parameter), that typically originate on the GUIs, are infrequent and should use an MTS *Write* Command.

A typical Resources Component exposes its functionality through a *Provided* interface which often includes an *Event* Command. It is imperative that the Component that connects to this interface is capable of receiving and processing the *Events* at a faster rate than they are generated. Alternatively, a more sophisticated *Event* handler could be implemented that discards some of the incoming *Events* based on their significance. The same consideration

[8] SM GUI retrieves status from the SM Controller.

should be given to the *Write* command generated by a Component with a high-frequency control loop.

In general, *Events* should not include large payloads but rather generate a lightweight *Event* which signals that a new dataset is available[9]. The Component on the receiving end can then choose to fetch the large data with a *Read* command. Furthermore, *Events* should be used when low latency synchronization is important, or when the *Events* are infrequent.

Whenever possible, locate the Behaviors and Resources in the same process to significantly improve communication latency, bandwidth, and robustness. The distributed and dynamic nature of the *eyeSAW* System causes fluctuations in the performance of the network communication and is also susceptible to physical disconnects. In addition to hardware requirements and portability needs, communication requirements often influence how the components are grouped into physical subsystems (e.g., robot, OCT, visualization, etc.).

Scenario Manager - One of the most practical aspects of a centralized and "smart" configuration system is the management of the overwhelming numbers of parameters that are associated with each hardware and software component, especially when many are operating together. Some of the parameter control is internally managed by the configuration and control logic of the Behavior, but many are exposed through the Behavior's configuration interface and manipulated by the SM Controller for a given task. Defining these parameters in the System Configuration Model (i.e., Scenarios and Tasks) should be as explicit as possible (i.e., provide a value for all Behavior parameters) and consider possible conflicting topology and parameter assignment as described in **Resource Management**. A common conflict arises when two

[9] For example, small thumbnail images ($\sim$ 0.5 MB) that are generated at a relatively slow rate (30 Hz) and distributed on a local network can be sent as Event payloads, while 500 MB raw OCT scan payloads are probably not a good design choice.

Behaviors in a Task use the same Resource (e.g., Audio Component) and the last Behavior that is configured sets the particular Resource parameter (e.g., volume), overwriting the previously set value. As a result, the Behavior that was configured first will not end up with the expected value for that Resource's parameter. For non-essential parameters, a priority rating can be used for each parameter in the Task definition, and the highest rated parameter will be the one that is set. However, in some cases the parameters are more significant (e.g., set Force Instrument Type on the Force Component) and are essential for Behavior to function properly. Such parameter could be marked as *Essential* in the Behavior Configuration and a simple design-time graph analysis can be used for flagging conflicts. Similarly, a run-time analysis of the interaction graph could detect the conflict and inform the involved Behaviors, which then react accordingly. Alternatively, each Behavior can monitor the essential Resource parameters and react when they change, e.g., enter the Idle Mode and generate an Error event. In general, a thorough review of the control logic of the Behaviors that share Resources is advised since the Behaviors have direct access to the Resource Components and may change working parameters of Resources on-the-fly.

A related problem arises in the case of Task transitions, where some of the parameters set from the Behavior GUI should persist to the next task. This is addressed below in **Behavior Parameters** section.

SM Controller should perform system integrity checks at various stages of operation. First, the configuration file should be analyzed for any obvious conflicts, such as parameter redefinition, Behavior and Resource name clashes, etc. Second, a similar check should be performed during Task Transitions, where the required Behaviors provide a self-description (e.g., interfaces, parameters) which are then compared to the Task configuration to ensure that it is feasible.

The configuration check ensures that the Behavior has not changed since the definition of the Configuration Model, and vice versa. And finally, a similar check is performed continuously during run time (after initial configuration), which considers the self-descriptions from the Resources. This check is mostly responsible for ensuring that the topology of the system is consistent and omits the parameters set through the Behavior GUI interfaces.

The above run-time checks rely on explicit descriptions of the Components in the system[10] and can also be used to detect possible circular dependencies and race conditions in the system.

Resource Management - Occasionally, a Resource may be needed by two Behaviors at the same time. Some Resources (Audio Component) may be able to handle such a scenario, but in some cases a Resource (e.g., hardware) functionality may be only used by a single Behavior at a time. In such a case, the Resource should be duplicated; if it is a software Resource, it can be as simple as new instantiation with a different Component name, otherwise a physical copy of the hardware and the corresponding Component is created. Another approach is redesigning the Resource to provide multiple Interfaces for many Behaviors. A more complex situations could arise where a Resource (R1) is controlled by two entities and Behavior B1 is using a Resource R2, and R2 is using R1. In this case, R2 and B1 may attempt to manipulate (e.g., configure) R2 simultaneously which can cause unexpected Behavior. The system designer has to be aware of such possible conflicts. The system can assist the developer by providing topology checks of system configuration before it is applied to the system. A similar check can be executed during runtime to account for the dynamic topology changes. This requires a self-description mechanism of each Resource and Behavior. The SM Controller can generate a graph and determine how these resources are being utilized and whether there are any

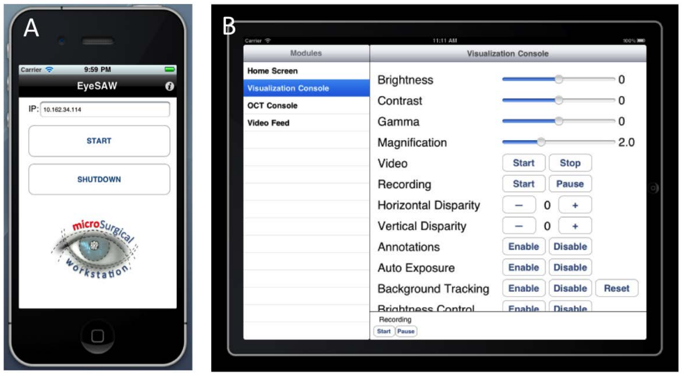
[10] Currently, only names of the Components, Interfaces, and Commands are available through MTS component definitions.

conflicts, circular dependencies, or missing Resources, etc.

Behavior Parameters - In the current SM implementation, any time the task changes, the predefined configuration parameters are loaded from a file and sent to the Behavior. However, there may be cases where the particular Behavior's parameters should not be updated during the Task transition. For example, Task T1 has an active Behavior B1 which has its parameters altered during runtime by the nurse technician. The next transition loads Task T2 which also uses Behavior B1. Should the SM controller use the Behavior B1 parameters that were set in the previous session or load the parameters from the initial Configuration Model? There are number of approaches to this problem. The simplest one is to use the predefined configuration. Another is to have persistent parameters that, once loaded, are not changed by the SM Controller, although a button can be added to each GUI to load predefined parameters from the predefined configuration. Alternatively, the configuration could include a descriptor for each parameter that specifies whether the parameter should always be loaded from the calibration, loaded only once per Task, or fetched from the Behavior directly.

Configuration Model - The initial implementation of the system configuration (see Figure 2.14 for an example) is in JSON markup language which is an open standard format that uses human-readable text to represent data objects consisting of attribute—value pairs [63]. These data objects are easily parsed and converted into corresponding C++ classes. Sections of the JSON file can be extracted and forwarded to a Component in a single entity to minimize interface duplication (e.g., a set of Resource definitions can be sent in a single *ConfigureBehavior* Command with the JSON object as the payload).

Scenario Manager GUI - One of the more important rules of interface design is that it should be intuitive and not surprise its users. I.e., the GUI should change in a consistent way, if at all necessary. The proposed SM GUI layout (see Figure 2.15) clearly separates the Scenarios and Tasks from the Behavior Interfaces. With a glance, the user should always be able to know the state of the system (e.g., active Task) and if there are any pressing errors. Furthermore, the SM interface widgets should also include descriptions of their functionality (e.g., cursor hover, a pop-up window exposes a description of the Task), while a widget that is associated with unavailable functionality should indicate its state (e.g., "grayed out"). Tablets and smart phones have been successfully used to control various aspects of the *eyeSAW* system (see Figure 2.17) and are excellent platforms for an SM GUI.



IP: 10.162.34.114

Brightness 0
Contrast 0
Gamma 0
Magnification 2.0
Horizontal Disparity – 0 +
Vertical Disparity – 0 +

Figure 2.17: A) Smart phone used as a simple SCM GUI. B) Tablet used to control the visualization subsystem.

Behavior GUIs - Similarly to the guidelines of the Scenario GUI, the individual Behavior GUI should be simple, clean, consistent, and user friendly. A consistent design should be applied to all Behavior GUIs. The Behavior GUI widget abstraction should include basic functionality common to all Behavior GUIs (Status, Error, Enable/Disable, etc). Since the SM GUI loads Behavior GUIs dynamically for each Task, their count and layout may change. It is imperative to layout Behaviors GUIs in a predicable pattern to help the users quickly locate a widget, e.g., alphabetical, most used, etc. Some Behaviors have specialized Operating Modes that may only use a subset of available Behavior control parameters. In that case, a context sensitive View logic should adapt the GUI so that only those parameters are active, e.g., non-active parameters are “grayed out”. Additionally, Task-specific GUI settings can be included in the preplanned system configuration that is sent to the SM GUI by the SM Controller during Task transitions. In general, real-time feedback from the Behaviors is very useful and should be included in the Behavior GUIs. For example, the Microscope Display Behavior GUI should include a thumbnail-sized real-time view of the microscope display. Although it is possible to generate all the widgets for a given Behavior GUI dynamically based on the Behavior GUI Interface description (e.g., parameter name, datatype, range of values, current value), in practice this yields visually inefficient layouts and also requires significant development to cover many special interpretations of the parameters and other data types.

The Behavior serves as a relay of the data from its Resources, which simplifies the system architecture at the cost of lower efficiency (data is transmitted to the Behavior and then to the Behavior GUI). In some cases where the data is large and network bandwidth is limited, the Behavior GUI could access the data directly from a component, bypassing the Behavior. This is generally not advised as it requires specialized logic and breaks the function and data encapsulation provided by the Behavior pattern.

Hardware Duplication - One of the major advantages of the Behavior pattern and CBSE is that it minimizes hardware and software duplication through dynamic system reconfiguration. However, there may be practical considerations where duplication of hardware is desirable. One example is a foot pedal (see Figure 4.9) that is used as variable input or a trigger for various Behaviors such as variable robot control gain or to trigger an OCT scan. Since these subsystems are often used independently and also in different locations, they each should have their own pedal device. In that case, it is imperative to use a Component to encapsulate the hardware functionality (input pedal Component) and assign it a unique name in the system. When the two subsystems are in the same location, it is still possible to configure one pedal as the input for multiple Behaviors even though they reside on different subsystems.

Scenario Manager Tools - To facilitate the design of Scenarios and Tasks, an application should be provided that allows the Surgeon and system developer to visually "script" the system configuration, i.e., design each Tasks' Behavior layout and its running parameters. Furthermore, a tool that saves snapshots of the system configuration during actual operations or mock operations could be used to grab real scenario parameters (e.g., robot gains, or volume settings). Sections of these configurations can then be imported into a new Scenario.

Data Collection - The data generated by the system is essential for experimentation, system development, and potentially for surgeon education. In many cases Resource Components already provide some inherent data archiving functionality, which can be controlled directly by the associated Behavior. The Behavior should specify which data streams are archived, their format, the location, and also provide feedback on the state of the archiving process. If the Behavior generates data it is responsible for providing methods to archive it. In some cases, a Logger Behavior that controls the archive parameters on various Behaviors and Components in the system can be implemented.

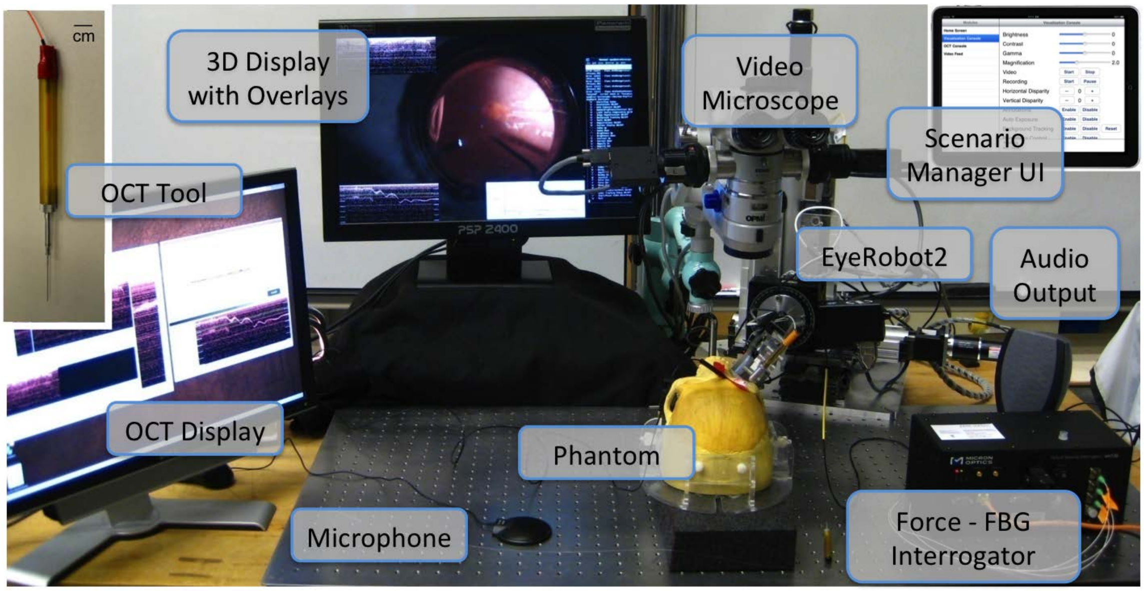


Figure 2.18: Hardware devices in the *eyeSAW* system.

2.5 System Implementation

2.5.1 Hardware

The *eyeSAW* system has evolved from a few independent applications running on dedicated hardware into a large, inter-connected, multi-module, distributed, and dynamic system running over many physical machines. Many custom hardware devices have been developed (e.g., micro-force sensor, robotic manipulator, OCT imaging system) and combined with off-the-shelf products (e.g., stereo display) to provide sensing, aural, visual, and haptic feedback functionality. A typical laboratory setup is shown in Figure 2.18 (see Section 3.1 for more details). These devices are often located on dedicated computers due to hardware and software dependencies, and computing requirements. Following the guidelines outlined in the Section 2.4.3, the system has been architected into distinct subsystems based on physical properties and collocated with relevant software functionality, as shown in Figure 2.19.

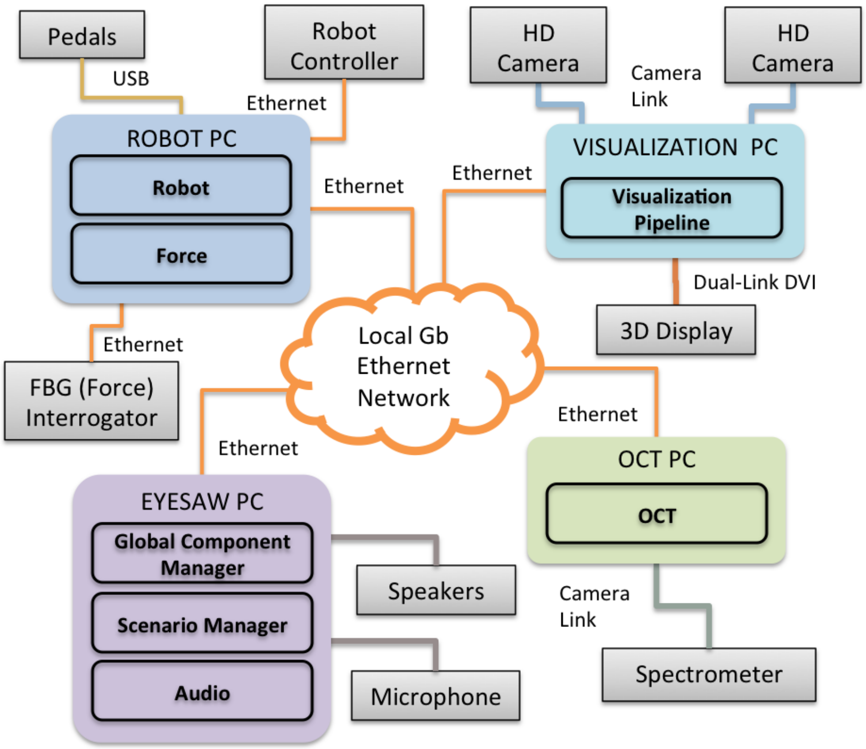


Figure 2.19: The physical layout of software and hardware subsystems comprising the *eyeSAW* system.

For example, the OCT subsystem requires specific data acquisition hardware (frame grabber running on Linux OS) [64] and performs computationally demanding signal processing at ~ 5 kHz to create OCT A-Scans and also to extract range information, while rendering 2D OCT images at ~ 30 Hz. Furthermore, the raw OCT data from a high resolution scan (e.g., M-Scan, or C-Scan) can be archived locally, in parallel, at 200 MB/sec. Simultaneously, the surgical visualization system, described in detail in Chapter 7, is running on a separate computer capturing ~ 350 MBytes of HD-quality stereo video, processing and displaying it at 20–30 FPS, and saving it to a local hard disk at ~ 17 FPS.

The communication between all the computers running *eyeSAW* software is over a Gigabit Ethernet network, which ensures dependable and fast communication. Wherever possible, peripheral devices, like the EyeRobot Controller (DMC-4080, Galil Motion Control, Inc.) described in Section 4.3, are "wrapped" by a single *eyeSAW* software Component and use dedicated communication hardware to minimize bandwidth usage on the main communication network. Data synchronization relies on system-wide timestamping and is realized through computer clock synchronization using Network Time Protocol (NTP) services [65]. The resulting millisecond accuracy is sufficient to synchronize data streams, most of which have 1–3 ms acquisition jitter. For example, this feature is used by the M-Scan Behavior, where a large set of OCT A-Scans are collected, bundled, and then transmitted over from the OCT computer to the visualization computer where they are synchronized by the M-Scan Behavior with a list of time stamped OCT tool positions from the visual Tool Tracker. Synchronized archive data also enables a "replay" of the procedure with *cisstDataPlayer*, a very useful tool developed for offline analysis and simulation, which is described in detail in Section 3.4.

2.5.2 Software

The entirety of the *eyeSAW* software is developed on top of the *cisst-SAW* [45–48, 66] C++ libraries (and also its Python bindings) that were introduced in Section 2.3.1. Figure 2.20 shows the overall software organization used for the development of the *eyeSAW* system with the following modules:

***cisst* Libraries** - includes OS abstraction, vector, numerical, vision, component abstraction, multi-threading and communication, robot control, etc.

***SAW* Components** - a collection of wrapped APIs of common hardware devices and encapsu-

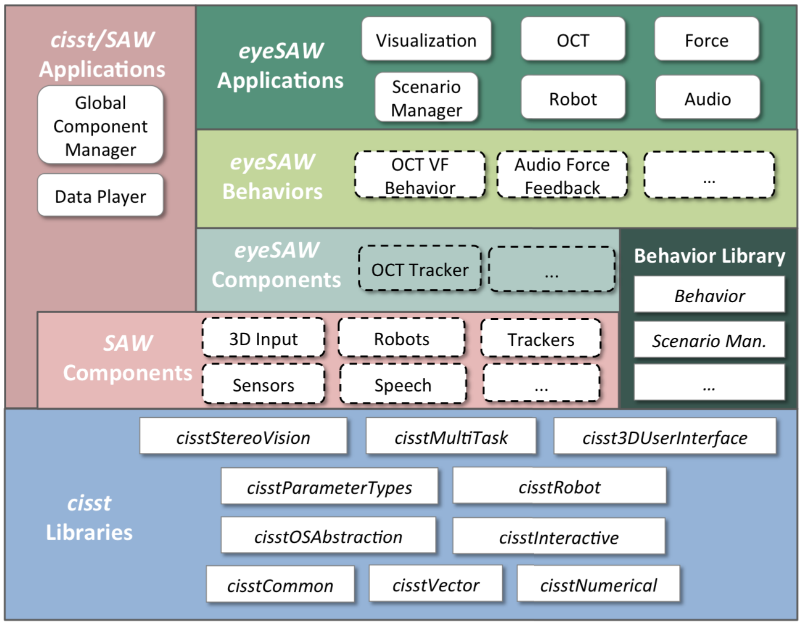


Figure 2.20: *eyeSAW* software organization.

lated functionalities relevant to computer-integrated surgery. Uses *cisstMultiTask* Library to implement Component Interfaces.

***SAW* Applications** - standalone executables of basic system utilities.

***Behavior* Library** - Behavior and SM abstractions, and utilities (C++) used to build Behaviors.

***eyeSAW* Components** - a collection of wrapped APIs of *eyeSAW* specific hardware devices and encapsulated functionalities.

***eyeSAW* Behaviors** - a collection of Behaviors that derive from Behavior Pattern Abstractions and are specific to the *eyeSAW* system. Behaviors use instances of *eyeSAW* and *SAW* Components.

***eyeSAW* Applications** - standalone executables that run in the same process and include *eye-*

SAW Components and *SAW* Components.

The *cisstMultiTask* (MTS) Library[11] [66] has been integral in realizing the Behavior Library. The MTS Library is used extensively for inter-thread and inter-process communication through its component-based abstraction – where a Component's*Required Interface* can be connected to a *Provided Interface* from another *Component*, as shown in Figure 2.21. Once connected to an interface, the *Component* can receive *Events* from and send *Commands* to the connected *Component*. These *Commands* resemble function calls (using the Command Pattern [61]) and, for intra-process *Command* execution, they "wrap" a standard function call. This avoids inefficient data copies and serialization/deserialization. For inter-process execution, the *Commands* are marshalled and transmitted via TCP/IP sockets. This mechanism is managed by the ICE middleware layer [67].

From the developer point of view, the calls to the *Commands* are identical whether used for inter-process or intra-process communication. In the case of multi-process applications, the connection of interfaces are managed by the *Global Component Manager* (GCM), which serves as a centralized dispatch for establishing connections between interfaces. Furthermore, this connection and disconnection mechanism can be executed dynamically at run-time from anywhere in the system and is an essential function required to reconfigure the topology of the system. It is used by many entities in the Behavior pattern: the Scenario Manager Controller requests connections to many Behaviors during a Task transition; during configuration, the Behaviors themselves request connections to Resource Components, and in some cases, disconnect and connect the interfaces between Resource Components directly using the GCM functionality. For example, the Audio Force Feedback Behavior (AFFB, see Section 5.3) may receive a request to use a new force sensor

[11] The latest MTS Library has been developed by Dr. Peter Kazanzides, Anton Deguet, Min Yang Jung, Balazs Vagvolgyi, and others.

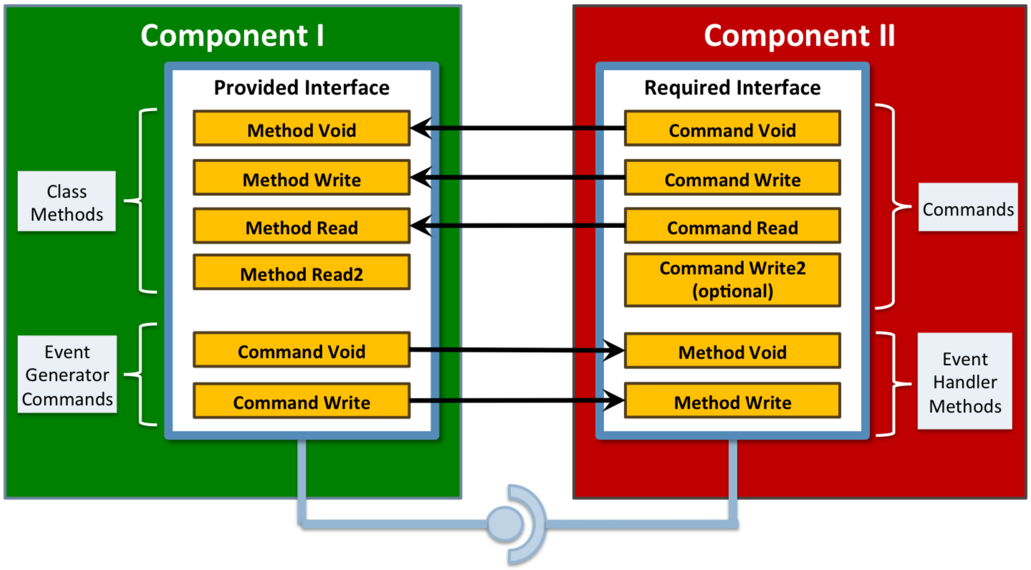


Figure 2.21: Schematic of a connection between two Components using *Required* and *Provided* interfaces, including how the *Commands* and *Events* are mapped to method calls. Note: Although not shown here, the *cisst* implementation of the Command functionality is done through the *mtsFunction* class which owns the Command object. The Command object's method binding, i.e., assignment of the method pointer, is done during the connection process.

component (3 DOF), in which case, AFFB disconnects from the 2 DOF sensor interface it has been using and requests a connection to the 3 DOF force sensor Component.

MTS also includes a utility for dynamic creation of Components. This functionality can be used by SM Controller, and by the Behaviors to remotely instantiate Component objects. Furthermore, the GCM, via the *ManagerComponentServer*, also provides the full system snapshot of all the Component Interface connections, as well as the Component status (configuring, enabled, etc.). This can be used by an application (e.g., SM Controller) to monitor the connection status of any Component or Interface in the system.

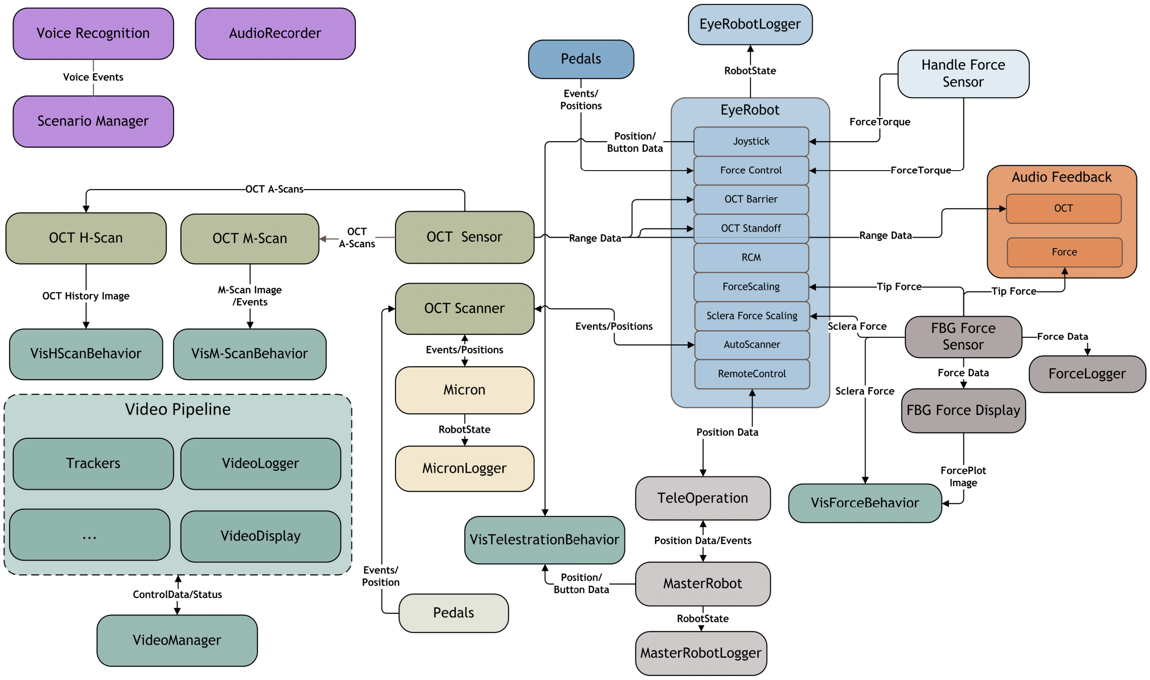


Figure 2.22: Schematic of the inter-connectivity of the Components and Behaviors in the *eyeSAW* system. Note: only a subset of Components and connections are shown.

The MTS library has enabled the *eyeSAW* system to grow quite rapidly, resulting in an extensive information exchange network as shown in Figure 2.22. The Figure shows a subset of the different Components (Resources) that are involved in providing functionality, mostly in the form of information, to the Behaviors in the system. The Components and Behaviors are color-coded based on their association with a particular subsystem, which in most cases implies that they are instantiated in the same process.

The *cisstStereoVision* (SVL) Library[12] has been used extensively to develop the 3D display subsystem which includes integrated visual tracking of the surgical tools and the retina, dynamic overlays, and most importantly the stereo microscope visualization. A typical SVL-based application uses stereo-specific processing *Filters* that are connected together to form a *Stream*. The SVL framework adopts a stream threading approach where each *Filter*, uses multiple threads operating on a section of the image(s), see Figure 2.23. Once all the threads in the currently active Filter are finished, the Stream processing continues to the next Filter.

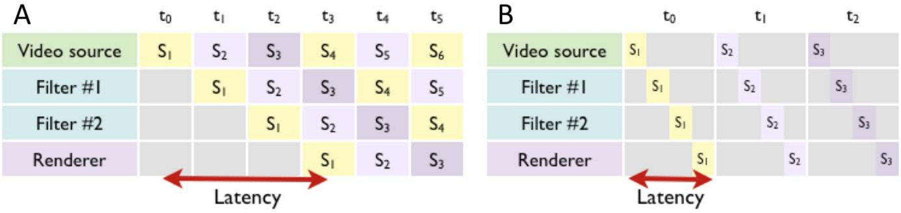


Figure 2.23: Execution models for processing pipelines: A) Image propagation in conventional pipeline. B) Image propagation inside an SVL Stream. Figure credit: Balazs Vagvolgyi.

Since the SVL filters operate sequentially, it is possible to use a single buffer for the output of one Filter and the input of the next Filter in the Stream. This minimizes large data copies and ensures economic use of the processing resources where threads are rarely idling. Furthermore, the Stream can be branched to allow for lower priority functionality to process information in a separate execution path using additional thread(s). Most importantly, such an architecture enables the main 3D display branch to run as fast as possible without waiting for Filters to finish processing the

[12] SVL has been developed by Balazs Vagvolgyi

latest video frame, as is the case with video archiving located on a separate branch. The branching architecture of the Visualization subsystem is shown in Figure 2.24, where the main display uses two threads (one per video channel), while branches that perform heavy processing use more threads (e.g., tracking uses 6) but are not in the critical path of the low latency display.

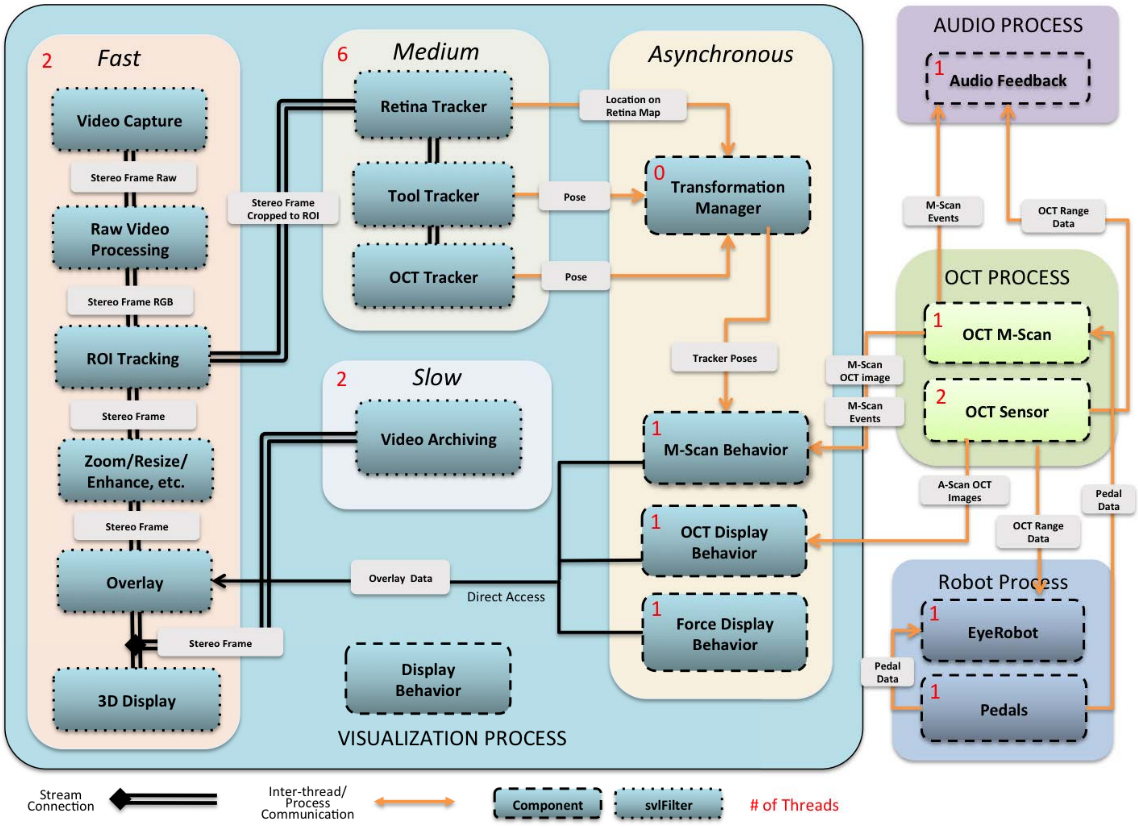


Figure 2.24: Detailed schematic of the video processing application and associated processes.

Additionally, the SVL Filters derive from an MTS Component and include standard Provided Interfaces to enable external configuration or to provide information (such as anatomical feature locations) to other Components, Behaviors, and Filters. The visual Behaviors run as independent (asynchronous) threads and use the MTS Library infrastructure to gather data from the system and control visual elements in the video processing Stream. One example is the Display Behavior, which dynamically creates the Stream for a particular Task configuration and provides a Behavior interface to control the Filter parameters, etc. Another example, is the Visual M-Scan Behavior (see Section 7.10) which receives an image from the OCT M-Scan Component[13], and then accesses a block of OCT tool positions from the Visual Transformation Manager (TM) (see Section 7.3) corresponding to the time span of the scan. It then creates a path overlay object representing the OCT tool trajectory, and registers it with the *svlOverlayFilter* along with the OCT Image overlay. During main video Stream execution, the svlOverlayFilter iterates through the overlay list and renders them directly from memory onto the desired image location to be displayed in stereo.

2.5.3 Inter-Process Communication Performance

The proper function of the *eyeSAW* system relies heavily on robust and low-latency communication between components. In the case where the components are on the same process, the command execution is a direct function call and considered to be instantaneous[14]. However, inter-process communication has a significantly extended execution time due to serialization, transmission over TCP/IP sockets and deserialization. To quantify this communication latency, an experiment has been devised to measure the time for a basic *Read Command* execution between two high frequency tasks running on separate modern multi-core LinuxOS machines located on the local *eyeSAW* Ethernet network (zero hops). The *Read Command* is a blocking call that accesses the latest available data sample from the remote *Component*. The following simple data payloads[15] were considered:

- 8 Bytes representing scalar range data

[13] OCT M-Scan Component creates OCT images that are used by the Visual M-Scan Behavior to construct an M-Scan visualization.

[14] Not considering the time to process the logic inside the function.

[15] The data payloads assume identical message metadata overhead.

- 1 KB representing robot's state
- 500 KB representing a 2D M-Scan thumbnail color PNG image (e.g., 640 x 240 px)
- 10 MB full-resolution M-Scan PNG image (e.g., 3500 x 1024 px)

All data are originally uncompressed[16]. For each data payload type a test trial included 1000 samples, where each sample is the execution time of a *Command* call. The wait between the *Command* calls was randomly chosen between 0–1000 ms and only one payload type was tested at a time during the trial session (1000 samples).

The same test was repeated while the *eyeSAW* system was in normal operation with the visualization computer at 90% processor utilization and ~ 4MB/70KB (incoming/outgoing) of traffic, mostly overlay images sent from the OCT and Force components. The test results are collected from the test running on the robot computer (30% processor utilization) requesting data from the visualization computer. Table 2.1 shows the results for the latency measurements.

Table 2.1: Command Latency for various payloads.

Payload	Basic System (max)	*eyeSAW* (max)
8 Bytes	0.42±0.07 ms (0.70 ms)	0.42±0.33 ms (4.63 ms)
1 KByte	0.57±0.06 ms (0.80 ms)	0.59±0.45 ms (10.10 ms)
0.5 MByte	26.28±0.29 ms (27.41 ms)	26.7±1.28 ms (41.82 ms)
10 MBytes	523.20±3.87 ms (536.22 ms)	534.28±10.83 ms (600.27 ms)

The inter-process communication latency was found to be acceptable for most processes that run at 200–500 Hz process loops and utilize small sized payloads < 1 KByte. It only becomes

[16] ICE middle layer may compress data to improve overall transfer speeds.

significant in the case of larger *Command* payloads (> 10 MBytes) that are sent rapidly, congesting the network and significantly loading the CPUs. Also the design of high frequency, real-time control loops requires consideration of an occasional transmission delay. The payload serialization/deserialization processing causes most of the delays, followed by network bandwidth usage. To optimize resource usage, especially in case of the video processing, many of the data streams, like OCT images, need to be pre-processed and packaged immediately on their originating machines to minimize data transmission overhead and fully utilize the available processing.

2.6 Conclusions and Future Work

This chapter presented the challenges of designing a computer-integrated surgery (CIS) system to work in cooperation with a surgeon in surgical scenarios that demand extreme skill and push the limits of human ability. A CIS system design methodology was proposed which decomposes the surgery into tasks that are individually analyzed to extract essential challenges faced by the surgeon. The capabilities required for each task were translated into functional units called Behaviors that combine functionality from various software and hardware components to address a particular surgical challenge. The Behaviors also form organizing units that are used to build a very dynamic surgical system.

The system and software engineering challenges associated with designing and implementing a CIS system for microsurgery were presented. Many of these challenges are associated with clinical, practical, software and hardware requirements, and the distributed architecture of the resulting system that requires its numerous components to share information and function. Such modular, yet extremely integrated approach, enables reuse of the functionality in the system by dynamic reconfiguration of resources to create new system capabilities.

Many of these concerns were addressed with a novel Behavior System Architecture Design Pattern based on Component-Based Software Engineering (CBSE) principles, and implemented with an MVC-like design pattern built on top of a CBSE Component framework included in the *cisst-SAW* libraries. A new Behavior should be implemented based on the provided Behavior Design Guidelines and an associated Behavior abstraction (control and GUI interfaces, logic, states, etc.) that allows it to be centrally configured through the Scenario Manager (SM). The SM comprises a Controller and one or more SM GUIs. The Controller is responsible for configuring and monitoring the Behaviors in the system, while the SM GUIs provide a control interface to the system. The SM GUI is used by the nurse technician to cycle through steps of the surgery corresponding to the tasks required to complete the procedure. A SM's Tasks configurations define Behaviors that are loaded and configured at run-time, while the corresponding Behavior GUIs are loaded into the SM GUI providing the nurse an interface to adjust the running parameters of the Behavior.

The presented architecture and the resulting *eyeSAW* system generalizes on multiple levels. The *eyeSAW* system can be easily applied to other microsurgical disciplines that have similar surgical requirements, such as otology or neurosurgery. This can be done by providing new Scenario and Task configurations supplemented by new Behaviors, e.g., Behaviors that are specific to the geometry of the procedure. In fact, parts of the *eyeSAW* system (EyeRobot, Visualization, OCT) have been used in research to improve cochlear implant surgery [68, 69].

Furthermore, the flexibility and modularity that enables the system to be applied in different procedures is increasingly important as hospitals are looking to cut costs while simultaneously striving to improve the quality of surgical services. The modularity could also allow a hospital and its surgeons to choose a baseline system configuration to cover a limited set of procedures that are essential but very challenging for the surgical staff, and later on, when the practice expands, incorporate additional Behaviors as “add-ons”.

The system architecture generalizes well to other domains that involve many devices and software functionality, and have high performance human-centric requirements. For example, the overall architecture can be used in a robot-assisted satellite servicing system such as the one presented by Xia and Kazanzides at al. in [70, 71]. In such a system, many components are marshaled to provide multiple capabilities and operate in unison with a remotely-located human operator to perform a highly skilled mechanical task. In a satellite servicing scenario the Behaviors may exhibit more autonomy to tolerate constraints of high latency, low-bandwidth communication with the remote operator.

The Behavior design pattern presented in this chapter is a conceptual breakthrough that provides a concrete framework for formally organizing a complex computer-integrated surgical system. The Behavior design and abstraction implementation are a work in progress and will require more iterations to refine the overall design for clinical trial level. The following points describe the lessons learned during the development of the *eyeSAW* system and the associated software framework, and also outline directions of future research:

- To refine the Behavior design pattern it is necessary to gather more use case data from a large pool of system developers who apply the pattern to build surgical systems. It is virtually impossible to design a high level architectural design pattern that will fit all design scenarios, so the design pattern and its implementation will need to evolve to cover the currently unforeseen requirements of future hardware and software technologies.

- Future research should investigate methods for a comprehensive self-description schema for Behaviors and Resources, including interfaces, parameters, dependencies, estimated CPU, memory and network bandwidth usage, etc. A rich self-description of Behaviors would enable a more in-depth control of each Behavior, especially in cases of resource conflict detection and resolution.

- As in any multiprocess distributed system, concurrency control and resource management are important issues that lead to deadlocks, inconsistent system topology, etc. This is a general problem that has been investigated for many decades (e.g., in operating systems) and should be considered in more depth in the next *eyeSAW* design iteration.

- Although system safety was not the main focus of the work presented in this chapter, it is a very important aspect of a surgical system design. Kazanzides and Jung et al., have been actively developing a scheme for fault detection and recovery in component-based systems [72]. Additionally, Kazanzides and Kouskoulas et al. investigated more formal methods to reason about the correctness of concurrency mechanisms for data exchange between components [73].

- Even though surgical procedures are well planned in advance, unforeseen complications may require the surgical staff to alter the order of the surgical steps or require new devices and information to address the emergency. For this practical reason a new functionality is required where a capability can be added on-the-fly by the nurse technician during the procedure. E.g., a GUI widget that lists Behaviors that could be added to the current Task during run-time. Note that this may add complexity in regards to Resource conflict management, parameter configuration clashes, and in some cases require costly and time-consuming regulatory approval.

- Another practical feature that would be very useful in experiments is an *Event Macro*. An Event Macro is a set of timed events that are generated across a number of Behaviors. For example, the macro could start audio, OCT, and robot loggers; wait 5 seconds; then trigger a robot-assisted OCT scan; stop loggers. A solution could use a scripting language (e.g., via Python bindings) that interacts with the Behaviors directly, or build a scripting module into the Scenario Manager that also has a corresponding SM GUI interface, and displays button widgets to trigger the Macros.

- Good ergonomics of human machine interfaces (HCI) are crucial in the acceptance of the system but were not formally addressed in this work. Pedals have been adopted as the preferred surgical interfaces into the system because they are common input devices used in the ophthalmology ORs. However, pedals may not be the most efficient multi-functional interface to trigger various Behavior events. The future HCI for the *eyeSAW* system requires proper user experience studies and optimization.

  The recent advancements in the accuracy of speech recognition could provide a non-contact alternative to control system parameters that is easily configurable and direct. Such capability should be encapsulated in a Behavior and used to trigger system events, e.g., “Start OCT Scan”.

- Future implementations of the system should include the recording of all SM-level events including Scenario and Task transitions, Behavior parameter changes, etc. Such history of user interaction with the system can be combined with contextual information derived from surgical data recordings (audio, video, notes, force data, etc., see data collection in Section 3.4, and analyzed to create statistical system usage models. These models can than be applied in many interesting ways. One is automatic or semi-automatic transitions between tasks based on detecting the intent of the surgeon. Kragic et al. [74] presented a method where traces of user activities are parsed into task state models and used in an on-line recognition to transition to subsequent states and initiate the appropriate robotic assistance. Alternatively, a similar analysis can be used to suggest a common set of Behaviors and parameters that have been used by an expert surgeon for a given surgical situation. This can also be extended to refine user interfaces by pruning unused widgets or making the commonly accessed widgets (or full Behavior GUIs) more prominent. The default settings could also be automatically tailored this way for a particular surgeon and a particular procedure.

Comparing the current surgical context to the current Task configuration could be used as an additional safety feature to warn if the active SM Task or its Behaviors are not matching the step in the procedure. For example, the Force Scaling Behavior should only be enabled if the force-sensing instrument is in the field of view.

The user interaction, along with all the data collected during the procedures, could be annotated and stored in a *Case Library* for off-line educational purposes, or used as a reference during cases, or for immediate review.

- One of the benefits of a fully integrated system is that it facilitates the discovery of multiple uses of a single purpose device or a software function. For example, the force/torque sensor on the EyeRobot was re-purposed as virtual joystick (see Section 4.8) as a natural input into the visualization system. The use of the force/torque sensor in this way was not obvious until the two subsystems (robot, visualization) were used simultaneously for another Behavior. Furthermore, implementing new capabilities is very fast due to standard interface definitions of the existing Components. For example, the Cooperative Teleoperation Behavior where the EyeRobot is remotely controlled using a DaVinci Master console (see Section 4.9) was developed and demonstrated in one day. The system has also been extensively used to assist in bench-top experiments. In one example, an existing force sensing Component was used to verify tool-with-retina collisions in visual proximity detection experiments (see Section 7.11).

The final success of the architecture will be the ability of other system designers to follow the design patterns presented here while carefully considering surgical and clinical engineering requirements to build a safe, efficient, and effective cooperative surgical system.

2.7 Recapitulation of Contributions

Surgical System Development Methodology (Section 2.2) A design methodology was presented for developing a complex distributed software and hardware system for surgery by dividing surgical procedures into discrete tasks, then extracting significant surgical challenges from each task, and proposing assistive capabilities that could address these challenges. Each capability is then realized in hardware and software, and encapsulated in a software construct called a Behavior, and is enabled whenever needed in the procedure. This approach can be used in developing other computer-integrated systems that assist the human in multiple ways in highly demanding tasks. The process was used to develop surgical Behaviors presented in the following chapters. **Credit**: Marcin Balicki.

Behavior - Software Architecture Pattern (Section 2.4.1) A novel (Behavior) software architecture design pattern was developed for realizing surgical system. It is based on the MVC design pattern and incorporates the principles of CBSE. The Behavior pattern is primarily concerned with encapsulating a surgical assistance capability using existing hardware and software CBSE Components and providing standardized interfaces for centralized control and configuration, and user interfaces for run-time parameter adjustment. The design includes a configuration framework that simplifies the dynamic organization of the system for multi-step surgeries by grouping Behaviors into Task and Scenario configurations. This enables run-time configuration and simultaneous use of multiple surgical Behaviors for a given surgical task.

The Scenario Manager (SM) simplifies administration of the system by providing run-time switching of preplanned configurations of the system and also performs high-level monitoring of the system components during operation. The software architecture can be implemented for any multi-component and distributed system that requires centralized configuration and control. **Credit**: Design of the architecture pattern and prototype Behavior library development by Marcin Balicki with assistance from Balazs Vagvolgyi.

Behavior Design Guidelines (Section 2.4.3) The lessons learned during the development of the *eyeSAW* system were analyzed and presented as challenges and the best practices that should be considered when implementing a system based on the Behavior pattern. **Credit**: Marcin Balicki.

***eyeSAW* System Implementation (Section 2.5)** The first implementation of a distributed, multi-functional "smart" retina surgery system that includes intraocular force and optical sensors; robotic assistants; video microscope visualization; and aural, visual, and haptic feedback methods. The hardware and software are formally organized into subsystems and Behaviors, and are developed based on the *cisst-SAW* infrastructure and architecture described in this chapter. The current system has been applied in otology research, e.g., cochlear implant surgery, and can be applied to other microsurgery applications. **Credit**: Marcin Balicki designed the overall architecture of the *eyeSAW* system and is responsible for putting all the pieces together into a functional system, including hardware, software framework, and software applications. Many of the individual components were developed by the members of *eyeBRP* team, often in collaboration with Marcin Balicki. The *cisst-SAW* infrastructure was developed by Dr. Peter Kazanzides, Anton Deguet, Balazs Vagvolgyi, Min Yang Jung, and others.

Chapter 3

Experimental Testbeds

A significant effort has been invested by the *eyeBRP* team to implemented several experimental testbeds to cover specific testing scenarios. Not only are these used for validation, but they are also indispensable in every step of system and individual technology development, from surgical ergonomics to specific device functionality validation. In some instances a single purpose phantom (e.g., BandAid membrane peeling phantom) has been developed to control for many environmental conditions, which can include lighting effects, vibrations, repeatability of material properties, geometric eye constraints, simulation of costly biological tissues, preparation time, ease of use, etc. Without these testbeds, it would be virtually impossible to accurately measure the performance of the system. This is especially true in the case of vitreoretinal surgery where micron-level scale is inherently difficult to work with, and access to consistent biological samples is limited and time consuming, often requiring expert skill.

3.1 Bench-top Setup

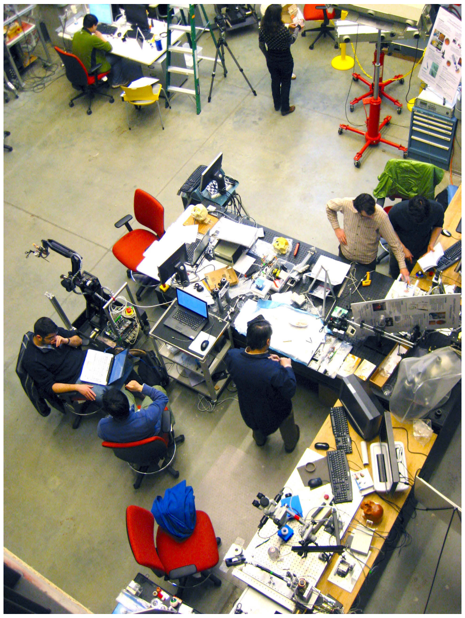


Figure 3.1: Eye Surgery development area in the Robotorium.

The experimental stereoscopic visualization system comprises a conventional ophthalmology stereomicroscope (Carl Zeiss OPMI MD) with a magnification of 40× and 12.5× for the eyepieces, and an effective focal length of $f = 200$ mm. A digital pedal interface is used to control the actuated zoom and focus. This microscope is also customized for video microscopy with two Grasshopper 20S4C Firewire cameras (Point Grey Research, Inc.), which have been coupled to the microscope using custom C-mount adapters from Microstereopsis, Inc. These cameras are capable of capturing 1600×1200 pixels images at 30 FPS and are aligned mechanically to have zero vertical disparity. Several different 3D display technologies are used in the system. The most recent is a progressive 24" LCD display with 1920×1080 pixels resolution (Alienware Optx AW2310T) which functions with active 3D shutter glasses worn by the viewer(s). An alternate LCD display (PSP2400, Panoram Technologies Inc.) used for some of the experiments employs a passive glasses technology. It also provides 24" LCD view, but with 1920×1200 pixels resolution (vertical 600 pixels per eye). At this resolution, if the monitor is placed at least 90 cm away from a person of nominal vision acuity, the individual pixel boundaries become invisible.

To light the experimental area fluorescent overhead lighting and a table mounted fluorescent ring lamp are used. Intraocular lighting is provided by a conventional halogen light source

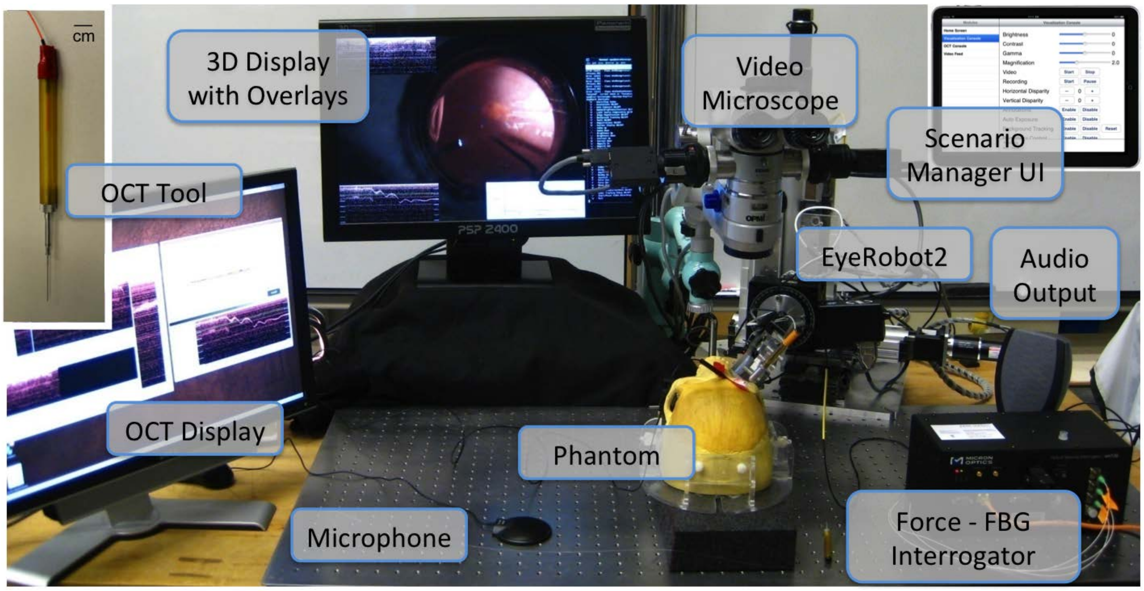


Figure 3.2: Bench-top experimental setup.

with a flexible shaft output, coupled into a disposable 23 Ga surgical light pipe (Alcon, Inc.). Off-the-shelf surgical instruments, like disposable 23–25 Ga forceps (Alcon, Inc.) and 25–27 Ga (BD, Inc.) hypodermic needles are used to closely simulate real surgical scenarios. To facilitate single operator development of the system, passive positioning arms (e.g., DG Holder DG61003, Noda, Inc.) are used to hold surgical instruments inside the eye, while various adjustable hight platforms ("Lab Jacks") are used as material and hand rests. An omni-directional conference microphone (CM-1000 USB, SoundTech, Inc.) is placed on the table near the operator or attached to the microscope. Custom software records time-stamped audio during experiments which is then synchronized with video or other collected information during post-experiment data analysis. All the experimental data is stored immediately after the experiment on a redundant storage system. The data includes room audio recordings, microscope video, sensor (force, OCT) data, robot state, application source code and settings, and any external photos and videos that are taken during the

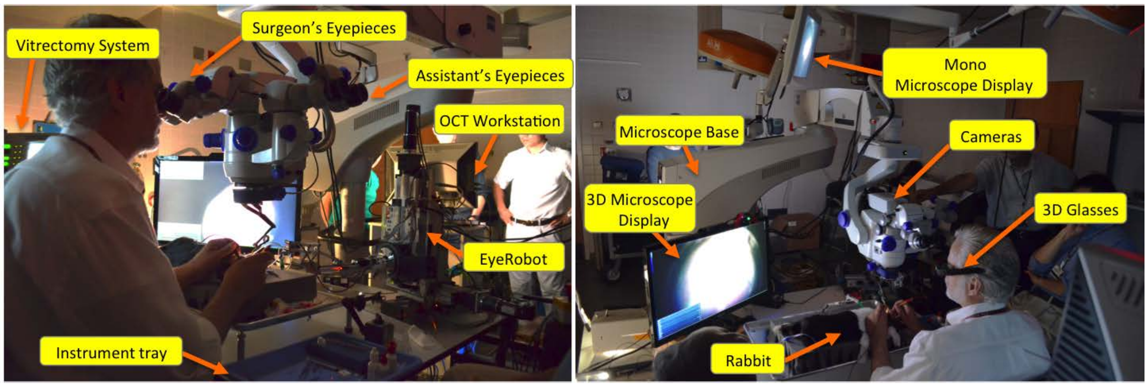


Figure 3.3: Operating room setup.

procedures. Furthermore, the storage system has an associated incremental off-site backup.

Figure 3.2 shows the typical system setup with a 3D display, EyeRobot, force sensor interface, OCT sensor display, audio microphone, etc. The custom software applications (e.g., visualization pipeline) run on off-the-shelf standard multi-core workstations.

3.2 Operating Room Setup

In order to closely simulate a typical surgical environment, the *eyeSAW* system has been replicated in the (decommissioned) Operating Room #3 (OR3) of the Wilmer Eye Institute at The Johns Hopkins Hospital[1]. The space is a former eye surgery operating room with standard fluorescent overhead lighting and no natural light. A sturdy workbench (36"×72") was placed in the center of the room to provide ample surface area for the 3D display, rabbit tray, two instrument trays, and the robot. Various workstations and custom equipment were placed around the table, with care taken not to interfere with the surgical workflow. The visualization system uses a free

[1] Much credit should be given to Kevin Olds and Balazs Vagvolgyi for assisting in the development of the experimental setup in the operating room. Many thanks to Carl Zeiss Meditec for the loan of the video microscope.

standing OPMI Lumera 700 operating stereo-microscope (Carl Zeiss Meditec AG) with two proprietary built-in, full-HD, progressive cameras (60 Hz at 1920×1080 pixels resolution). The cameras are aligned mechanically to have zero vertical disparity. The 3D progressive LCD display is 27" with 1920×1080 pixels resolution (Model VG278, Asus, Inc.) and is used with active 3D shutter glasses worn by the viewer. If necessary, the main operator and an assistant could also use the two standard optical stereo view ports. The microscope has a wireless foot control panel for zoom, focus, translation and integrated slit illumination control. The microscope has two integrated wide angle lenses which, when needed, are slid into position by the assistant. In some cases, the optics affect the video capture requiring real-time software-based image inversion, and adjustment of parameters in visual tracking algorithms.

For ex-vivo experiments, a standard bench-top light source is used while for in vivo experiments, an Accurus Vitrectomy system (Alcon, Inc.) provides a xenon light source, ensuring essential surgical functionality - infusion pressure control, vitrectomy, fragmatome, etc. A standard surgical hand support bar is attached to the table to mimic real vitreoretinal surgery ergonomics.

3.3 Experimental Phantoms

3.3.1 Human Skull

To provide an anatomically realistic testing environment, a customized solid foam human skull model (Sawbones; Pacific Research Laboratories, Inc.) is mounted firmly onto specially-designed acrylic platform to lay supine on the flat surface. The platform can also be fixed to the optical bench with screws for added stability. To provide traction, the forehead is covered with brush-on latex rubber (Mold Builder Liquid Latex, Castin' Craft). A lab jack type adjustable plat-

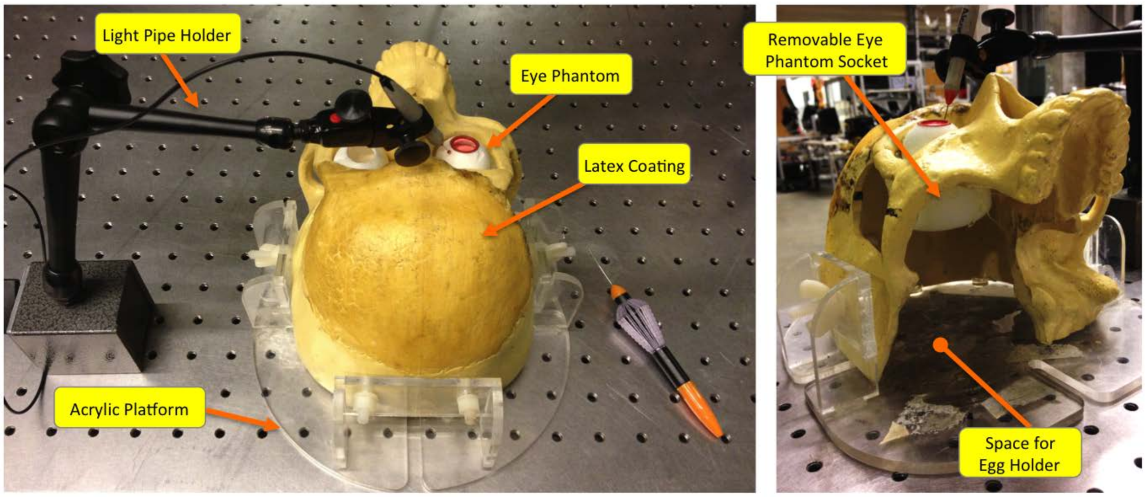


Figure 3.4: Human skull with JHU Eye Phantom.

form is outfitted with a custom acrylic plate to provide sufficient wrist/arm support when operating in the eye region. The skull is altered by drilling out the right orbital cavity and surrounding skull structures (e.g., sphenoid and temporal bones) to generate space for egg and eye ball phantom holders. For egg based experiments a sclerotomy port is simulated with a rapid prototyped adapter made of ABS plastic (Dimension 3D Printer; Stratasys, Inc.) that fits perfectly over the orbit. The adapter is designed by scanning the surface of the orbit using a MicroScribe 3D digitizer (Immersion, Inc.) and incorporates a circular window to accept a variety of inserts with rubberized sclerotomy ports (see Figure 3.11D–F). These ports are made of rubber to simulate the approximate elasticity and feel of the sclera.

3.3.2 JHU Eye Phantom with Interchangeable Retina

Realistic eye phantoms, especially ones that are filled with liquid are limited in availability [75]. Therefore a custom phantom mimicking the mechanical and appearance properties of the eye

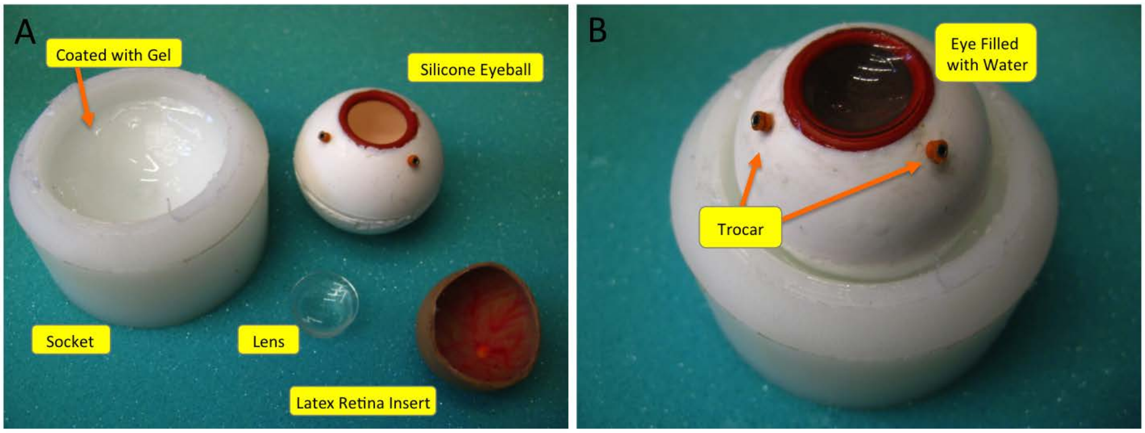


Figure 3.5: : A) Components of the JHU human eye phantom. B) Assembled phantom. Note: Iris is not shown.

has been developed[2]. The 25 mm ID ball (sclera) is cast out of soft silicone mold rubber (SORTA-Clear 18; Smooth-On, USA), which is thinned to be more flexible (Silicone Thinner; Smooth-On, USA) and colored white (White Silicone Pigment; Smooth-On, USA). This artificial sclera is 1mm thick, and has a ∼14 mm circular hole into which a silicone O-ring (AS568A-111; ∼12 mm ID & ∼16 mm OD) is bonded. A disposable direct imaging plano-concave contact lens (20 Degree Flat Vitrectomy Lens 1284.DD; Dutch Ophthalmic USA) is inserted in the O-ring (flat lens side is facing the retina) to simulate typical visual access. The retina is approximated by a ∼1 mm thick, multi-layer insert cast out of latex (Latex Paint Base; The Monster Makers, Inc.). It covers ∼60–70% of the inner eye surface and has hand-drawn vascular patterns, painted on the middle layer during the multi-step casting process. This simulated retina is inserted through the O-ring opening and placed on the back of the eye wall. The opening is also used when filling the eye with water and creates a tight seal when the lens is installed which is necessary to keep the fluid in the eye. The whole eye rests in a plastic base with a 30 mm diameter. The socket is coated with methylcellulose

[2] Thanks to Dr. Iulian Iordachita, Kevin Olds, Amrita Gupta and other students for their help with the design and fabrication of the eye phantom.

jelly to facilitate natural eye rotation and translation, while minimizing the unnatural elevation of the eyeball. Two standard surgical trocars are inserted through the sclera, one for the light pipe and the other for the surgical instrument. Figure 3.5 shows the components and the assembled version of the eye phantom. Qualitative assessment by experienced vitreoretinal surgeons verified the ability of the model to simulate realistic eye behavior in surgical conditions. The visual field of view is ∼35 degrees considering a 5 mm iris opening, which is comparable to that of a surgical case where ∼20–45 degree vitreoretinal contact lenses are used. Additionally, the latex insert is a good substrate for artificial membranes such as the liquid bandage phantom [75] or a thin layer of pure silicone adhesive painted directly on the retina insert. These simulate the membrane for an ERM peeled scenario, where the transparent ERM is difficult to visualize directly.

The same concept has been applied to fabricating a rabbit-sized eye phantom as shown in Figure 3.6. Since the eye is significantly smaller and lighter than a human eye, it is attached to the base with a rubber band tether to minimize unintentional lifting. The eye sits on a nylon ring to provide a low friction translation of the eye globe. The lens is placed directly into the sclera opening and is occasionally trimmed with surgical scissors to prevent collisions with the surgical instrument shafts.

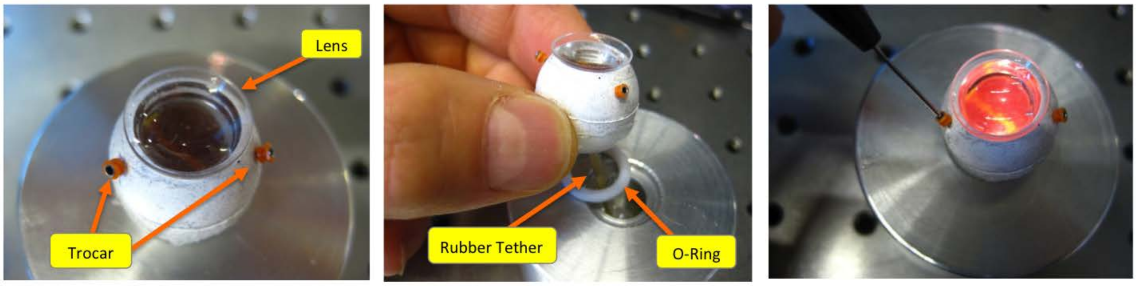


Figure 3.6: Rabbit Eye Phantom. Note: Iris is not shown.

3.3.3 Band-Aid Membrane Peeling Phantom

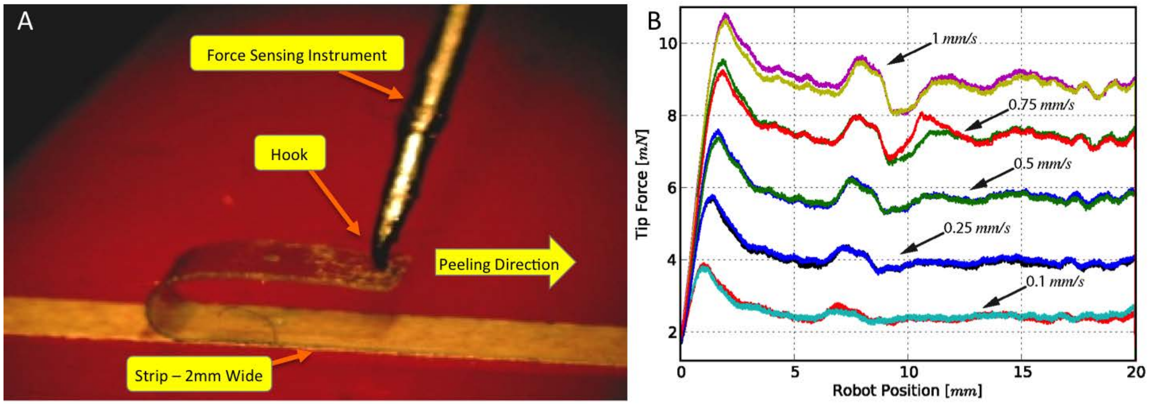


Figure 3.7: A) Band-Aid based membrane peeling phantom. B) Peeling phantom's force response to different instrument velocities.

To develop and assess the performance of force control and auditory feedback methods in membrane peeling, a consistent and easily fabricated phantom model, behaving within the parameters of vitreoretinal surgery is required[3]. The actual peeling procedure involves grasping or hooking a tissue layer and slowly delaminating it, often in a circular pattern. To reduce the factors needing to be controlled, we simplified the target maneuver to a straight line peel using a hooked instrument, and created a complementary phantom that behaved predictably. Video recordings of actual ERM peeling procedures were analyzed and tool velocities during membrane peeling were found to be 0.1–0.5 mm/s. Gupta et al. found that retinal tissue manipulation forces are likely to be below 7.5 mN [36], while Jagtap et al. found them to be comparable but slightly higher [76]. With these values in mind, after extensive search and trial-and-error testing of many materials, we identified the sticky tabs from 19 mm Clear Bandages (RiteAid brand) to be a suitable and repeatable phantom for delaminating. The tab is sliced to produce 2 mm wide strips (see Figure 3.7A) that can be peeled multiple times from its backing, with predictable behavior showing increase of

[3] Credit should be given to Dr. Iulian Iordachita and Ali Uneri for their help with developing this phantom.

peeling force with increased peeling velocity. The peeling force profile can be adjusted by varying the width of this tab. The plastic peeling layer is very flexible but strong enough to withstand breaking pressures at the tool attachment site (e.g., forceps or hook). The backing is affixed to the platform by the double-sided tape to provide traction. A ∼10 mm section of the sticky bandage tab is considered, requiring 20 mm of tool travel to complete a peel. Figure 3.7B shows the forces observed at various velocities. These were measured by commanding a robot at constant velocity in the axis parallel to the strip, while measuring the force applied to the force sensing hook held by the robot and anchored to the strip. See chapter 5 for more details on robot and force sensing.

3.3.4 OCT Imaging Phantoms

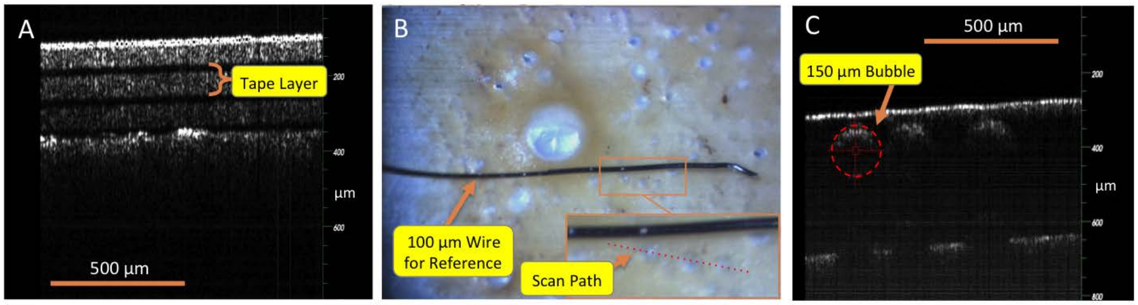


Figure 3.8: Tape on wax phantoms for OCT imaging: A) Cross-sectional OCT scan of tape phantom showing three distinct tape layers (2 bands per tape sample). B) Microscope view of the wax phantom with 100 μm wire for scale reference and multiple visible bubbles indicating the location of the cavities. C) Cross-sectional OCT scan (corresponding to the path in B)) showing distinct cavities.

A number of artificial phantoms have been developed for consistent evaluation of OCT imaging. **Tape Phantom:** For simple tests, a composite of three 60 μm thick layers of Scotch$^{®}$ tape (3M, Inc.) on a dense wooden plank is used. This provides a strong multi-peak axial scan signal that is analogous to that generated by the multilayered structures of the retina. Figure 3.8A is a B-Scan cross-section showing clear layers of this phantom.

Wax Phantom: For the targeting tasks described in Chapter 6, a model with 100–300 μm diameter cavities located near the surface is required, as well as the ability to display any physical interaction with the sample. This has been achieved with a sheet lining wax (McMaster 8691K31) with a low softening point (135°C) that is placed on an aluminum sheet and heated to boiling. Rapid cooling produces many thin-walled bubbles and any physical contact between an instrument (e.g., pick) and the sample's surface leaves permanent marks visible in OCT. Figure 3.8C shows a cross-sectional OCT scan showing clear cavities corresponding to barely-visible marks on the microscope view in Figure 3.8B.

3.3.5 Liquid Bandage Membrane Peeling Phantom

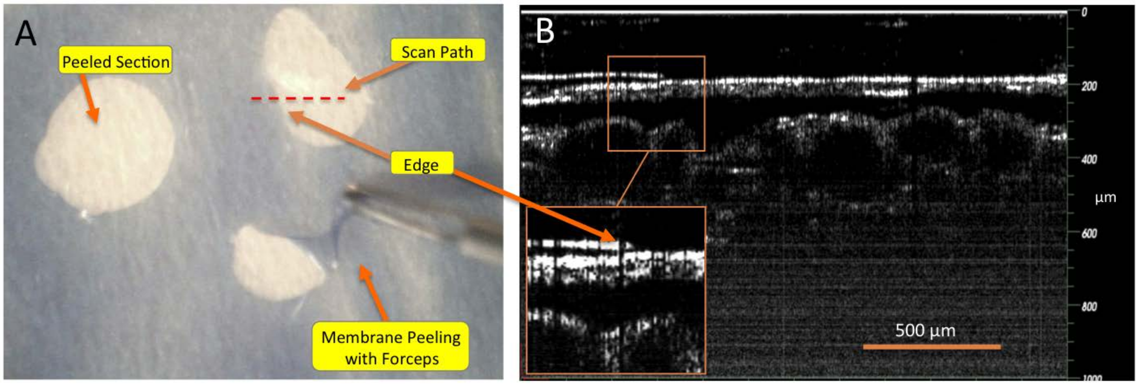


Figure 3.9: A) Screen shot of membrane Peeling of liquid-skin bandage membrane. B) OCT image showing the cross-section of the phantom with a visible membrane edge.

Another membrane peeling phantom, inspired by [75], has been employed in cases where a thin membrane structure for a non-linear peeling path is required. The membrane is created by applying a coat of liquid skin bandage (New Skin Liquid Bandage; Prestige Brands, Inc.) to standard wax paper attached to a piece of acrylic with double stick foam tape. The liquid bandage can be mixed with a dye (Blue Sharpie Permanent Marker, NewellRubbermaid, Inc.) to facilitate visualization of areas that have been peeled as shown in Figure 3.9. The thickness of this artificial membrane is roughly 20 μm. Once applied it dries within 15 minutes and tends to peel off in patches similar to human retinal membranes, although it is slightly stiffer. The liquid bandage can also be placed on the surface of the retina inserts of the eye phantom.

3.3.6 Raw Egg Membrane Peeling Phantom

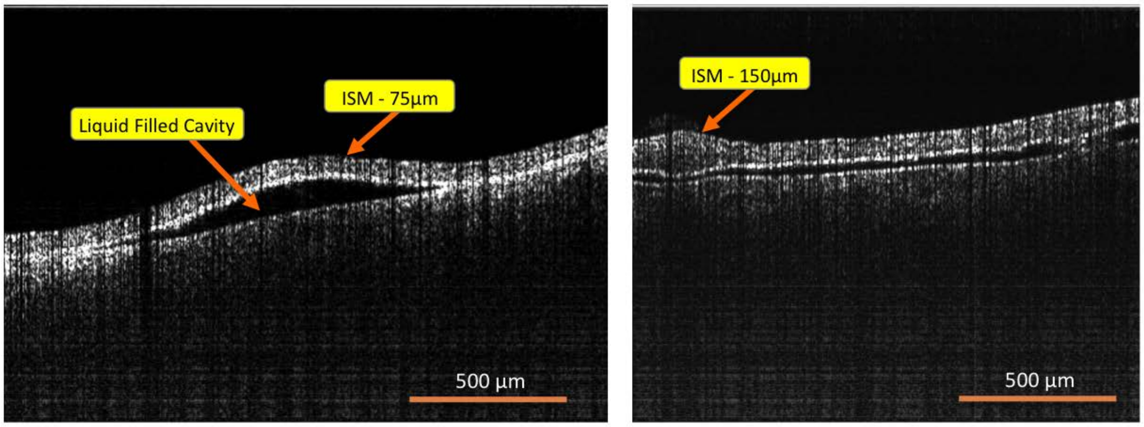


Figure 3.10: Two OCT scans of a chicken egg inner shell membrane showing membrane thickness and cavities.

The raw egg membrane peeling model uses conventional three-week-old chicken eggs acquired at the supermarket and refrigerated. The egg is cracked in half and the yolk is removed. The inner shell membrane (ISM) that is adherent to the shell wall is used for membrane peeling. It is 75–150 μm thick and is randomly detached from the shell producing small (e.g., $\sim$200×700 μm) liquid filled cavities that are clearly visible in OCT in Figure 3.10. A drop of food coloring is dropped on top of the ISM to better visualize it during removal, a similar technique to that used in human internal limiting membrane peeling. The eggs are rested firmly upright in a custom egg holder laser cut out of acrylic.

3.3.7 Chick Chorioallantoic Membrane

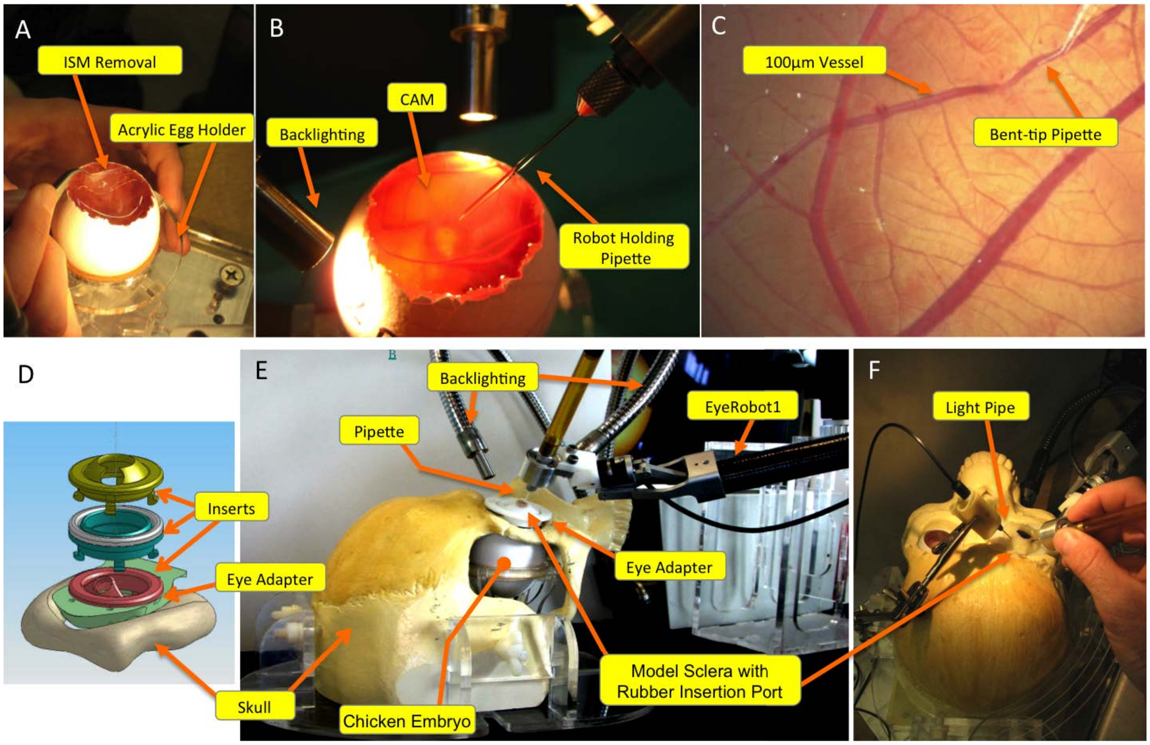


Figure 3.11: A) Preparation of chick chorioallantoic membrane (CAM). B) CAM with backlighting and pipette. C) Bent-tip pipette aligned with a blood vessel. D) Design of artificial sclera adapter for the skull. E) Cannulation experiment setup - side view showing location of the egg inside the skull. F) Same setup with a rubber cover of the sclera eye adapter and light pipe.

The chorioallantoic membrane (CAM) of an incubated chicken egg (11 to 15 days) has previously been reported to serve as an excellent phantom of the retina that is suitable for vitreoretinal surgery testing [77]. The CAM mimics the neurosensory retina because it is similar in thickness (approximately 100 μm), has its own vasculature, and if sufficiently traumatized will bleed or tear. The inner shell membrane (ISM) is adherent to the CAM and simulates an ERM that is attached to the retina, providing a convenient phantom for membrane peeling. The arborizing network of veins found in the CAM is similar to that found in the retina and can be used for cannulation and injection experiments. These vessels vary in size but contain many sections that are in the 50–100 μm target diameter. In comparison, veins on the human retina range from 40 μm to 350 μm [77].

To prepare CAM for a typical experiment, the egg shell is cracked and peeled away from the region over the air space (allantoic vesicle) that exists between the shell and the ISM at one pole of the egg (see Figure 3.11A). The 10–20 mm diameter opening provides sufficient access to the ISM for membrane peeling. For cannulation experiments the ISM is peeled away from the CAM with fine forceps. The CAM is moistened with saline solution to prevent excessive dehydration. Multiple light sources are used to evenly backlight the CAM (see Figure 3.11B,E) to increase the contrast of the vessels and injection pipette, as well as decrease the specular reflections from the surface of the CAM (see Figure 3.11C).

3.3.8 Enucleated Animal Eyes

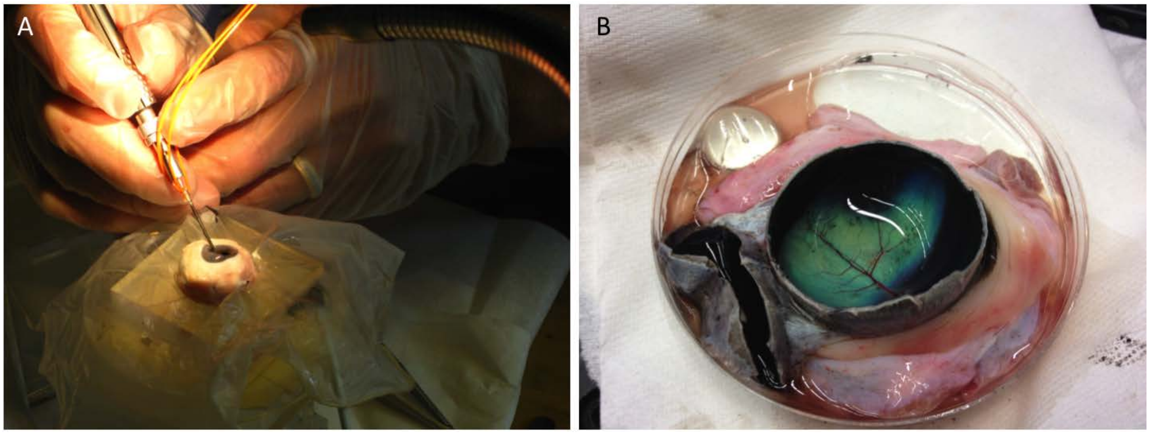


Figure 3.12: A) “Open Sky” porcine eye with cornea, lens, and vitreous removed. A force sensing forceps is inserted through the pupil. B) Bovine eye prepared for an OCT scanning experiment.

Fresh bovine and porcine eyes are common surrogates for cadaveric eyes in surgical training and research. The lenses of these eyes, acquired from a local slaughterhouse, cloud almost immediately, and as a result have to be removed to provide a clear optical view of the retina ("open sky"). The eye is pinned down to a foam block by the sclera while standard ophthalmic instruments are used to remove the lens, the cornea, and any visible vitreous (Figure 3.12A.) The retina is moistened with saline to preserve its properties for the duration of the experiment. For some experiments, most of the sclera is cut away and the distal section of the eye is splayed out in a dish to facilitate direct access (Figure 3.12B.)

3.3.9 In Vivo Experiments

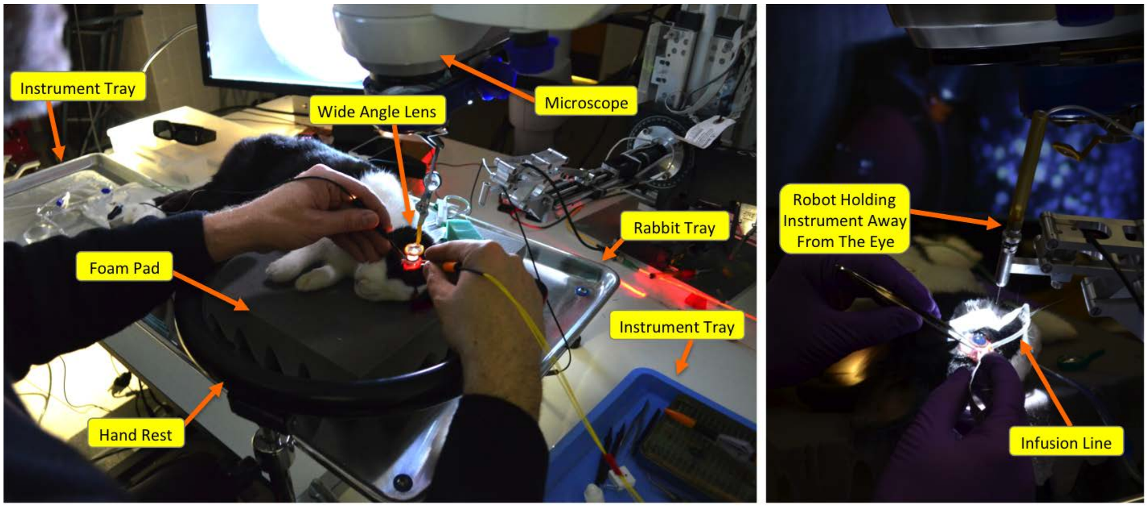


Figure 3.13: In vivo rabbit experiments.

Although human cadaveric or enucleated animal eyes are useful in many benchtop experiments, an in vivo model is required for the development of new technologies that involve physiological measurements and/or affect surgical workflow. In vivo experiments with rabbits are conducted in the OR3 (see Section 3.2) and have been approved by the Johns Hopkins Animal Care and Use

Committee[4]. New Zealand Albino or Dutch-belted rabbit breeds without any ocular disorders and with a body weight between 2 and 3 kg are used for these experiments. It must be noted that the size of the rabbit eye is 15–18 mm in diameter, or 60–70% of the human eye, and the retina has a very contrasting (white) myelinated streak, but does contain sections that are similar to the human retina, both visually and physiologically.

First, the rabbits are anesthetized with an intramuscular injection: a mixture of ketamine hydrochloride of 50 mg/kg of their body weight (Fort Dodge Laboratories) and acepromazine of 10 mg/kg (Fort Dodge Laboratories). Their pupils are then pharmacologically dilated with 1 drop each of 1% tropicamide ophthalmic solution (Alcon Laboratories) and 2.5% phenylephrine hydrochloride ophthalmic solution (Akorn, Inc.). Next, the rabbit is placed on a foam pad atop an elevated tray so that its head is lifted off the table surface to simulate the height of the human head on a surgical bed. The surgical eye is topically anesthetized with 1 drop of proparacaine (Wilson Ophthalmic) and prepped for the surgical procedure by irrigation with 5% propidium iodide sterile ophthalmic prep solution (Betadine; Alcon Laboratories). A standard 3-port pars plana vitrectomy is performed using a 23-gauge system (Alcon Laboratories) and the Accurus Vitrectomy system (Alcon Laboratories). Whenever possible both eyes are operated on in one session[5]. The typical experimental protocol is as follows[6]:

1. Fundus image of rabbit eye is taken a few days before the experiment.
2. Any equipment is moved into the OR3 a day or two before the experiment.
3. A dry run of all system components is conducted soon after setup, including essential surgical devices, e.g., Vitrectomy system. There is a status check of surgical supplies, e.g anesthetics, fluids, etc.
4. On the day of the experiment, the rabbit is anesthetized and its eyes are dilated 30 minutes before the surgical team arrives.
5. System applications are started and idling, e.g., video/audio recording.

[4] All rabbit handling has been done by Kevin Olds.

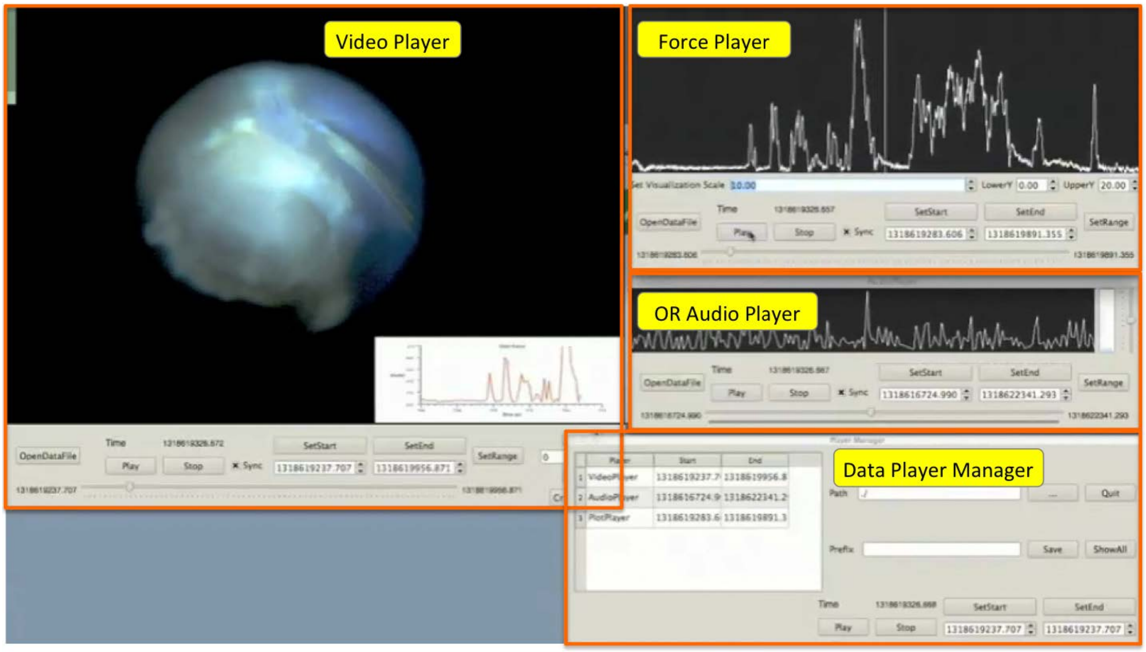
[5] The second eye is often not usable due to corneal damage.

[6] The rabbit preparation stages of the protocol were developed by Kevin Olds and Laura Pinni with oversight from Dr. Gehlbach, Dr.Handa, and Sam D'Anna.

6. The rabbit is placed on the foam pad and one eye is prepped for surgery, while the other is taped shut to prevent damage to the cornea.
7. The infusion line and two trocars are inserted into the sclera.
8. A lensectomy is performed if necessary.
   (a) If possible, tool force measurements for capsulorhexis, zonular dehiscence, and lens dislocation are collected.
9. Pars plana vitrectomy is performed.
10. Exploration of the retina, both with and without instruments, is captured in raw stereo video. It is repeated with wide angle and contact lenses.
11. Pre-op annotations are tested, e.g., registration of fundus to retina.
12. For the remaining procedures (outlined below), the system is set to record the display video (one channel), including the overlays.
13. OCT Imaging
    (a) M-Scans with handheld probe of avascular retina, vascular retina, optic nerve.
    (b) Test OCT range audio feedback
    (c) M-Scans with robot assistance of avascular retina, vascular retina, optic nerve.
    (d) Test OCT/robot based behaviors.
14. Force Sensing Instrument
    (a) Measure forces for generating retinal hemorrhage, retinal tear, retinal detachment
    (b) Image areas with OCT probe.
    (c) Test force/robot based behaviors.
15. The entire experiment is repeat for the other eye.
16. The rabbit is euthanized.
17. Proper disposal is carried through.
18. A thorough cleanup of the wet lab is conducted.
19. All data (OR photos, audio, video, OCT, force etc.) is backed up, including source code for all the system components used in this experiment.
20. A debriefing is conducted to recap any issues.

3.4 Data Collection and Review Subsystem

Data collection is essential not only in the development of the system (to produce sample data) but also for validation and archiving of procedures. One approach is to implement a centralized data collection mechanism, but that is not practical in a distributed system where bandwidth limits in communication and storage methods affect the performance of the system. For simpler devices, the system uses a *cisstDataCollector* service that is analogous to a data flight recorder for collection of relevant component data during operation. In some cases, specialized data collection mechanisms have been developed (e.g., video stream) for complex data types or where large amounts of data are to be stored[7]. The data collection scheme relies on reliable system-wide time synchronization of each computer's clock. This is accomplished with Network Time Protocol [65] providing ∼ 1 ms synchronization over a local Ethernet network, which is sufficient for the types of data compared in the system. For example, most of the users rely on the video as the reference that runs at 20–30 FPS with 30–100 ms frame time uncertainty.



Set Visualization Scale 10.00 LowerY 0.00 UpperY 20.00

☒ Sync 1318619237.707 1318619956.871

☒ Sync 1318619283.606 1318619891.355

☒ Sync 1318616724.990 1318622341.293

	Player	Start	End
1	VideoPlayer	1318619237.7	1318619956.8
2	AudioPlayer	1318616724.9	1318622341.2
3	PlotPlayer	1318619283.6	1318619891.3

Path ./

Play Stop 1318619237.707 1318619237.707

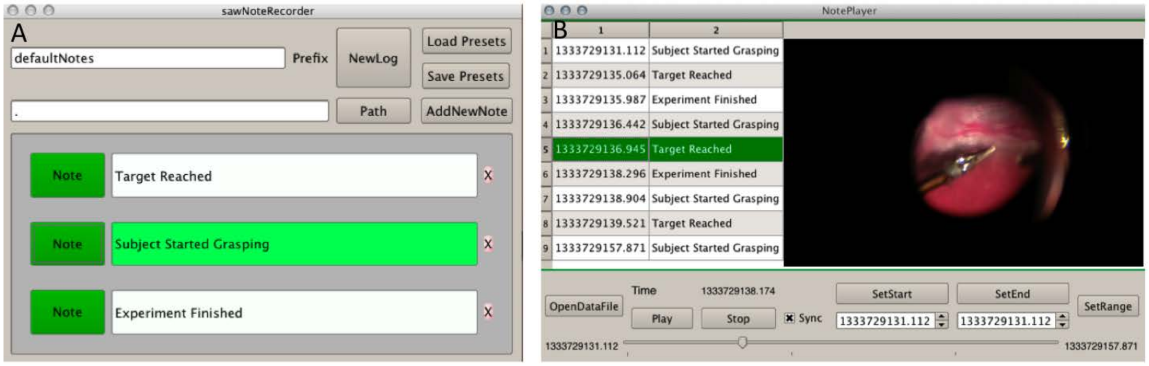
Figure 3.14: Review of in vivo force sensing experiment using *cisstDataPlayer*.

[7] It is imperative that all the experimental data (video, force, audio, photos, source code, etc.) is stored immediately after the experiment on a redundant storage system with incremental backup.

In many situations (e.g., in vivo experiments) it is not feasible to analyze the data as it is collected. Thus, a data replay tool, called *cisstDataPlayer* (Figure 3.14) has been developed by Balicki and presented in [47]. Conceptually, it is analogous to a movie editing and playback software, but with the ability to support any type of data, not just video, audio, and subtitles. *cisstDataPlayer* is conceptually close to the recently developed *rqt-bag* from the ROS framework, which is a GUI tool for visualizing, inspecting and replaying histories of ROS messages [59].

The basic *cisstDataPlayer* concept is to define data player components (plug-ins) for the different types of data (e.g., collected by the state collector, the video recorder, or other collectors), which are loaded by the Data Player Manager. All collected data is assumed to be time-stamped relative to the same system-wide time origin (e.g., 1/1/1970) and each data player component can be synchronized with the Player Manager (see the Sync checkboxes in 3.14). If they are synchronized, the Player Manager can simultaneously sequence through all data sources; if not synchronized, each data player provides independent controls for just its own data. Subsets of data based on time ranges can be output as separate files for analysis by external software, such as Matlab. Furthermore, to facilitate quick data analysis, a number of convenience widgets have been developed to assist in quantitative data analysis. For example the Video Player provides measure pixel distance tools, zoom, image cropping, and the ability to swap RGB values or change the contrast of the video frames. A note taking application has been developed to facilitate quick event logging for experimental purposes. The corresponding *cisstDataPlayer* plug-in is used to review these timestamp-synchronized events. It enables the operator to rapidly navigate long data sets, especially video and audio sequences, by directly “seeking” to a time of a note event taken during the experiment, see Figure 3.15.

The *cisstDataPlayer* implementation is elegant, only requiring the data player components



defaultNotes

.

Target Reached

Subject Started Grasping

Experiment Finished

	1	2
1	1333729131.112	Subject Started Grasping
2	1333729135.064	Target Reached
3	1333729135.987	Experiment Finished
4	1333729136.442	Subject Started Grasping
5	1333729136.945	Target Reached
6	1333729138.296	Experiment Finished
7	1333729138.904	Subject Started Grasping
8	1333729139.521	Target Reached
9	1333729157.871	Subject Started Grasping

Time 1333729138.174

SetStart 1333729131.112

SetEnd 1333729131.112

1333729131.112

1333729157.871

Figure 3.15: A) Note Recorder application. B) Note Player showing video frame corresponding to a note.

to implement a few basic functions in C++, e.g., *connectToManager, stop, play, seek, loadData, saveData*. These are agnostic to the type of data that is being managed as long as it can be placed in a timeline based on timestamps. By using the cisst/SAW components frameworks these plug-ins can be located anywhere on the network in the same or different processes as long as the machine times are synchronized. Each plug-in has a default timeline GUI with *stop, play, seek, load, save* widgets. These can be used to control the main timeline, which is effective in reviewing data based on a particular data type, e.g., video or audio cues. The data player can load a number of streams. Each data player is responsible for reporting the metadata of these streams (start, end timestamps) to the player manager. Each command (*seek, play*) within the system is sent with the local timestamp, which is used as a reference for the timing of the data rendering. For example, the play command is sent from the video player to the player manager with the current system timestamp and the timestamp of the data at the current play location. The player manager forwards this request to all connected components. It is up to the component to honor the play request, perform data timestamp interpretation and bound checks, account for any latencies, and render the data to the user. This provides the most flexibility to the developer, a requirement in the case where data types vary widely including, occasional textual note, continuous audio and video, sparse scalar force data, and large 3D OCT scans.

3.5 Conclusions and Future Work

This chapter presented two full experimental setups, the experimental protocols and specialized experimental testbeds that have been used in the development and testing of the *eyeSAW* system and associated technologies. The artificial testbeds, such as the JHU Eye Phantom, have been indispensable for testing of robot ergonomics, sensor-based robot control methods, computer vision algorithms, and for general demonstrations of the system. Future work includes a more realistic, multifunctional eye phantom with visually more realistic vessels, vessels with lumens that can be used for cannulation, and barely visible membranes that have realistic peeling properties. Mechanical and functional aspects of the retina and membrane phantoms need more fine-tuning, which is dependent on acquiring precise in vivo tissue properties.

In vivo rabbit experiments have been used not only for measuring tissue properties and testing sensors in a realistic biological setting, but also for testing the whole system in a time-constrained, high-stress scenario. The rabbit is a good model because it is relatively small, commonly available and easy to work with. However, the rabbit's size and eye spacing do not simulate the ergonomics of human surgery. Furthermore, the rabbit eye is much smaller than the human eye and contains a proportionally large lens. This requires specialized instrumentation and also makes standard vitreoretinal surgical procedures more challenging. Future development should include in vivo eye models that are closer in size to the human eye.

The Data Collection and Review subsystem has been essential to the in vivo testing of the technologies discussed in this dissertation. Next steps include implementing more data players such as OCT scans and robot kinematics display, and new data analysis tools. An exciting prospect for another future extension is the use of the Data Review framework to simulate the devices that have originally recorded the data, i.e., generating time-synchronized signals to replay the surgical procedure. This would enable device simulation for rapid prototyping of algorithms, especially those that use multiple data sources.

3.6 Recapitulation of Contributions

***eyeSAW* Experimental Setups (Section 3.1 and 3.2)** Built two experimental setups (lab and OR versions) for the development of various vitreoretinal technologies and protocols on in vivo and artificial models. These setups include microsurgery microscopes with stereo-video and 3D display capability, vitreoretinal illumination, surgical instrumentation, ergonomic workbenches, etc. These setups were essential in testing of the individual system components as well as the overall surgical *eyeSAW* system. The experimental setups are also being applied in otology research, e.g., cochlea implantation surgery and can easily enable research in other microsurgical domains. **Credit**: Marcin Balicki developed the experimental setups in collaboration with Kevin Olds responsible for general OR equipment and Balazs Vagvolgyi who managed the video-microscope development. The in vivo rabbit protocol was developed by Kevin Olds, Laura Pinni and Sam D'Anna.

JHU Eye Phantom (Section 3.3.1 and 3.3.2) A process was developed for building a realistic silicone model of the eye with interchangeable retina inserts. It improves over available commercial models in a number of ways: it is soft, filled with fluid, facilitates the use of standard surgical instrumentation, is compatible with the surgical stereo microscope, and the eye sits in a viscous cup inside of a human skull that mimics a real surgical scenario. The eye phantom is indispensable in the development of the system and has the potential of being used in surgical training. **Credit**: The general concept was developed by Marcin Balicki with help from Iulian Iordachita and Kevin Olds. Amrita Gupta and many students helped with fabrication.

Band-Aid Membrane Phantom (Section 3.3.1 and 3.3.2) A novel membrane model was created for peeling experiments. The model can be easily adjusted to provide a repeatable force vs. a velocity response. This is important in multi-subject studies where constancy between trials is essential for performance comparison. The peeling model has been effective in testing of devices, robot control algorithms, and new feedback methods for controlling mN-forces in membrane peeling. It also has potential in surgical training as a simple and inexpensive model for membrane peeling. **Credit**: The general concept was developed by Marcin Balicki and Ali Uneri with help from Iulian Iordachita.

OCT Imaging Phantoms (Section 3.3.4) A wax-based phantom was developed specifically for testing of OCT-based interventions. Unlike the alternatives, this provides barely visible subsurface targets for robot assisted tissue imaging and targets. Any interaction with the malleable surface leaves a visible mark which is used in analysis. Similar phantoms can be used for other microsurgical testing where contact detection of delicate features is required. **Credit**: Developed by Marcin Balicki.

Software Framework for Data Review (Section 3.4) Developed a novel software framework and implemented a sample application to facilitate analysis of voluminous multi-media data collected during subject and rabbit experiments. It is analogous to a standard movie editing tool, but with the ability to support any type of data, not just video, audio, and subtitles. The framework is easily extensible to include other proprietary data formats and types collected during experiments, e.g., force. This framework is generic and can be used in other research projects that have multiple timestamp-synchronized data streams. It is integrated into the open source *cisst-saw* library and available via a public software repository [48]. **Credit**: Developed by Marcin Balicki. The XY-plot widget was developed by Anton Deguet and Joshua Chuang.

Chapter 4

Robot Behaviors

One of the most important and challenging aspects of vitreoretinal surgery is safe and precise manipulation of hand-held instruments inside the eye. This is extremely difficult for the surgeon due to the challenging eye geometry, limited intraocular access, micro-scale of the target anatomical structures and miniature surgical instrumentation. Precise manipulation is further affected by surgeon's involuntary hand tremor, e.g., $\sim 100\ \mu\text{m}$ at 2–20 Hz, that is an order of magnitude larger than the surgical targets [78].

Robotic assistance has great potential to provide the needed stability and precision in micro-manipulation. Robots have been successfully applied in many surgical areas, and can become an integral part of an eye surgery system [79,80]. Furthermore, by incorporating motion constraints and real-time sensing feedback into the robot control, the robotic assistant can provide safety (e.g., reduce tool speed near the retina) and guidance (e.g., force scaling) capabilities.

This chapter presents various robotic paradigms that have been proposed for eye surgery. A description of the development of the JHU EyeRobot is presented, followed by various robot control methods and setups that are specific to vitreoretinal surgery, including variable gain control, cooperative robot teleoperation, two-robot (bimanual) setups, etc. Finally, the EyeRobot is used in a vein cannulation experiment to assess its assistance capability in a surgical skill task.

4.1 Background

A number of robotic paradigms for eye surgery have been developed to address the human physiological limitations associated with micro-manipulation in eye surgery. The most common paradigm is the master-slave system, where the surgeon controls the robotic manipulator (the slave) which interacts with the patient, from a remote master console (the master). One significant advantage of such systems is the ability to scale down the motion (1-100 times) of the operator to achieve a fine degree of tool control, inherently reducing the effects of hand tremor. The best known example is the da Vinci Surgical System (Intuitive Surgical, Inc.) described earlier. Although it is designed for general MIS surgery, and hence is too large to be practical for microsurgery, it has been used in in vivo corneal suturing [81] (see Figure 4.1A), as well as for pars plana vitrectomy, intraocular foreign body removal, and anterior capsulorhexis on porcine eyes with modified robotic instruments [82].

Although the da Vinci surgical system has been demonstrated in corneal suturing on cadaveric animal eyes [81], three limitations have been reported: inappropriate pivot point position of the remote center of motion (RCM), inadequate visualization of the endoscope compared to an ophthalmic microscope, and lack of force control and haptic feedback. Mulgaonkar et al. developed an additional micro-manipulator that can be attached to the tool shaft of the da Vinci manipulator [83]. It consists of two parallel X-Y stages and provides the correct RCM rotation. Another alternative that attaches to the da Vinci's arm is the Hexapod Surgical System [84] shown in

Figure 4.1B. It is a modified Stewart platform that provides micro-manipulation while the daVinci is used for gross positioning.

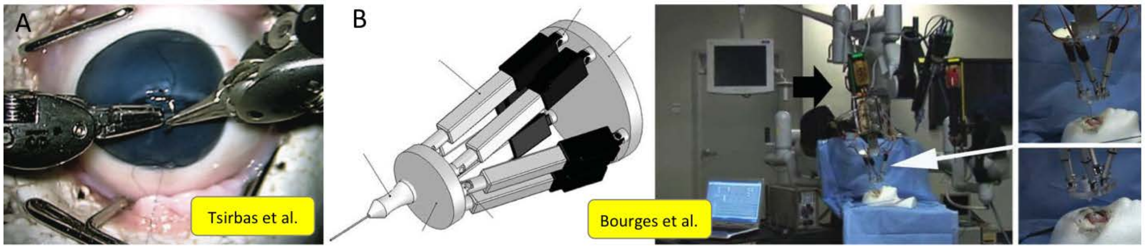


Figure 4.1: Tele-operated daVinci based retinal surgery robots.

An early, tele-robotic example specialized to eye surgery is the MSR-1 master-slave system based on a parallel-link robot architecture [85,86]. It not only incorporates motion scaling but also detects forces applied to the instruments by the environment (e.g., tissue) scales them up by a factor of 100, and relays them to the surgeon via the master. This permits the surgeon to feel forces that would otherwise be too low to perceive. [87,88] present another parallel robot designed specifically for retinal vein cannulation and is controlled through a trackball type master interface (see Figure 4.2). The system has been used to successfully cannulate 20–130 μm retinal vessels in a cat to measure intra-vascular pressure and inject medication. Nakano et al. [89] presented a compact parallel mechanism with actuated disposable surgical instruments and software simulated RCM. They showed that motion scaling has a significant benefit in a line tracing task. Wei et al. [90] proposed a theoretical system that combined a 6DOF Stewart/Gough platform with an actuated 2DOF concentric cannula end-effector providing 5DOF tool control inside the eye (most others provide only 4DOF).

The Stereotaxical Microtelemanipulator for Ocular Surgery (SMOS) is another early robot design that included a slave with spherical micromanipulator mounted on x, y, z linear stages [91,92].

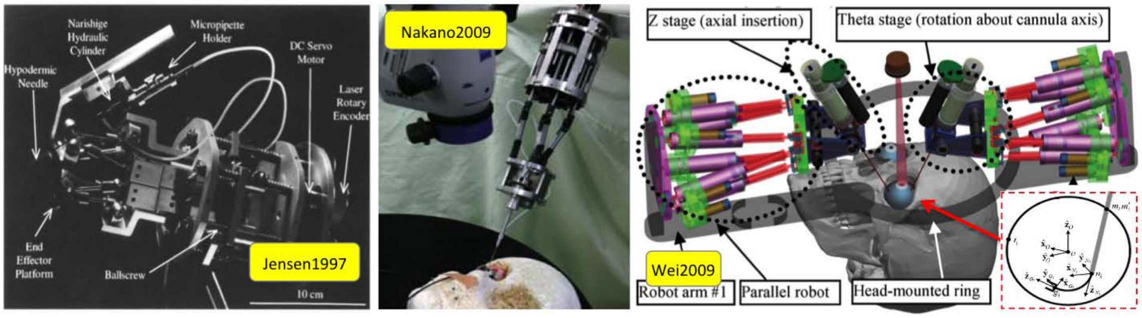


Figure 4.2: Tele-operated parallel architecture robots for retinal surgery.

A similar system developed by Yu et al. [93] has been used in animal trials for intravascular drug delivery, oxygen measurements, micro-cutting and implantation of micro-drainage devices. A more recent robot design by Ueta et al. [94] also incorporated this architecture, which has been shown to improve micromanipulation precision in simulated surgical tasks [95].

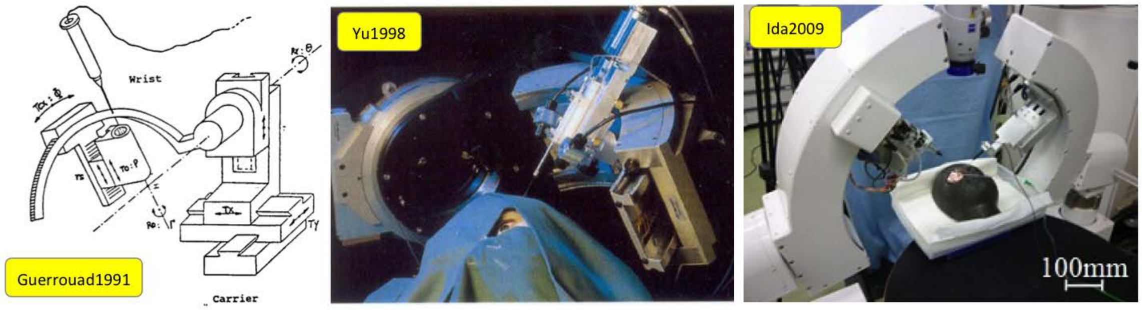


Figure 4.3: Stereotaxic architecture robots for retinal surgery.

A serial robot arm architecture (see Figure 4.4) has been used by Schenker et al. in the design of the generic Robot Assisted MicroSurgery (RAMS) system [96–98]. It includes a cable driven 6DOF robot slave and is designed to provide amplified force feedback and programmable constrained motion of the instruments. More recently, Meenink et al. designed a vitreoretinal surgery specific robotic system that integrates with the surgical table and has a mechanical parallelogram RCM [99]. It also includes an automated instrument changer that holds different interchangeable surgical instruments.

Salcudean et al. [100, 101] presented a magnetic levitation based motion scaling and force feedback robot system for general microsurgery. The master and the slave manipulators are tethered via a common stator, which according to the authors, results in a very natural control of the slave because their orientations are identical and the master is very close to the operating site. Other investigators have proposed intraocular robots that are controlled wirelessly through electromagnetic fields [102]. These are designed as targeted drug delivery vessels and have diameter of less than 500 um.

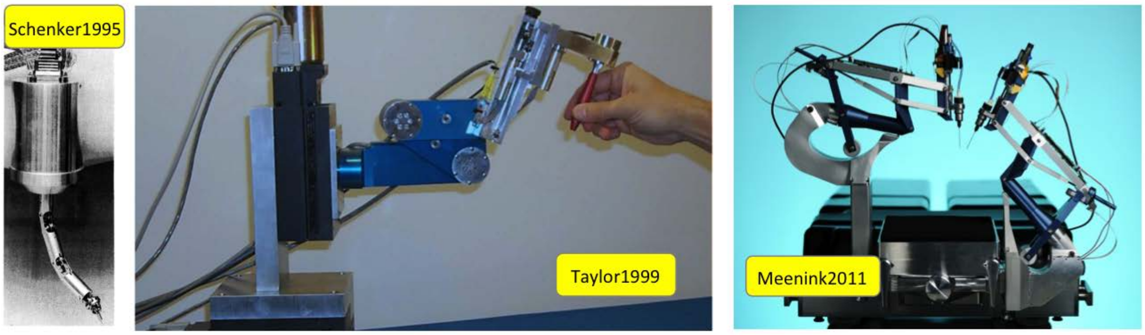


Figure 4.4: Serial robot arm architecture: RAMS system; Early version of the Johns Hopkins University Steady-Hand robot; and the eye surgery robot from Eindhoven University of Technology (TU/e).

Currently, there exist a few practical challenges associated with the master-slave paradigm. One disadvantage is the complexity and cost associated with providing two mechanical systems, one for the master and one for the slave. Another is the remote location of the surgeon and resulting lack of direct visualization which makes it difficult to perform gross motions of the robot outside of the eye, e.g., adjusting the robot base near the patient, or insertion of the tool into the trocar.

The fact that the surgeon's hands are not physically present near the surgical site requires more complex slave design to comply with stringent safety requirements (e.g., to eject the tool in case of an emergency), especially considering that the patient is often conscious and can abruptly, and without a warning, translate his or her head. Additionally, most of these tele-robotic systems rely on precise definition of the sclerotomy location to prevent excessive motion of the eye by actively constraining the tool motion to insertion and rotation about this point. This location needs to be updated often, considering that frequent translations of the eye are necessary to visualize various regions of the retina, especially under high magnification. This adds more steps to an already complex surgical protocol.

There are two alternative microsurgery robot paradigms that have been used extensively in this work. One is the Micron system developed by Riviere's group at Carnegie Mellon University. It is a hand-held micro-manipulator designed to actively remove involuntary hand tremor [78]. The surgeon holds the Micron similarly to a standard surgical tool handle, while the tool shaft is driven independently, canceling any undesirable disturbances. Hook and laser instruments have been used with this system. The Micron can also be used as a precise, high-speed, hand-held robotic manipulator. It is described in more detail below.

Another microsurgery robot paradigm used in this work is a line of cooperatively controlled robots developed at Johns Hopkins University by Taylor et al. [103]. Steady-hand cooperative control is a variant of force control where the operator and the surgeon both hold the surgical instrument. The robot senses the forces exerted by the operator on the tool handle and moves the tool to comply. Like other robots mentioned above, the physiological hand tremor is reduced, but the main advantage is that the surgeon is directly controlling the instrument that is used to operate on the patient. The combination provides the precision and sensitivity of a machine with

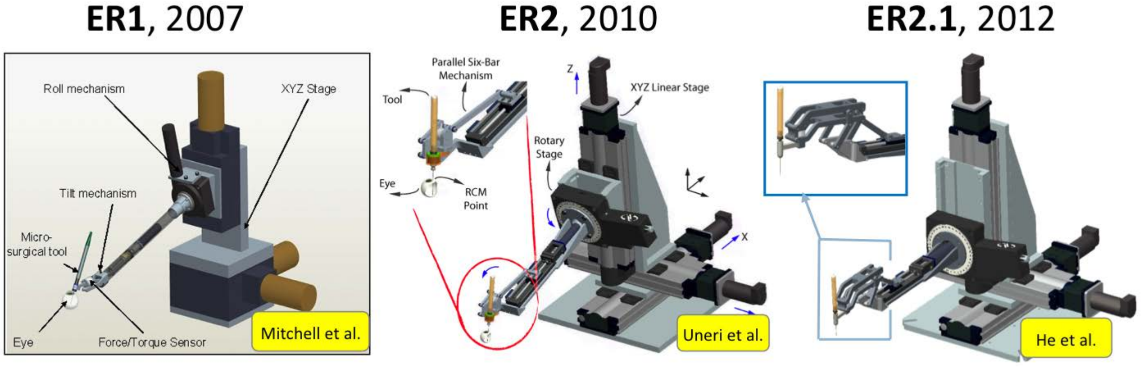


Figure 4.5: The evolution of the JHU EyeRobots.

the manipulative simplicity, immediacy and the natural hand-eye coordination of hand-held tools that the surgeons are already accustomed to.

The principal drawbacks include the inability to provide motion scaling and the loss of the ability to manipulate objects remotely. The first version of the Steady-hand robot by Taylor at al. [103, 104] is based on a 7DOF general microsurgery robot with mechanical RCM design shown in Figure 4.4. This robot inspired a line of cooperative robots for retinal surgery called the JHU EyeRobot [105–108] shown in Figure 4.5. The development of the platform is described in section 4.3.

4.2 Micron

Micron is a handheld robot designed to actively remove involuntary hand tremor and provide micrometer-scale tool tip positioning. Two versions of the device have been used with the *eyeSAW* system[1] : a 3DOF version, we call Micron3D [78, 109]; and a newer 6DOF version

[1] Many thanks to Cam Riviere for making the Microns available for our use, and to Rob MacLachlan, Brian Becker, and Sungwook Yang for assisting with Micron based experiments.

we will refer to as Micron6D [110]. Both of these devices rely on an optical position measuring system (ASAP) that tracks the position of the handle and the end-effector platform at 2kHz with 4μm accuracy [111]. It uses multiple active LEDs attached to the end-effector and the handle, and provides full 6DOF tracking of the tool pose and handle location for feedback control.

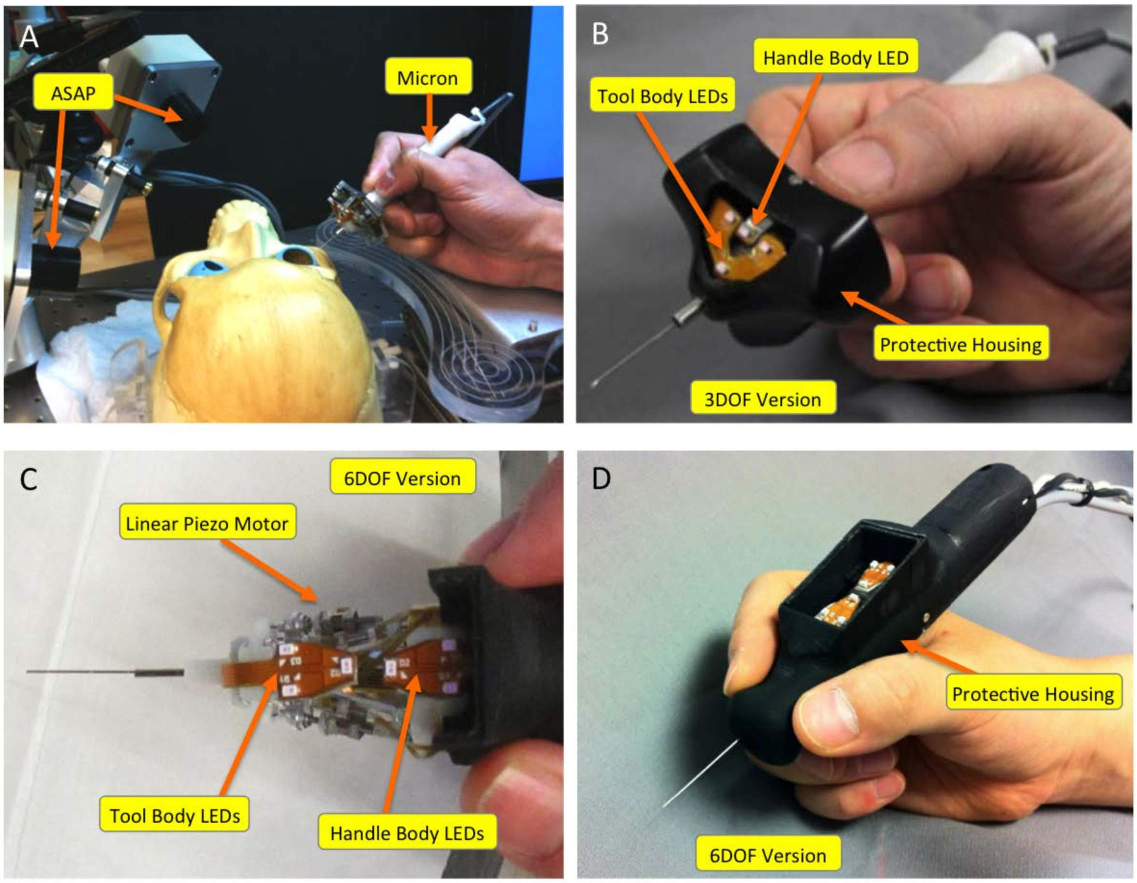


Figure 4.6: Two versions of the Micron system. A) Typical setup showing the ASAP tracking system facing the Micron. B) 3DOF Micron with four IR LEDs, three for tool body, and one for the handle. C) 6DOF Micron with six IR LEDS, three for each: tool and handle bodies. D) 6DOF Micron with housing.

The Micron3D has a ∼1 cm diameter tubular handle easily grasped by a human operator as seen in Figure 4.6. The distal end has a transversely mounted triangular plane that holds three piezoelectric stacks actuating the end-effector platform. The actuation of each stack is mechanically amplified to increase the range of motion and also increase the end effector velocities. The end-effector probe is 30 mm in length and is mounted to the platform and aligned coaxially with the handle. This configuration facilitates very fast positioning of the probe tip relative to the handle. It has 1N force capability and $\sim$100 Hz bandwidth in a work volume of approximately 1000 x 1000 x 400 μm. The location of the probe tip is established using a standard pivot calibration. The system interface is based on UDP sockets and provides real time position feedback and way point setting capability. It operates at 2 kHz with minimal latency.

Micron3D is a proof-of-concept and bench-top development prototype and was found to have a number of practical disadvantages. These include limited range of motion, inadequate controllable degrees of freedom to account for the fulcrum at the sclerotomy point, and a wide manipulator profile which is ergonomically undesirable and tends to obstruct the sight-line of the operating microscope.

These limitations are addressed by the newer Micron6D with a 6DOF actuation capability [110, 112]. Micron6D has a maximum diameter of 27.5 mm and the overall length is 130 mm, excluding the length of the intraocular tool section. When pivoting about a software based remote center of motion, the workspace for the tool tip is a cylinder 4 mm long and 4 mm wide. It uses ultrasonic linear micro-motors to actuate a Gough-Stewart platform which provides a bandwidth of approximately 40 Hz, with position error of 10-25 μm RMS varying with the tip location in the workspace. The Micron6D tolerates transverse loads up to 0.2 N applied on the tool shaft at the location of the remote center of motion.

Micron provides two default behaviors: a) tremor cancellation via a low-pass filter which attenuates the tool tip response above 1.5 Hz; and b) motion scaling, which is approximated by a shelving filter to provide a flat shelf near 1 Hz with a gain of 0.33, i.e., input motions are scaled down by a factor of 3 while higher frequencies considered to be hand tremor are minimized.

A *cisst-saw*-based wrapper (mtsComponent) has been developed for standardized system interoperability. Although many of the following behaviors can be implemented on the Micron, a number of OCT based Behaviors (see 6) are more suited for this platform and are discussed in later chapters.

4.3 EyeRobot

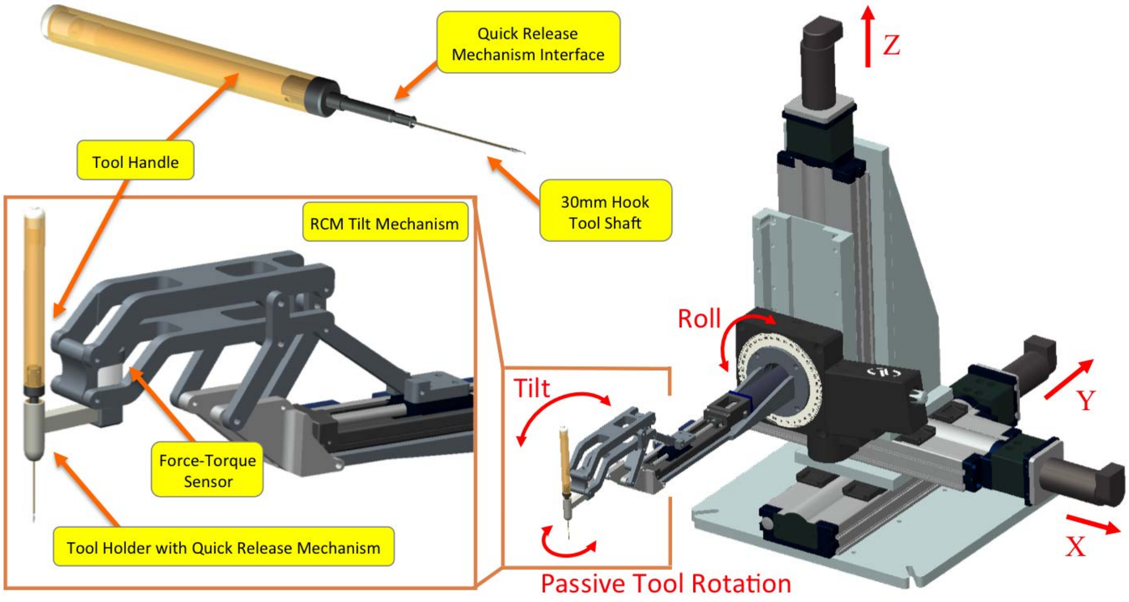


Figure 4.7: Schematics of the JHU EyeRobot.

The design and fabrication of the EyeRobots is an ongoing collaboration between engineers, scientists and surgeons. [2] For this project, two major iterations of the Eye Robot were designed

[2] EyeRobot1 was developed by Dr. Iulian Iordachita, Dr. Russell Taylor, and Ankur Kapoor. EyeRobot2 was

and fabricated, and extensively used in phantom and in vivo experiments presented by Uneri, He, and Balicki et al. [107,108]. Based on this experience the latest version (Eye Robot 2.1) addresses mostly safety and ergonomics requirements and is presented here.

4.3.1 Robot Design Requirements

The specification for EyeRobot design are based on requirements gathered from general patient anatomical structures, clinical safety and ergonomic constraints, surgeon input, and direct experimentation with previous systems. The resulting robot specifications with respect to the motion of the tip are summarized in Table 4.1.

Retinal surgery requires extremely fine tool positioning ability. Therefore an emphasis is placed on the ability for the robot to move the tool tip by fine increments rather than provide precise absolute position measurements. This is important because the tool shaft is often in contact with the sclera or intraocular tissue, generating forces on the flexible tool shaft that cause significant deflection. As a result, the estimate of the tool tip location through forward kinematics can have an error as high as 1 mm for a 20 mN force applied at the tip as shown by Uneri and Balicki et al. [107]. In practice this disparity is accounted for by the surgeon who closes the control loop by guiding the robot directly to the target based on visual feedback of the tool tip. To provide this requisite precise incremental motion capability high stiffness, low backlash and low stiction need to be considered in the mechanical design of the robot. This also minimizes the deflections generated by the user applying a force on the tool handle and enables stable control loop behavior.

Through extensive hands on robot testing with expert surgeons, it was found that the

developed by Dr. Iulian Iordachita, Dr. Russell Taylor, Dr. Peter Kazanzides, Marcin Balicki, with input from two surgeons: Dr. Handa, Dr. Gehlbach. The RCM mechanisms were designed by Dr. Iulian Iordachita, and Daniel Roppenecker. The tool quick release mechanism was designed by Dominikus Gierlach. The RCM mechanism for EyeRobot2.1 was developed by Dr. Iulian Iordachita and Xingchi He. Many thanks to Jason Hsu and Robert Romano for assisting with the assembly of the EyeRobot2.

surgeons move tools surprisingly fast (>50 mm/s) when the tip is in the central (safest) region of the eyeball; and even faster outside of the eye. These speeds, although slightly lower, are in agreement with previously published measurements [35]. Since most of the retinal procedures also involve a few extremely slow and delicate maneuvers (e.g., 10 μm/s), to be practical, the robot speed range has to accommodate both of these scenarios.

Although the work volume inside the eye is relatively small ($<$ 25 mm), the robot requires a much larger overall work envelope. This facilitates a variety of arrangements of the robot near the surgical site which consider the patient related and surgical equipment geometric constraints. Also this minimizes gross adjustments during procedures that could be cumbersome and take up valuable time. The travel limits of the tool roll and tilt axes are also important due to the constraints in approach to the distal retinal regions that have to consider the sclerotomy, the microscope view axis, and the robot workspace position, while avoiding contact with the lens. These are roughly based on the published tool motion ranges measured by Hubschman et al. [113]. Figure 4.8 shows the motions of the tool inside the eye, and the resulting access to the retina based on the limits in Table 4.1.

A major barrier to acceptance of robotic assistance by our vitreoretinal surgeon collaborators is the ability for the instruments to be rapidly retracted from the eye in case of emergency. This mechanism should mimic the existing methods where the surgeon ejects the tool from the eye by simply rapidly moving it away, along the tool shaft axis. Additionally, the robot should also have quick and hassle-free tool change capability. Lastly, the robot ergonomics should allow access to the eye across the bridge of the nose avoiding any contact with the patient. This is important in cases where two robotic assistants are used simultaneously, either for bi-manual operation, or where one or two robots are used as semi-autonomous assistants, e.g., light pipe holders.

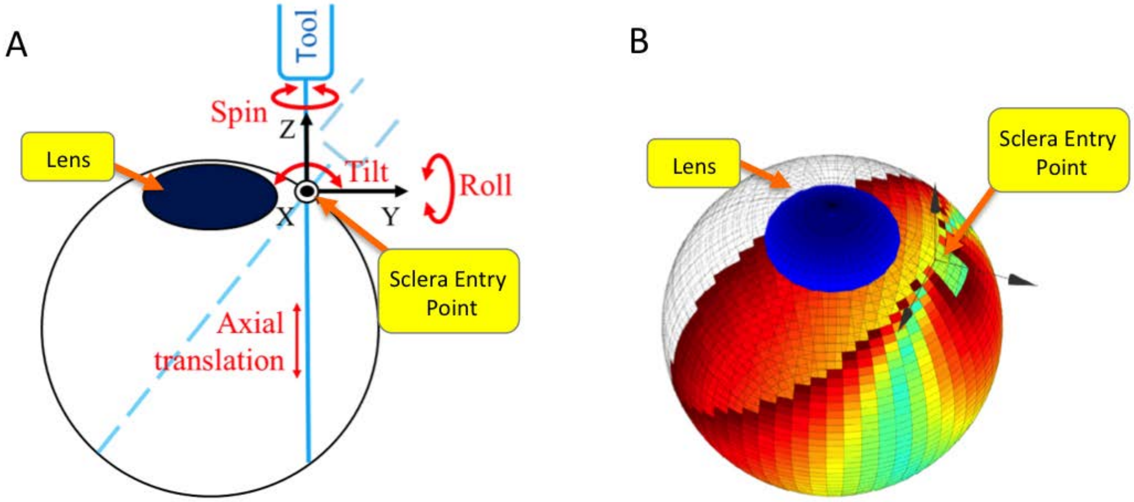


Figure 4.8: A) Schematic of the rigid tool motion inside of the eye constrained by the sclerotomy (sclera entry point). B) Tool tip reach based on ±45° tilt and ±180° roll limits. Darker colors depicts high absolute value of the tilt angle and no color depicts inaccessible areas.

4.3.2 EyeRobot Design

The resulting robot manipulator consists of four sub-assemblies: 1) XYZ linear stages for translation; 2) a rotary stage for rolling; 3) a custom tilting mechanism with a mechanical RCM; and 4) a quick release tool adapter with a handle force-torque sensor. Parker Daedal 404XR linear stages (Parker Hannifin Corp.) with precise ball-screws are used to provide 100 mm travel along each of the XYZ axes with a bidirectional repeatability of ±3 μm and positioning resolution of ±1 μm. A rotary stage (URS 100B, Newport Corp.) is used for the roll axis, with a resolution of 0.0005 degrees and repeatability of 0.0001 degree. A THK KR15 linear stage (THK America, Inc.) with travel of 100 mm and repeatability of ±3 μm is used to actuate the custom-designed symmetric parallel RCM mechanism for the tilting motion. The RCM tilt mechanism can be used in left or right handed fashion and provides sufficient workspace and stiffness (~20 - 50 N/mm, depending on loading direction), as well as improved kinematics characteristics. It is described in detail by He

Table 4.1: EyeRobot design specifications.

Robot Specification	Value
XYZ Limit	±50 mm
XYZ Incremental Step Size	2 μm
XYZ Speed	50 mm/s
Roll/Tilt Limit	±120°/±45°
Roll/Tilt Speed	40°/s
Roll/Tilt Incremental Step Size	0.01°
Handle Force & Torque range	5 N & 0.12 Nmm
Handle Force & Torque resolution	0.003 N & 0.015 Nmm

and Balicki et al. in [108]. None of the actuation assemblies are back-drivable which is an important requirement for admittance type control schemes. Furthermore, all the translational stages should have the same acceleration and velocity performance to provide naturally symmetrical feedback, and the same applies to the rotational stages.

A 6-DOF ATI Nano17 force/torque (F/T) sensor[3] (ATI Industrial Automation, Inc.) is mounted between the RCM and a custom-designed, very compact, tool holder with a quick release mechanism for surgical instruments. The quick release mechanism enables the surgeon to perform

[3] It is important to use minimal filtering (Kalman) on the strain gauge signals to remove measurement noise, and use more heavy low-pass filtering on the resolved force torque values. One such filter that seems to work well for this application is the adaptive 1–Euro filter [114] that adjust the band-pass filter frequency based on the input intensity, e.g., higher input values, yield less aggressive filtering. Note that individual force and torque values are independently converted to velocities and therefore can have (non-linear) adaptive filters with different gains.

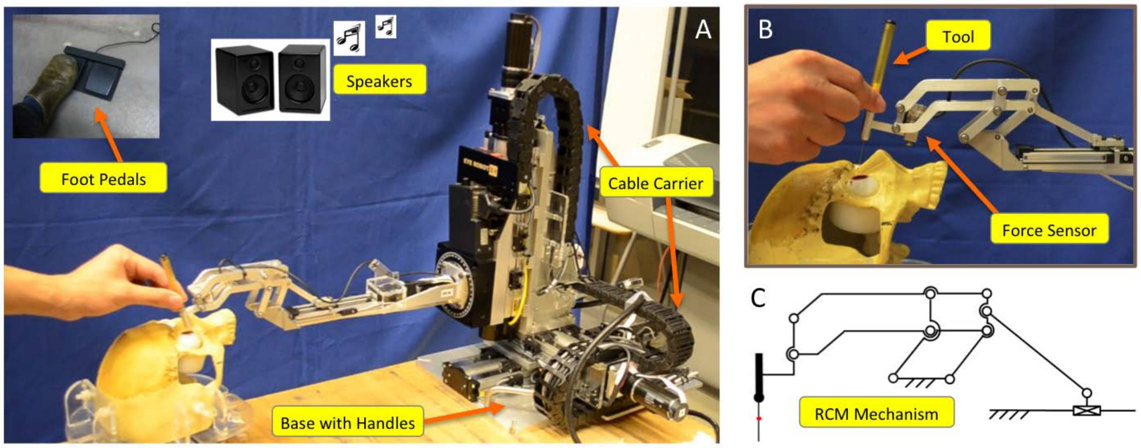


Figure 4.9: JHU EyeRobot2.1 A) Robot controlled is controlled by holding the instrument with the hand while control gains are adjusted with the Pedals. B) Close-up of the distal end of the robot. C) Schematic of the RCM mechanism.

a rapid retraction of the instrument from the patient's eye during an emergency. It also provides a convenient way to change between different surgical instruments. More details on the design and behavior of this mechanism is in [108]. Robot compatible tools are either completely custom made, as in the case of a simple hook, or are based on a modified conventional surgical instrument, such as a micro-forceps. For a typical instrument, the shaft length is about 30 mm, while the robot's mechanical RCM location is 25 mm proximally from the tip along the tool shaft. The force/torque is resolved at the tool handle 40 mm above the RCM.

An off-the-shelf 8-Axis motion controller (DMC1886; Galil Motion Control, Inc.) is used to drive five actuators, the electromagnetic brake on the Z Axis, and also to digitize the voltages from the ATI force/torque sensor. The motion controller runs a real-time PID loop at 1000 Hz and communicates via dedicated 100 MBit Ethernet connection with a standard PC running LinuxOS. The PC runs a custom robot application based on the CISST libraries [47,48,66] for easy interfacing with the *eyeSAW* environment. Its high-level control loop is updated at 200 Hz and is described in more detail in the following section. To further facilitate rapid prototyping a scripting (Python) library has been developed to control robot parameters in real-time, to write simple routines for instrument calibrations, to test performance, and for behavior development. A connection "watchdog", implemented on the motion controller, stops all actuators if the connection to the EyeRobot computer is lost.

To enhance user experience, robot warnings and errors are rendered aurally in short words in English. These included travel limits ("LimitX", "LimitY", ...), excessive forces on the F/T sensor ("Force Saturated"), or system errors ("E-Stop On"). Without this feedback the occasional travel limit encounters would cause frustrating operator confusion, since the robot would fail to cooperate as expected for no apparent reason. This simple feature was very practical in bench-top experiments and dry runs in the operating room.

A dual foot pedal with potentiometer based variable input has been incorporated into the robot system and used extensively in all the development and experiments. Its use is described in section 4.3.3.

4.3.3 Cooperative Robot Control

The Eye Robot's cooperative control concept is a form of admittance control where the robot is commanded to move with velocity proportional to the force applied on the robot by the operator [103, 104, 115]. In other words, the robot minimizes the force being applied to it by moving in the direction of the force. This enables very stable and slow motions and is an excellent control architecture for creating very stiff virtual walls. One major drawback comes from the fact that non-backdrivable robots that are ideal for this type of control tend to have inherently high impedance. This makes rendering totally transparent (low-impedance) interaction very difficult, i.e. tool manipulation with the admittance-type robot control appears dampened and generally requires more force than handling the tool directly. Figure 4.10 shows the typical dual feedback loop for the EyeRobot.

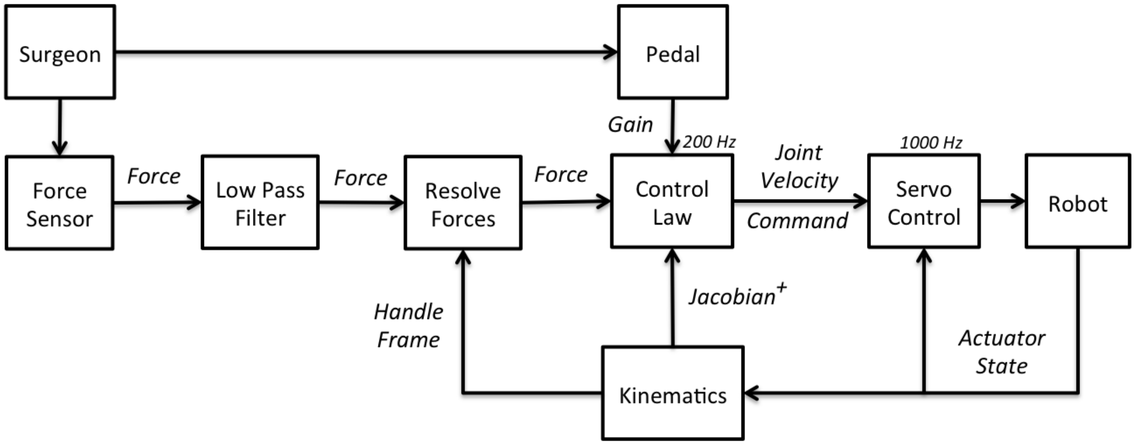


Figure 4.10: Block diagram of a typical EyeRobot control loop.

In general, the high-level admittance control operates as follows. Given sensed handle forces and torques $f \in \Re(6)$ (in N & N·m) exerted by the operator and expressed in the robot coordinate frame, a handle velocity screw $v \in \Re(6)$ (in mm/s & rad/s) is computed in the following way:

$$v = Kf \tag{4.1}$$

The matrix $K \in \Re(6 \times 6)$ represents the admittance gains applied to the sensed forces and torques, and thus permits fine tuning of the relative motion response of the inputs. In practice, K is a scalar matrix with the same value (k) in all six elements of the diagonal (zero elsewhere). The values for k have been in the 3-30 range depending on surgeon preference and the requirements of the given task[4]. Some surgeons prefer a snappier response from the tilt/roll axes in which case the

[4] The gains are in the same magnitude range because the torque is in N·M, otherwise the k_τ corresponding to the

gain for the torque values of the input is increased. The desired handle Cartesian velocity v is then converted into joint velocities ($\dot{q}$) using the pseudo-inverse of the handle Jacobian (J_h^+) :

$$\dot{q} = J_h^+ v \tag{4.2}$$

The elements of $\dot{q}$ are converted into corresponding actuator velocity units and immediately commanded to the low-level PID velocity servo loop.

Pedal Control

The pedal input can be used in many ways to alter the response of the robot to make the surgeon-robot interaction more intuitive, safer for the patient, and provide a way to make real-time adjustments in highly-variable conditions. The following two methods have been devised and implemented.

Pedal Gain Control (PGC) incorporates the variable pedal input to linearly modulate the gains K of the cooperative control algorithm from 0 to maximum K_{max} value where g_p is a scalar from $0.0 - 1.0$, corresponding to how far the pedal is pressed.

$$K = g_p K_{max} \tag{4.3}$$

This effectively changes apparent robot impedance and hence the force required to guide the instrument at a given velocity. For example, for fine manipulation the surgeon may press the pedal down only 10% of maximum, which lowers the gain and significantly reduces the effects of hand tremor, drift, and provides finer resolution tool positioning. This simple innovation gives the surgeon the ability to trade off between the fast and imprecise motion that is associated with little effort (force input), and the precise micron-level positioning of the tool tip that requires more effort and time.

torques will need to be corrected, e.g., for torques in N·mm the $k_\tau = 10^{-3} k_f$, where k_f is the force gain.

Furthermore, PGC is very intuitive to use because it has a feel similar to the automobile accelerator pedal. PGC is an alternative to nonlinear gains presented by Kazanzides et al. [115].

Pedal Speed Limit Control (PSLC) uses the variable pedal input to dynamically set the maximum speed generated at the handle. This renders a more transparent control especially at lower speeds, which tends to lower fatigue while still providing stable motion. Compared to PGC, PSLC is considered less intuitive, especially during initial use. The PSLC uses the pedal input g_p to limit the velocity in a given moment, up to a preset maximum $v_{max} \in \Re(6)$, otherwise the standard force control is used[5]:

$$v = \begin{cases} g_p v_{max} & \text{if } Kf > g_p v_{max} \\ -g_p v_{max} & \text{if } Kf < -g_p v_{max} \\ Kf & \text{otherwise} \end{cases} \tag{4.4}$$

In practice these pedal-based controls are very practical, especially in managing the state of the robot when the surgeon is not holding the instrument, i.e. when the robot should not be moving. The FT sensor is very sensitive and any environmental contact with the tool generates forces that can inadvertently command robot motion. This can cause undesirable damage to the patients or to the OR equipment. Furthermore, gravity compensation is generally not implemented, and would be impractical due to the previous reason, just as is simple thresholding of the FT data. When using pedals, the surgeons are sure that when the pedal is not pressed, the robot will not move. In fact, this feature was often used in experiments by the surgeons to temporarily rest their hands without the need to remove the instrument out of the eye. It should be noted that the particular force sensor used in the EyeRobots could go into state of full saturation where the readings are not valid. In this case the robot control would use the last valid FT input for a short period (e.g., 0.5 sec) and decelerate after that to a full stop, unless a vathe FT input reading is

[5] Note that each element of the vectors is considered independently.

available again. This ensures an intuitive response when the readings when the surgeon applies a large load on the sensor. It also protects the force-sensor from possible permanent damage, by having the robot travel in the direction of the FT input, which reduces the extreme force/torque that was applied by the surgeon.

4.3.4 Virtual Fixtures Framework

The above control algorithm works well in the simplest of scenarios but is insufficient in cases where a robot assistant needs to simultaneously consider limitations of the device, user preferences, specific task requirements and conditions. In a surgical setting, the task requirements are characterized by restricted access to the workspace as well as constrained manipulation of a surgical tool. A class of robot control algorithms called virtual fixtures (VF) can augment surgeons' abilities to provide safety and precision. A number of these have been presented in literature [116–118], with some works directly applicable to vitreoretinal surgery [119, 120]. However, most of them implement a simple virtual fixture that either restricts tool motion in a certain region in the workspace or provides haptic guidance in path following or targeting. Often these VF are based either on a specific robot type or on a specific task. A more flexible robot control framework introduced by Funda et al. [121] provides an optimal motion control method to control both redundant and deficient robotic systems in constrained working volumes. Kapoor et al. [122, 123] extended Funda's work by developing a method to generate complex virtual fixtures from standard primitives that combine user input and are directly applicable in surgical assistant robots.

This VF framework is used extensively in this work. It is based on setting up a linear optimization problem with explicit specification of objectives and constraints defined in incremental steps. The constraints can be geometric, such as a restricted geometric workspace (e.g., RCM) or a desired geometric relation between two or more task frames. They can also include robot specifications; Cartesian and joint position limits; velocity constraints; etc. In each control loop iteration, all of these are represented in terms of incremental joint motions Δq and included in a least squares optimization with the following objective function:

$$\begin{aligned} &\underset{\Delta q}{\arg\min} && \|J(q) \cdot \Delta q - \Delta x_d\| \\ &\text{subject to} && \Delta q_{up} \geq \Delta q \geq \Delta q_{low} \end{aligned} \tag{4.5}$$

where the Δx_d vector is the desired incremental Cartesian motion (desired input), and $J(q)$ is the corresponding Jacobian[6]. The optimization solver generates the best set of Δq that will minimize the two-norm motion error. These are then converted into velocity or incremental position goals and sent to the actuator servo control loop. The formulation includes a set of constraints, with simple joint velocity limits shown in the above equation.

The VF framework assumes that the result of the optimization, the incremental joint motion, can be executed by the robot, within the velocity limit constraints. However, in less dynamic robots, such as the early EyeRobots, the accelerations required to achieve two consecutive incremental motions that are significantly different may not be within the specifications of the device. In contrast, in more dynamic systems, like the latest EyeRobot2.1, for stability and safety reasons it is desirable to control the commanded joint accelerations to fine tune the robot performance. To provide an additional level of control, the formulation above is extended with an acceleration limit constraint (see Equation (4.6)) where Δq_{prev} is the measured incremental motion resulting from the previous loop iteration and the a_{up} and a_{low} are constants for the upper and lower incremental motion acceleration limits. The acceleration limits can be tuned and assigned once, or dynamically

[6] The framework is very flexible allowing multiple objectives (e.g., from other task frames) to be combined in one objective function

adjusted per task requirements.

$$a_{up} \geq \Delta q - \Delta q_{prev} \geq a_{low} \tag{4.6}$$

The VF framework is used to implement the EyeRobot's admittance control law in the objective function with Δx_d being the desired Cartesian velocity at the handle[7] (from Equation (4.2)) with the corresponding Jacobian J_h:

$$\underset{\Delta q}{\arg\min} \quad ||J_h(q) \cdot \Delta q - \Delta(Kf)|| \tag{4.7}$$

In practice, there may be instances where the solution to the optimization problem is not feasible due to conflicting constraints for a given state of the robot and task goals (e.g., the velocity limit prevents a large incremental motion required for the robot to return to a safe zone; this is typical when the robot can achieve high velocities but has low acceleration). One way to address this is to use soft virtual fixtures that incorporate slack variables and weighting factors that define the "softness" of a constraint. This provides some compliance in the robot response and ensures a feasible solution. Examples of this are fully described by Kapoor et al. in [122, 123]. Another approach is to detect the conflicting constraints and execute an alternate control model that relaxes or eliminates some of these constraints. Whenever using virtual fixtures special attention needs to be given to safety and to intuitiveness of the robot response for a particular task.

4.4 Virtual RCM Behavior

Although the robot architecture contains a mechanically constrained RCM, in practice this location does not always align with the required RCM point. One example is the use of the

[7] Note that desired Cartesian velocities need to be converted into incremental Cartesian motions by multiplying by the control loop period ($v \cdot period$). The Δ is omitted in many off the following equations to improve readability.

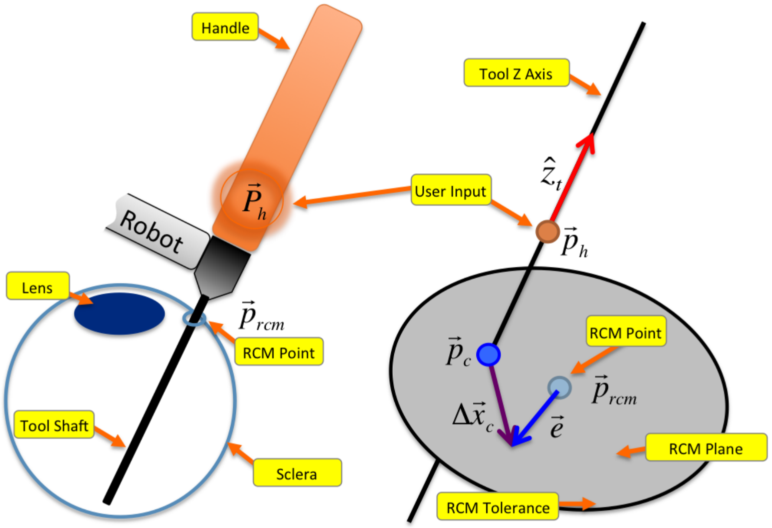


Figure 4.11: Geometry of Remote-Center-of-Motion virtual fixture used in the Motion Constrained Control Framework.

robot with human sized tools inside much smaller rabbit eyes. With the mechanical RCM being 25 mm from the tool tip, the RCM would be 5–10 mm away from the sclerotomy location above the rabbit eye. To facilitate various locations of the RCM point, a virtual RCM (vRCM) implemented in software using the above VF framework has been implemented in [105]. The following constraint is incorporated into the optimization formulation:

$$||p_c + J_c(q) \cdot \Delta q - p_{rcm}|| \leq \epsilon \tag{4.8}$$

where p_{rcm} is the desired location of the virtual RCM in the robot coordinate frame and p_c is its closest point on the line described by point p_h on the tool shaft with direction $\hat{Z}_t$ as shown in Figure 4.11. To maintain the vRCM constraint, the desired incremental motion (ΔX_c) of the tool in the RCM plane needs to remain within the tolerance radius (ϵ) of the virtual RCM point. Thus, the length of vector $e = p_c + \Delta X_c - p_{rcm}$ needs to be less than this tolerance value[8]. The incremental

[8] The RCM plane is transverse to the tool axis and passes through P_{rcm}. In terms of robot response over time, this VF ensures that the tool shaft will stay within a sphere of radius ϵ of the RCM point

motion is expressed as joint motion using the manipulator Jacobian $(J_c(q))^9$. The RCM constraint is linearized to be suitable for a linear optimization framework as described by Kapoor [123].

Kapoor also describes how the virtual RCM can be converted into a soft constraint that adds compliance to the robot behavior. From the surgeon point of view, the fixed sclerotomy has a tolerance region within which the tool motion the surgeon experiences has no resistance generated by the VF, as with standard vRCM. Outside this region, the surgeon experiences spring-like resistance, which is proportional to the distance of the tool shaft from the inner region. In this region the apparent force needs to be overcome if the tool is to travel away from the RCM point. Moreover, no motion that moves the tool outside of the outer region is permitted. This soft RCM is used in vein cannulation experiments described in later sections.

In practice, the simplest way to define a fixed virtual RCM location (p_{rcm}) in the robot's coordinate frame is by guiding the robot so the tool tip is just inside the trocar, effectively using the tool tip as a pointer. The RCM tolerance radius is selected based on surgeon preference and task requirements but, in general, it is between 0.25 mm and 2 mm.

4.5 PseudoRCM Behavior

During conventional surgical training, the vitreoretinal surgeons are instructed to produce minimal translation of the sclera during fine maneuvers. In most robot-assisted systems this is implemented with a fixed RCM location as described in the previous section. This approach is applicable when the patient is very still and the surgical task requires very fine manipulation in a small confined region of the retina. However, a significant portion of the conventional retinal procedure involves the rotation of the eye by translating the tools inserted in the trocars. This

[9] Only the first three rows of the Jacobian matrix are used

technique is needed to translate the very narrow field of view that is limited by the small iris opening, the optical properties of the eye, and high microscope magnifications, especially when a contact lens is used. Naturally, this rotation continuously changes the location of the trocars (or the intended RCM point) which can translate as far as 12 mm [113].

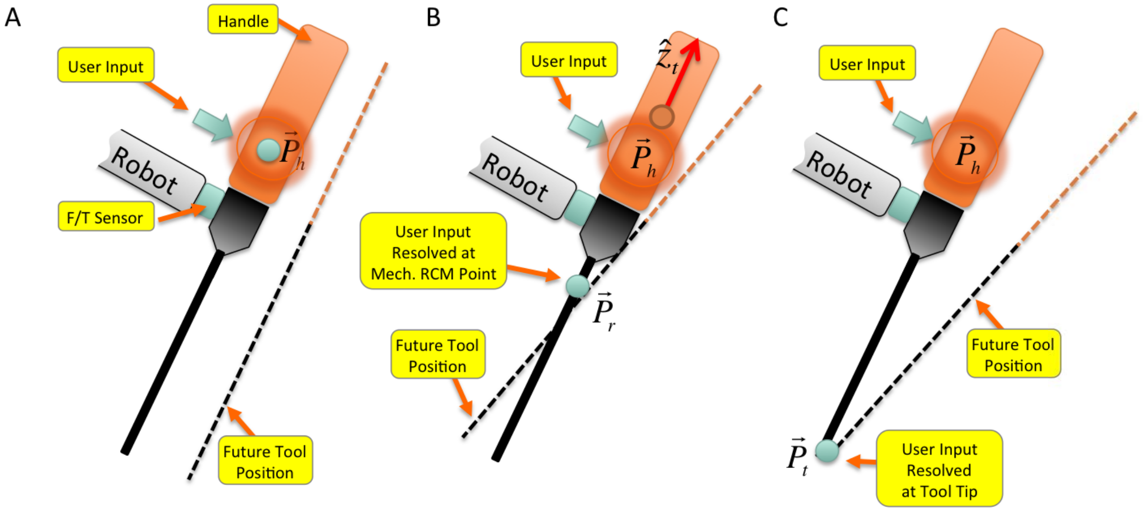


Figure 4.12: Schematic of PseudoRCM Behavior. A) The standard cooperative control where force/torque (F/T) are resolved at the handle input point and the corresponding Jacobian (J_h) is used to move the tool accordingly (pure translation). B) The F/T input at the handle is resolved at P_r which is 25 mm from the tip, towards the handle, along the tool shaft. The corresponding J_r is used to move the tool, which is mostly rotation due asymmetric admittance gains favoring torques applied at point P_r. C) Same as B) but applied at the tip, P_t.

One way to provide both stable RCM-like robot interaction and immediate ability to rotate the eye is to alter the way the force/torque ($f \in \Re(6)$) input is interpreted in the cooperative control algorithm. This "Pseudo RCM" control method resolves the f applied at the handle at the estimated RCM point on the tool shaft (using the adjoint transformation ${}^{h}Ad_r \in \Re(6 \times 6)$), which is normally 25 mm above the tool tip and roughly corresponds to the typical distance from the tip near the retina to the trocar[10]. The resulting force/torque is transformed into the robot coordinate

[10] Naturally, one has consider the units: if torques are in Nm then ${}^{h}Ad_r$ should be in meters. The ${}^{h}Ad_r$ is a local transformation and in general it only considers the translation along $\hat{Z}_t$ axis.

frame using adjoint transformation ${}^{r}Ad_R \in \Re(6 \times 6)$ and used as input in the standard admittance control[11]. The corresponding Jacobian is computed at this PseudoRCM point (J_r) and used in the following VF framework objective function:

$$\arg\min_{\Delta q} \left\| J_r(q) \cdot \Delta q - K \cdot {}^{r}Ad_R \begin{bmatrix} K_f \\ \\ K_\tau \end{bmatrix} \cdot {}^{h}Ad_r \cdot K_z \cdot f \right\| \tag{4.9}$$

This has a clear effect that biases robot motion so that the tool predominantly rotates about the PseduoRCM point P_r, especially when a force is applied perpendicular to the handle. This is shown in Figure 4.12B. This stems from the fact that the handle force is interpreted as a relatively small force and a large moment at the PseudoRCM point. For comparison with conventional cooperative control, the same input would generate only a translation in the direction of the force as shown in Figure 4.12A. The $\begin{bmatrix} K_f \\ \\ K_\tau \end{bmatrix} \in \Re(6 \times 6)$ is the admittance gain with first three elements of the diagonal equal to the scalar K_f which is applied to the force components, and last three are equal to K_τ and are applied to the torque components. By increasing K_τ and decreasing K_f, the translations are further minimized. With a high $\frac{K_\tau}{K_f}$ ratio the system response approaches that of the virtual RCM presented in Section 4.4. In order to cause the point P_r on tool to translate, the surgeon needs to apply a torque at P_h that will produce a large translation at P_r. The relative gains ratio is adjusted to provide the most intuitive and effective response (typically ~ 2), and is surgeon and task specific. The side-effect of this control is that the tool tip motion is predominantly from the rotational axes and any translations in the cardinal directions require significantly more force. Although this asymmetry may initially be considered counter-intuitive, it does stabilize tool tip motion in the direction of the retina. However, for most surgeons it is beneficial to reduce this

[11] A slightly different PseudoRCM Behavior involves standard cooperative control that uses the f applied at the handle and the jacobian computed at PseudoRCM point (J_r). The objective function is simply: $argmin||J_r(q) \cdot \Delta q - Kf||$

asymmetry between the rotation and insertion response of the robot by increasing the gains on the force along the $\hat{Z}_t$ axis. This is accomplished by increasing the $\kappa > 1$ in $K_z \in \Re(6 \times 6)$ with the diagonal $= [1, 1, \kappa, 1, 1, 1]$, which is then applied to f in the handle coordinate frame.

Through experimentation the same approach has been applied to improve the stability of translating the tip of the tool as shown in Figure 4.12. This especially useful in cases where specific rotation of the tool is not extremely significant, and can improve the precision of the control of the tip of the tool.

4.6 Bimanual Virtual RCM Behavior

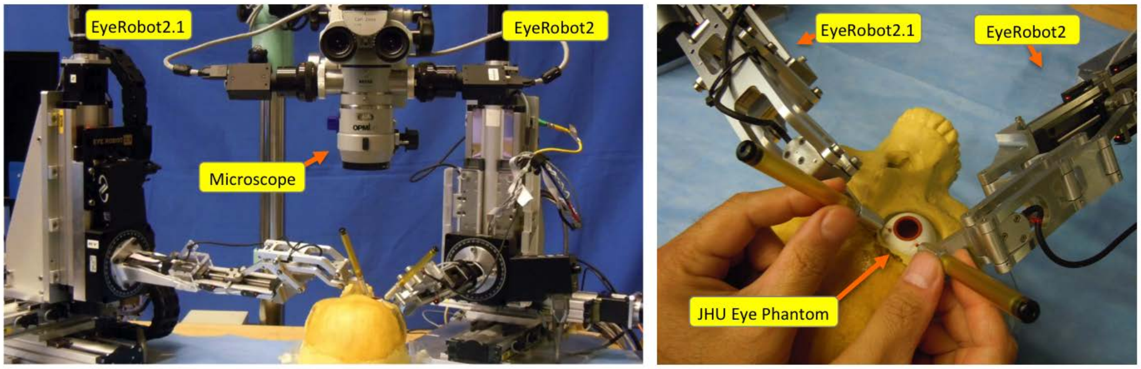


Figure 4.13: Bimanual setup of the EyeRobots.

In the case of two or more EyeRobots assisting the surgeon (e.g., the bimanual setup in Figure 4.13) additional safety constraints may be required. Since EyeRobot stiffness eliminates humanly-perceived level of forces between the tool shaft and sclera, a scenario where two robots are inadvertently guided to rotate the eye in opposite directions may generate excessive forces on the sclera and cause serious complications. One way to address this is to implement a force feedback control algorithm that incorporates a specialized force sensor sensitive only to sclera related forces, as described by He and Balicki et al. in [124]. Another way is to provide a simple fixed RCM constraint for both robots independently defined by the surgeon so as not to generate any stress on the retina, although this approach may be impractical, requiring too much time. Another way is to implement a smart virtual fixture that coordinates and constrains the relative motion of the two robots so that the instruments are at a constant distance apart when measured between the two sclerotomy locations (i.e. trocars), and ensure that these locations lie on the surface of the eye, as shown in Figure 4.14. In practice, this constraint can be used in a special robot operating mode used to translate the view by translating the RCMs along the surface of the eye, which inherently redefines the RCM locations. Or it can be enabled continuously to provide system interaction similar to freehand surgery. In an ideal implementation, an external tracking system would continuously provide the relative location of the eye surface, trocar locations, and robot tool positions which would be incorporated into the control methods for minimal user input, and provide additional information to improve safety.

The following describes the implementation of such a virtual fixture using the constrained optimization framework. The main requirement for this constraint is to ensure that any motion of the two instruments inside the eye results in the two trocars (sclerotomy locations) translating along the surface of the eye (i.e. rotating the eye). Additionally, tools can be manipulated in any way as long as the trocar locations are separated by this constant distance, plus a tolerance. The assumption is that the two robots share a common coordinate system, the location of the eye surface is known and approximated as a sphere, and the initial RCM locations are defined with correct separation spacing and are on the surface of the eye. Also, the separation should be significantly less than the diameter of the eye, e.g., 15 mm for trocars in a human eye. Each robot's controller is

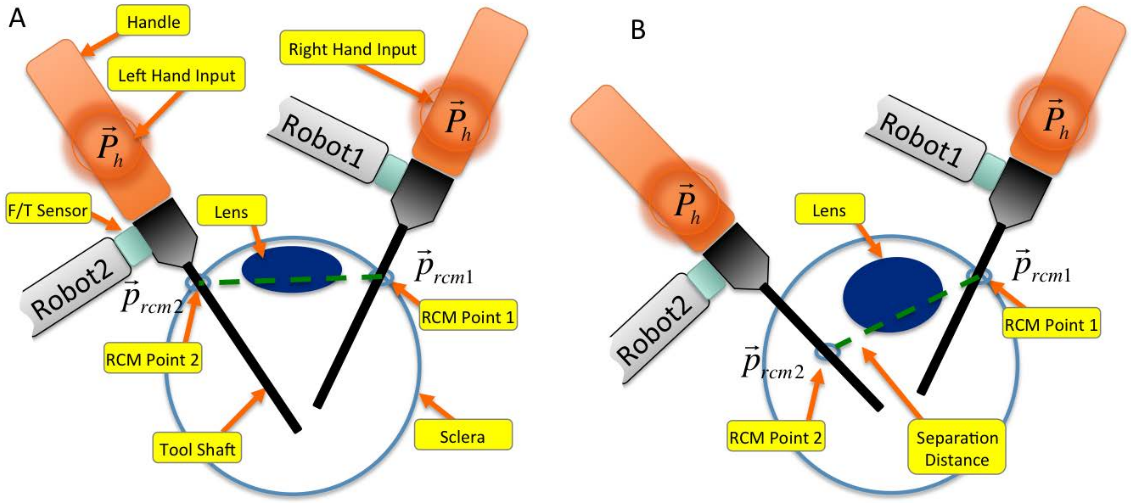


Figure 4.14: Coordinated translation of the eye orbit and associated RCM points using two robots. The distance between RCMs is constant before A) and during B) the motion.

operating independently and the location of the other robot's RCM definition is considered static for the time of the single loop iteration. The objective is the standard cooperative control described earlier in section 4.3.4. For demonstration purposes, the following presents the case from the point of view of Robot2 controller where the Robot1 RCM location is fixed.

For each iteration of the optimization, the eye surface is linearized by assuming that the surface of the eye is a tangential plane (S) at p_s, the intersection of the tool and the eye surface itself. The estimate of the ideal location of p_{rcm2} is done by constructing a vector d from p_{rcm1} (received from Robot1) in the direction of p_s. The magnitude of d represents the desired neutral distance between the trocars, while arcs of radius D_{min} and D_{max} about p_{rcm1} represent the tolerance bounds for the resulting incremental motion of the tool from p_s in the plane (s), i.e. length of vector $e = p_s + \Delta X_s - p_{rcm1}$ should be $D_{min} \leq ||e|| \leq D_{max}$. The incremental motion (ΔX_s) is expressed as joint motion using the manipulator Jacobian ($J_s(q)$) and the following constraint is

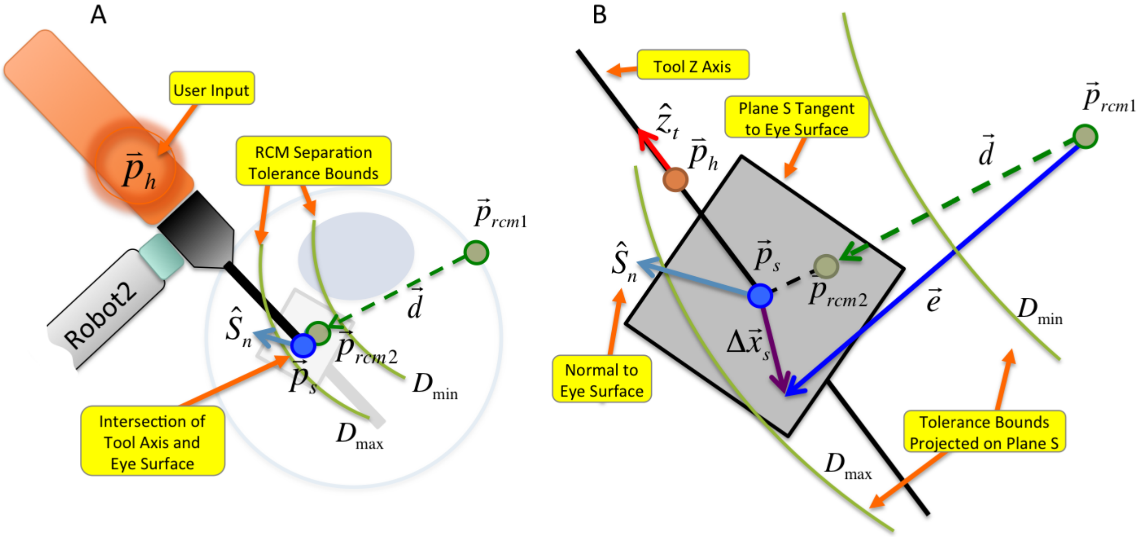


Figure 4.15: A) Virtual fixture geometry for safe rotation of the eye in a bimanual robot setup. B) Detailed schematic from Robot2's perspective, showing the plane approximating the eye surface at the tool-surface intersection, the RCM separation tolerance bounds, and the desired robot motion.

then incorporated into the control optimization:

$$D_{min} \leq ||p_s + J_s(q) \cdot \Delta q - p_{rcm1}|| \leq D_{max} \tag{4.10}$$

The tolerance can be asymmetrical ($||d|| - D_{min} \geq D_{max} - ||d||$) to favor the tools approaching each other, allowing maximum tool configurations while limiting excessive tensile stresses on the sclera. To ensure consistency in a dual robot setup where each robot is running asynchronously, at each control loop iteration, the other robot's latest available trocar location (p_{rcm}) is updated. It is defined as the last known position of the remote robot's tool intersection with the eye surface (p_s). If this position is updated more or less in real time, the global bimanual constraint will be enforced. To provide more elastic behavior similar to what the surgeons experience in surgery without robot assistance, the above constraint can be extended to include slack variables [123] to generate soft constraints.

The quadratic constraint in (4.10) can be directly incorporated into a non-linear optimiza-

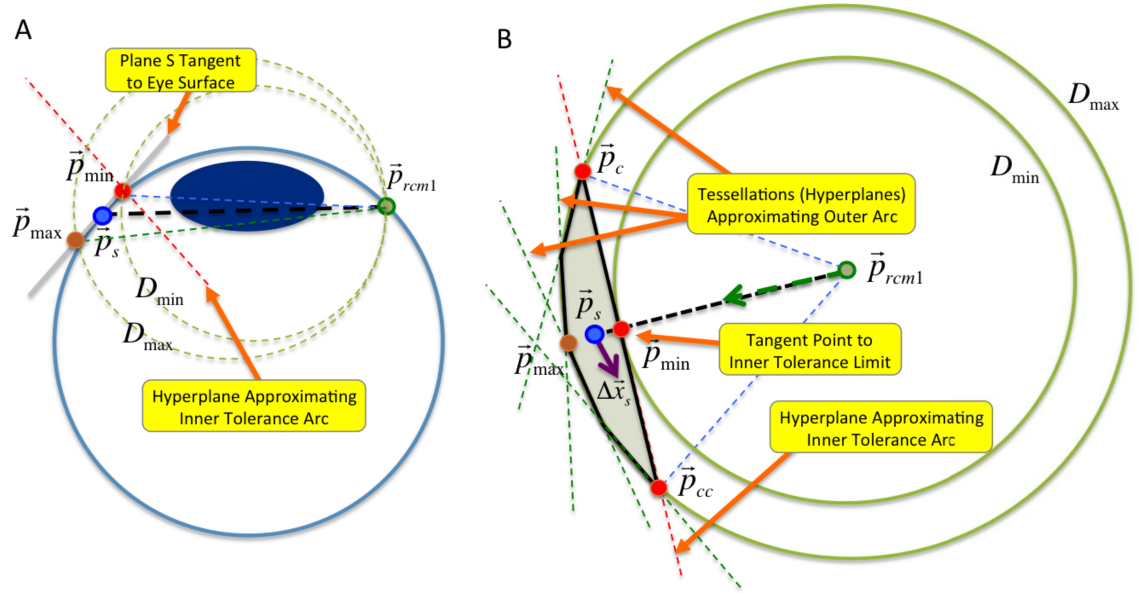

A
Plane S Tangent to Eye Surface
$\vec{p}_{\min}$
$\vec{p}_{rcm1}$
$\vec{p}_{\max}$
$\vec{p}_s$
$D_{\min}$
$D_{\max}$
Hyperplane Approximating Inner Tolerance Arc
B
$D_{\max}$
$D_{\min}$
$\vec{p}_c$
Tessellations (Hyperplanes) Approximating Outer Arc
$\vec{p}_{rcm1}$
$\vec{p}_s$
$\vec{p}_{\max}$
$\vec{p}_{\min}$
$\Delta\vec{x}_s$
Tangent Point to Inner Tolerance Limit
Hyperplane Approximating Inner Tolerance Arc
$\vec{p}_{cc}$

Figure 4.16: Convex linearization of the RCM separation virtual fixture with tolerances for bimanual robot setup. A) Lateral view of geometric construction of the bounding points. B) View normal to the plane tangential to the sclera at tool-sclera intersection. The shaded region depicts the approximated allowable tool (specifically tool-sclera intersection point) travel on the surface of the eye within one iteration of the robot control loop.

tion solver, or converted into a linear constraint as a linearized optimization problem. Figure 4.16 demonstrates a geometric approach to constructing hyperplanes (tesselations) that approximate the allowable region for incremental motion of the tool axis on the approximated surface of the sclera. This linearized set of constraints is recalculated at each iteration of the high level constraint optimization control loop. To ensure convexity, the inner tolerance (D_{min}) circle is approximated by a single hyperplane while the outer tolerance (D_{max}) is constructed with multiple hyperplanes approximated by an arc in plane S formed by points p_{cc}, p_c and p_{max}. As the number of the hyperplanes increases, the polygon approaches the curvature of the outer arc. As a rule of thumb, to ensure sufficiently smooth approximation in micro-surgical applications 20–50 tessellations/mm are chosen for a particular geometric constraint. Since the original inner tolerance boundary creates a non-convex region, the single plane is used to generate a locally convex approximation. This tends to create a very slight discontinuity if the tool is traveling along the inner tolerance boundary as the plane is redefined at each control step. The effects (e.g., micro-vibrations) of this approximation are minimal and decrease significantly with higher update rates (e.g., 200 Hz). The smoothness of the motion can be further improved by incorporating soft constraints. The response of this coupled system with variable stiffness inputs from the surgeon may conceivably become unstable. However, in practice typical cooperative control has been reliably stable due to the fact that the force control gains are low, and that the EyeRobots are inherently very damped and tuned to have low accelerations and decelerations.

4.7 Automatic Tool Retraction Behavior

For surgical applications, quick tool changes are desirable and often required for timing and safety reasons. Specifically, unintended patient movement requires an immediate compensatory

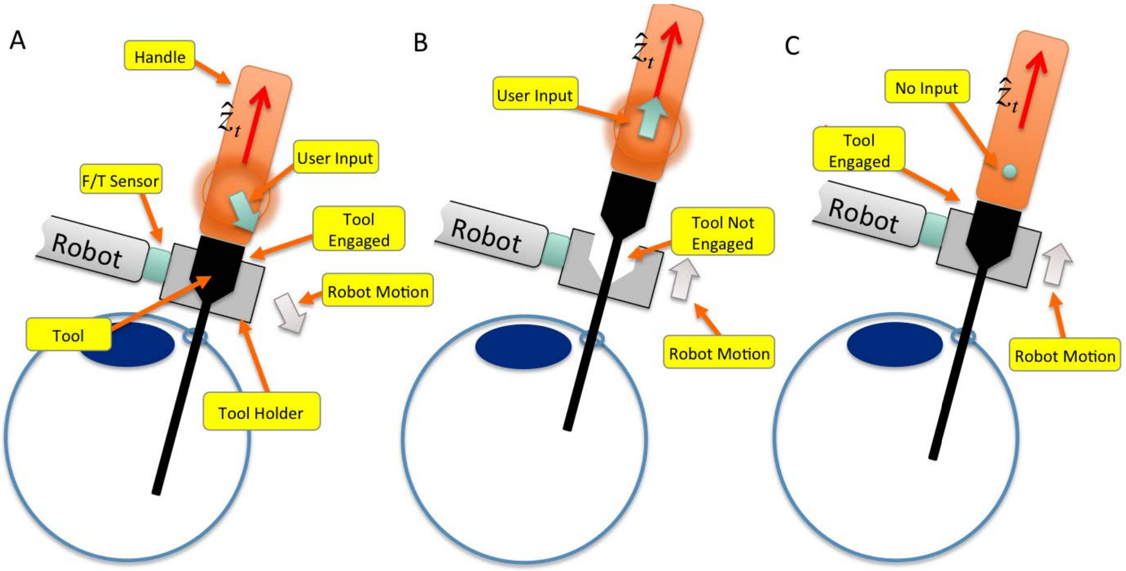


Figure 4.17: Schematic for Automatic Tool Retraction Behavior. A) Standard cooperative control with the operator commanding the robot with some force in the $-\hat{Z}_t$ direction. The tool is engaged in the tool holder. B) The tool is lifted out of the tool holder, the robot attempts to reengage by moving along the $\hat{Z}_t$ direction (retraction). C) There is no input so the robot retracts along the $\hat{\hat{Z}}_t$ direction.

repositioning of the instrument to avoid tissue injury. One implementation mentioned above is a quick release mechanism that allows the surgeon to eject the tool from the eye and separate it from the robot tool holder. An alternative method called Automatic Tool Retraction (ATR) is presented here. It is used for rapid engagement and disengagement of a surgical instrument (off-the shelf, modified, or custom) with/from the robot tool holder without compromising the benefits of cooperative control methods such as precision, stability, and virtual fixtures. The major technical challenge is to provide a solid connection between the robot and the instrument in enough degrees of freedom to be able to control the motion of the robot in a cooperative control manner. At the same time this connection should facilitate unimpeded and rapid removal of the instrument (i.e. minimal breakaway force) from the surgical field by the surgeon if the need should arise.

One version of ATR relies on a tool interface with a flange or a conical taper mating area that limits the depth of insertion of the tool into the tool holder, as shown in Figure 4.17A. The tool is simply inserted and removed by hand by guiding it axially into the tool holder. There exist a variety of means that may be used to detect when the tool is fully inserted into the tool holder (engaged), such as proximity sensors or direct contact sensors, etc. However, a more minimal approach is to use the existing force/torque (F/T) sensor to sense when the tool is engaging with the tool holder along the tool's z direction[12] ($\hat{Z}_t$). The tool may be considered disengaged if force ($f_{\hat{Z}_h}$) in the direction $\hat{Z}_t$ is lower than some threshold force f_{Zmin} (negative for pushing down). Once the tool is in contact with the tool holder, the handle input forces/torques (f) are transmitted

[12] Tool and handle axis are aligned: $\hat{Z}_t = \hat{Z}_h$.

from the F/T sensor and used by the admittance control law in the standard way:

$$v = \begin{cases} Kf & \text{if } f_{\hat{Z_h}} \geq f_{Zmin} \\ Kf & \text{if } ||f|| \geq f_{min} \\ Kf', where f'_{\hat{Z_t}} = R(d,t) & \text{otherwise} \end{cases} \tag{4.11}$$

One implementation of the $R(d,t)$ function redefines the force input component along the tool axis which results in a constant velocity for a short period t after which the motion stops. Another one incorporates a distance limit d, where the robot will retract a predefined distance along $(\hat{Z_t})$. Both of these result in the robot actively moving along the tool axis to sustain the connection with the tool, i.e. if the user applies a force along $(\hat{Z_t})$ as if the tool is to be removed, the robot moves along that $(\hat{Z_t})$ axis in the same direction, effectively following the desired motion of the operator (see. Figure 4.17B). Alternatively, $R(d,t)$ can result in the robot moving to maintain a minimum desired tool-to-holder contact force by $f'_{\hat{Z_t}} = f_{\hat{Z_h}} + F_{bias}$. This requires the surgeon to overcome the F_{bias} force otherwise the robot will drive the tool slowly away from the eye. By using high gains, the robot can continuously stay engaged with the tool, unless there is a complete and rapid removal of the tool, in which case the robot will move along the tool axis away from the eye for t seconds, or up to a distance d. These concepts can also be applied to a special case when the magnitude of the handle inputs $(||f||)$ falls below a desired threshold, i.e. there is no handle contact, the robot automatically retracts the tool away from the high-risk area, see Figure 4.17C.

In the initial implementation of this algorithm using EyeRobot and an instrument that is rigidly fixed to the tool holder, all the versions of $R(d,t)$ are able to retract the instrument directly out of the eye. It was feasible to perform very precise manipulations while still maintaining a bias force on the tool. Although this behavior solves the problem of tool retraction from the eye, it requires significant training and trust from the surgeon due to the fact that the robot may be in motion when the surgeon is not directly guiding it, which may seem to be out of control.

4.8 Virtual Joystick Behavior

Currently, the surgeons and the operating room staff are very comfortable interfacing with the surgical operating equipment through pedals and direct physical knobs, touch panels, etc. However, with the inevitable addition of new devices competing for user interface time and space, multipurpose interfaces are going to become more relevant. The force/torque sensor on the EyeRobot is predominantly used for controlling the motion of the surgical instrument but it can also be repurposed on the fly to provide 3D input for other system behaviors[13].

The deflection of the actual tool from user input when the robot is not moving is minimal due to a very stiff structural design and non-backdrivable nature of the actuators. The force/torque information can be interpreted as a 6DOF rate control joystick, much like the commercial 3D SpaceNavigator (3Dconnexion, Inc.) shown in Figure 4.18A. The current EyeRobot architecture only allows 5DOF input due to the non-encoded free rotation about the tool axis (see Figure 4.18B). For intuitive operation, the robot frame (F_R) with respect to which the force/torque measurements are generated is aligned with the surgeon's body frame (F_S), see Figure 4.18C). In typical robot-surgeon arrangement this has the effect of rotating the handle forces in the robot frame by ±30° about the robot Z axis. Note that the orientation of the handle relative to the robot base does not affect consistency of the input, i.e. if the handle is rolled to 20° position and the surgeon presses forward away from his body, the joystick behavior will produce an output that is the same as if the handle was rolled to -15° position.

[13] The initial idea for using the ER2 instrument as an input came from Anton Deguet.

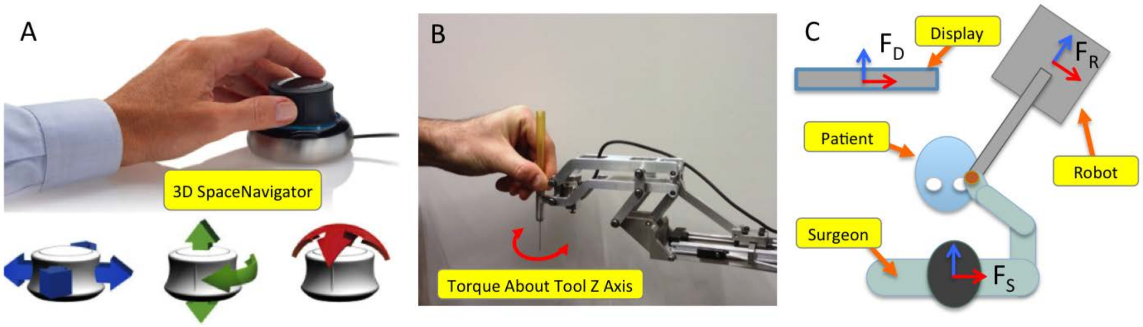


Figure 4.18: A) Commercial rate control 6 DOF joystick. (Image Credit: www.3dconnexion.com). B) Due to free-spin of the tool, the force/torque sensor does not measure the torque about the tool Z axis. C) Typical frame orientations in a surgical setting.

The output rate of the virtual joystick $v \in \Re(6)$ (velocity screw) is calculated as follows:

$$v = C\, Ad_S\, f \tag{4.12}$$

where the measured handle force and torques $f \in \Re(6)$ in robot frame are transformed by an adjoint transformation matrix ${}^{R}Ad_S \in \Re(6 \times 6)$ and multiplied by a diagonal matrix $C \in \Re(6 \times 6)$, the diagonal of which represents the factors to convert force (N) and torques (N/mm) into mm/s and rad/s.

In some instances it is more applicable to use the virtual joystick as a positioning device. This is accomplished by continuously integrating the velocity to generate the current position $p \in \Re(6)$ (twist) at time T :

$$p = \int_0^T C\, {}^{R}Ad_S\, f\, dt \tag{4.13}$$

where dt in this case becomes the sampling period of the force/torque measurements.

The virtual joystick capability provides a way for the surgeon to input cartesian coordinates into the system. It is used in the Telestration Behavior described in visualization Chapter 7 as a cursor input to virtually draw lines on the surface of the retina. The surgeon uses the virtual joystick to translate the cursor on the display over a region of interest. Then he or she pushes the handle in the downward direction which initiates a button down event, enabling the surgeon to draw a line by pressing on the handle in the desired directions. When the button is pressed again the drawing mode is disabled. Another possible application is to dynamically adjust working parameters of the microscope or the visual display such as the zoom, brightness, contrast, microscope position, etc. Using the virtual joystick to click or select a particular item by rotating through a list requires visual or aural feedback to represent discrete virtual joystick steps. Note that the XYZ joystick units have to be converted from mm to pixels or mm/s to pixels/s, and depending on the surgeon's preference, the joystick frame may need to be rotated about the horizontal axis (90°) to align with the viewing plane. The click events are implemented by monitoring out-of-plane motions with predefined values for button down, button up, etc.

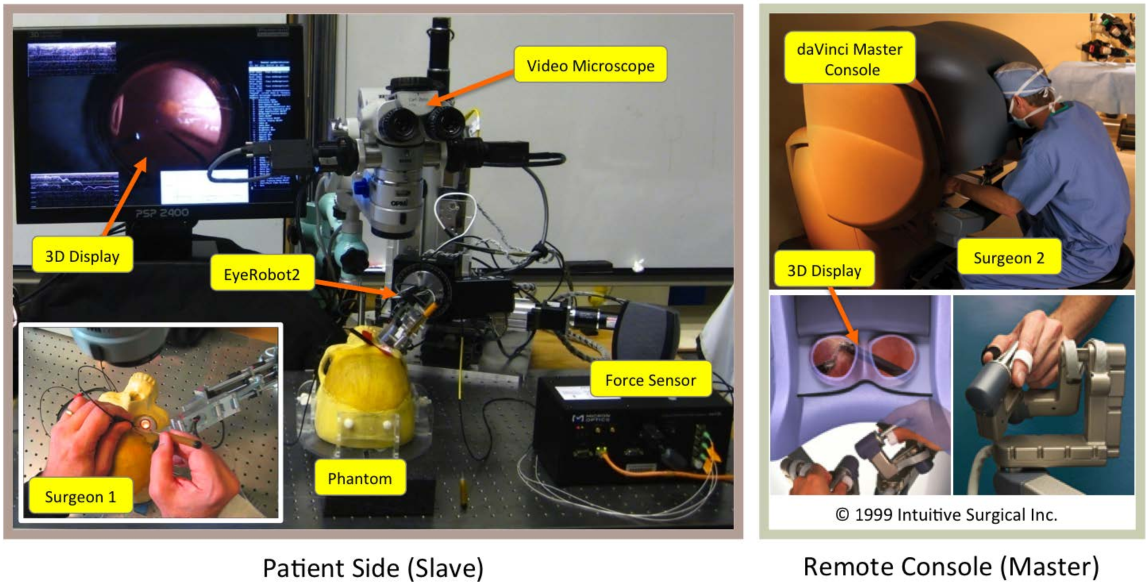
In some scenarios where the robot is not involved in a critical surgical task, such as in the preparation phase of a procedure, the joystick behavior can be combined with robot motion to provide haptic feedback, e.g., it will vibrate to acknowledge a selection, or a click. To further enhance intuitive communication, the haptic feedback should be combined with complementary aural or visual feedback.

Switching into the joystick behavior can be done by pressing a foot pedal, a voice command, or via a GUI button on the nurse console. Another option is to insert a special joystick tool into the tool adapter. This tool does not have a standard length tool shaft to prevent accidental collisions with the eye. When its insertion is detected via a contact switch, for example, the joystick behavior is enabled. Furthermore, this tool could constrain the rotation about the tool axis so the torques can be detected by the force/torque sensor to provide full 6 DOF input.

4.9 Cooperative Teleoperation Behavior

The cooperative control paradigm's main advantages are that the operator interaction with the surgical instruments is familiar and direct but much steadier than freehand operation. Also, since the surgeon is holding the tool, it can be removed from the eye without delay at any moment. In the teleoperation architecture the surgeon controls the robotic manipulator (slave) from a remote master console. This system has similar advantages of minimizing hand tremor, but can provide an even finer degree of tool control by employing a motion scaling scheme. There are a few disadvantages to teleoperation, including difficulty in performing safe gross motion inside and outside of the eye due to lack of direct visualization, significant reliance on correct definition of the location of the RCM, and the increased slave design complexity to comply with stringent safety and functional requirements such as an emergency tool ejection mechanism and patient-robot position tracking and fixation. By incorporating these two paradigms into a single hybrid tele-robotic and cooperative system their respective advantages are combined and weaknesses are supplemented. Figure 4.19 shows a prototype of such a system that incorporates the EyeRobot2, daVinci Master console, and tele-video stereo-microscopy. The system architecture is fully described by Balicki et al. in [125].

In such a system configuration, the instrument held by the EyeRobot can be controlled by the remotely located surgeon through teleoperation, as in the case of the master-salve daVinci Surgical System or it can be controlled locally by one of the surgeons directly manipulating it through steady-hand cooperative control (CC). The source of control over the instruments can be easily and quickly exchanged during surgery, which is especially applicable to teaching and mentoring where the trainee is able to hand over control of the tools to the expert at any time. The expert can fully manipulate the surgical instrument, for demonstration, or to execute a very



© 1999 Intuitive Surgical Inc.

Patient Side (Slave)

Remote Console (Master)

Figure 4.19: Hybrid Teleoperation and Cooperative Control system prototype.

difficult task where motion scaling through teleoperation is highly applicable. Also, specialists can be assigned to perform a specific task from a remote location, which is often a short but critical part of the surgery. One can envision such a scenario where this specialist assists in multiple concurrent surgeries. The daVinci SI Surgical System (Intuitive Surgical, Inc.) offers a similar feature by providing two identical master consoles (two masters controlling multiple slave manipulators in standard teleoperation), and can also be used for collaboration where two surgeons control different tools simultaneously in the same workspace. E.g., while one surgeon performs the primary tasks of the operation, the second surgeon can assist with another task, such as retraction. Another control paradigm involves sharing the control of the tool between operators. Some examples include work by Nudehi et al. [126], designed specifically for surgical training where two users operating a master each can share the control of a slave manipulator, and the effects of their relative contributions can be adjusted. I.e. each master interface provides feedback forces to the surgeons, proportional to the difference of their actions and reversely proportional to the control authority shared between them. The control authority is chosen based on their relative level of surgical skills and experience and determines the extent to which the motion of the surgical robot depends on their individual commands. Khademian et al. [127] extended this concept to include feedback from the interaction between the users and also the environment, where the position and force of two masters and slaves are exchanged.

The main distinguishing aspect of the proposed hybrid system is that the slave manipulator serves also as a pseudo-master manipulator for one of the surgeons. Additionally, the control methods are implemented with the constraint framework described in Section 4.3.4 and can incorporate virtual fixtures on the EyeRobot manipulator that are inherently rendered on the master side. The following control variations are implemented with the addition of the teleoperation manager responsible for exchanging robot state data, rectifying master-slave coordinate orientation alignment, and controlling teleoperation gains and modes. The communication delay between the teleoperation manager and the manipulator controllers is considered to be negligible. Swapping between the various control modes can be accomplished via a foot pedal, and is typically assigned to the more experienced of the two surgeons. However, the surgeon physically present with the patient has the ultimate control over the extraction of the instrument from the eye.

In **Classic unilateral teleoperation (UTO)** the EyeRobot is used as a simple slave manipulator, and is controlled remotely by a single surgeon. In this control mode the slave tracks the master with a simple position servo, and the master does not include force or position feedback from the slave. The surgeon on the master side is required to close the control loop solely via the visualization system. To diminish the effects of inherent human hand tremor and to enable more precise maneuvers required in micro-surgery, position motion scaling (MS) is implemented.

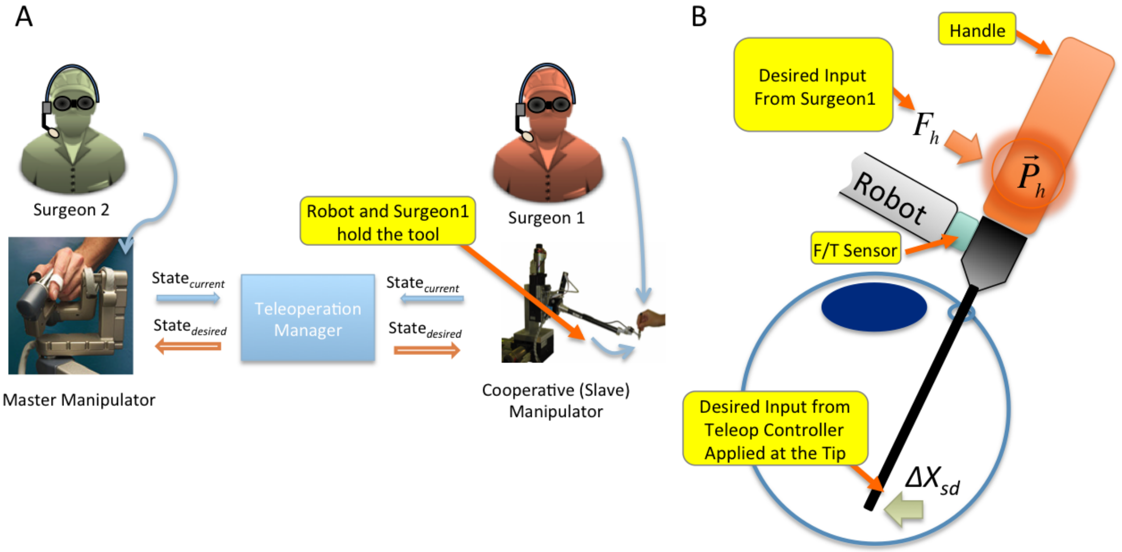


Figure 4.20: A) Schematic of the teleoperation system setup. B) Control points for the master and slave on the tool inside the eye.

It involves scaling the Cartesian position of the master by a factor ($K_m s$) relative to the Cartesian position of the slave. For highly scaled manipulation, the travel limits of the master often are insufficient and as a result a clutching scheme is implemented where the surgeon “clutches-out” (disengages from controlling the slave, typically with a pedal press), moves the master manipulator to the center of its workspace, and then “clutches-in” to regain control of the slave. This effectively extends the workspace of the master manipulator. Furthermore, to make the teleoperation appear natural, the slave position and orientation should be measured relative to the microscope, while the master position and orientation should be measured relative to the surgical view[14]. Equation (4.14) shows a typical method to generate the desired Cartesian positions on the slave (X_{sd}) where R_{ms} is the master-slave alignment rotation matrix, X_m is the master position, and X_{sref} is the

[14] Depending on surgeon preference the master’s coordinate frame’s X-Y plane can be aligned with the table or with the 3D display. i.e. by moving the master up, the EyeRobot could generate motion up away from the patient or in the plane of the retina. For expert surgeons the former is preferred.

reference position in the slave coordinate frame generated by clutching. In practice, K_{ms} is a scalar in the range 0.01-1.0, and can be dynamically changed through a GUI or a variable input pedal.

$$X_{sd} = K_{ms} R_{ms} X_m + X_{sref} \tag{4.14}$$

This can easily be extended to include rotations. However, special consideration is required when implementing motion-scaling and clutching which only works with rectilinear motion, and does not feel intuitive with orientation scaling or clutching. The EyeRobot2-side controller incorporates the X_{sd} in an error term of the following objective function where X_s is the current position of the tip of the tool and the $J_t(q)$ is the corresponding Jacobian:

$$\underset{\Delta q}{\arg\min} \quad ||J_t(q) \cdot \Delta q - K(X_{sd} - X_s)|| \tag{4.15}$$

The proportional gain K adjusts how responsive the slave is to the input from the master[15]. Since the majority of retinal surgeries only visualize the distal tool shaft and since the typical tool tip motion inside the eye is translational 3DOF, the master to control the slave's tool tip (XYZ translation) is very intuitive. This lends itself nicely to incorporation of virtual fixtures by simply adding constraints to the optimization formulation. UTO performs sufficiently in simple tasks but it lacks fidelity in more complex maneuvers or when visualization is poor due to lack of necessary feedback. Additionally, master side tracking errors are inevitable in a situation where virtual fixtures restrict the motion of the slave, since the actual motion of the slave is not communicated to the master.

In **bilateral teleoperation (BTO)** control, the behavior of the slave is fed back into the master controller, and vice versa[16]. If the slave motion is impeded by the slave-side operator through cooperative control, or a slave-side virtual fixture, the motion of the master manipulator will also be

[15] An additional damping term can be incorporated for finer tuning

[16] Other permutations of teleoperation control are possible (e.g., feedback to the master but no feedback from the master to the slave)

impeded, pragmatically creating a sense of force-feedback for both operators.[17] Since the EyeRobot is non-backdrivable the environmental effects are considered insignificant. There are many bilateral teleoperation algorithms [117,118,128] that can be implemented in this case depending on the type of performance requirements (e.g., transparency, network delays) and robot types (e.g., impedance vs. admittance). The most common method, the position exchange controller shown in Equation (4.16), is implemented as a proof of concept with the slave being an admittance type device taking position as an input (X_{sd}) and the master using an impedance type controller expecting a force as an input (F_{md}).

$$X_{sd} = K_m s R_{ms} X_m + X_s ref \tag{4.16}$$

$$F_{md} = K_p(X_m - K_{ms}^{-1} R_a^{-1}(X_s - X_{sref})) - K_d(\dot{X}_m - K_{ms}^{-1} R_{ms}^{-1} \dot{X}_s) \tag{4.17}$$

In (4.17) above, the $K_{ms}^{-1} R_{ms}^{-1}(X_s - X_{sref})$ converts the slave's position back into master's workspace considering the scaling factor and clutching positions. The $\dot{X}_m$ and $\dot{X}_{sd}$ are the velocities of the master and slave respectively, and the K_p and K_d are the proportional and derivative gains used to tune the response of the master. The challenge arises in how to incorporate X_{sd} into the control of the EyeRobot so that the motion of the tool is a combination of both of the surgeons' inputs. One approach is to treat the error term $(X_{sd} - X_s) \in \Re 3$, where X_s is the position of the tool tip, as a virtual force. This virtual force is then combined with the force applied by the local surgeon at the handle. In admittance control these forces are considered velocities and one is transformed from the tip to the handle as shown in Figure 4.20 where $\Delta X_{sd}^* = Ad_h^t(X_{sd} - X_s)$ with $\Delta X_{sd}^* \in \Re 6$.

$$\underset{\Delta q}{\arg\min} \quad ||J_h(q) \cdot \Delta q - (K_h F_h - K_t \Delta X_{sd}^*)|| \tag{4.18}$$

[17] This control scheme requires the master manipulator to poses actuation of the require DOFs to provide the sufficient feedback.

The relative contributions to resulting motion can be adjusted by altering the ratio of K_h to K_t. Furthermore, to ensure that one of the inputs can dominate, a maximum limit on one of the inputs can be implemented. In general use this works quite well, but it can fail where virtual fixtures create fulcrums reversing the direction of the desired velocity stemming from the Ad transformation.

Another approach is to generate two objectives in different task frames that are linked via the weighting matrix W as shown in Equation (4.19). The weighting matrices $W_h \in \Re(6 \times 6)$ and $W_t \in \Re(3 \times 3)$ are diagonal and contain the relative importance w_h and w_t of the task objectives. In the case where the two objectives are competing, the ratio between w_h and w_t effectively lowers the likelihood that the lower ranked objective is minimized (i.e. achieved). In practice the range for these weights is between $[0, 10^4]$.

$$\arg\min_{\Delta q} \left\| \begin{bmatrix} W_h & 0 \\ 0 & W_t \end{bmatrix} \begin{bmatrix} J_h(q) \\ J_t(q) \end{bmatrix} \Delta q - \begin{bmatrix} K_h F_h \\ K_s(X_{sd} - X_s) \end{bmatrix} \right\| \tag{4.19}$$

This teleoperation behavior was demonstrated with the daVinci classic console as well as the Phantom Omni (SensAble Technologies Inc). The end-effector tip position of the master is used to control EyeRobot2's tool tip while inside the eye. This provides sufficient and intuitive control of the tool tip when the RCM constraint is enabled. The clutching is achieved with a pedal or a button press, in the case of the Phantom Omni. The microscope was outfitted with an additional camera system (described in the visualization chapter) providing an additional view (picture-in-picture) of the patient's head in the master console. This is especially important in synchronizing surgeon interactions and observing the tool motions outside of the eye.

Just as with Equation (4.15) the above optimization formulation can include virtual fixtures that are dynamically updated based on real-time sensing. The following chapter shows how a real-time force information from sensors embedded in surgical instruments can be incorporated in a cooperative force control law (see Section 5.6) to enhance surgical force perception. Since the BTO involves position exchange feedback, the operator on the master side will experience a similar force, or any effects from dynamic virtual fixtures implemented on the slave manipulator. A paradigm that could improve the haptic fidelity on the master side console is implementing VF (with real-time information sensed at the slave) in the master controller, as shown by Abbott et al. [118]. Indirectly, the slave side control would be affected by the VF through position exchange, however, since the slave side operator would also contribute to the overall control, implementing VF on both is probably the ideal solution.

It is important to note that the EyeRobot2 was designed for pure cooperative control and has only 5 DOF of control (no actuated tool rotation axis or the ability to actuate (e.g., open/close) instruments), so even though the teleoperation algorithm can command full 6 DOF Cartesian position and orientation, the desired rotation about the tool axis is omitted by the ER2 control optimizer. As is, it can be used with axisymmetric instruments such as an illumination guide, and, in some cases, a surgical pick; for use with more sophisticated tools it will be upgraded with a new tool actuation mechanism that will allow for teleoperation of all available degrees of freedom of the surgical instrument. Besides requiring additional hardware, the main disadvantage of the Cooperative Teleoperation Behavior is that two surgeons need to be present during the operation. One embodiment that could address this is having a specialist surgeon assist on many cases through a single master console. This surgeon would “enter” the surgery for a brief time to perform a highly specialized task, much like a team of specialists is involved in a single surgery to perform various steps - one surgeon performs the keratoplasty, another the lensectomy, etc.

4.10 Vein Cannulation Experiment

The typical treatments for Retinal Vein Occlusion (described in 1.5.2) do not address the source of the occlusion, namely, the blood clot that has become trapped in the small ($10 \sim 130$ μm) retinal blood vessel [17]. Freehand surgical procedures that involve puncturing the small vessel upstream of the clot and delivering a clot dissolving medicine have proven clinically difficult due to the small vessel size, fragility of the surrounding tissue, and difficulty in reaching the site of occlusion. A few groups have attempted to assist the surgeons with teleoperated robots [88, 94], a hand-held device like the Micron [109, 129], or magnet driven capsules that deposit medication inside the vessels [102]. Preliminary work done by Fleming and Balicki et al. [106] presented a study that compared cooperative robot-assisted vein cannulation and free hand vein cannulation on the vessels in a Chick Chorioallantoic Membrane (CAM). The results indicated that with some training, the robot improved the surgical performance of maintaining the cannulation, and after two trials, improved the ability to cannulate the blood vessel. However, the geometry of the experimental set up allowed the surgeon unrealistic adjustments to position his/her hand and pipette in a comfortable position. Due to the straight tip of the pipette the subjects naturally navigated towards a more parallel approach to the surface of the CAM, which neglects the typical constraints of vitreoretinal surgery, specifically the sclerotomy constraint. These experimental limitations are addressed in a follow up experiment described below[18].

To realistically simulate the eye constraints an artificial sclera was built into a human skull phantom and the chicken embryo was placed under it so that the CAM was ~25 mm below the surface of the eye, simulating the retina, see Figure 4.21, and refer to Section 3.3.7 for details. EyeRobot1 was used in fixed virtual RCM mode with the force control gain set to $K = 5$. The

[18] Thank you to CIS2 (Johns Hopkins University, 600.466) course students : Bryce Chiang, James Gao, Yinfei Xu for executing this experiment.

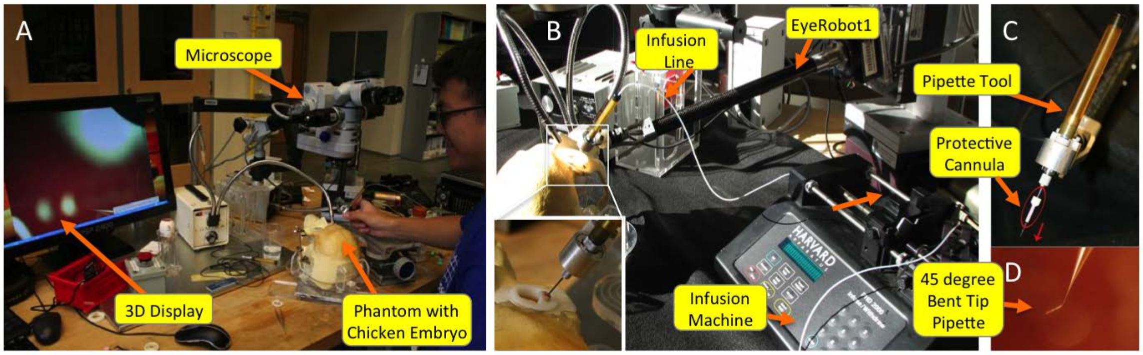


Figure 4.21: Vein cannulation setup: A) Experimental setup showing the subject operating with the EyeRobot1, using 3D display for feedback. B) Closeup of the skull and the sclerotomy site. C) Pipette tool integrated into the robot end effector and a protective cannula (trocar) for pipette insertion through the sclerotomy. D) Bent-tip (±45°) pipette.

previous trials were done with straight pipettes oriented nearly parallel to the vessel. To enable a similar vein approach while considering the constrained environment, a bent-tip pipette is used for safe and reliable entry into the vessel to minimize over-puncturing or damaging of the deeper retinal structures. A custom cannular trocar is used to introduce the pipette through the sclera insertion site. The lumen of the trocar is large enough to accommodate the bent tip portion of the pipette so it is ideal for protecting the tip during the insertion, see Figure 4.21C. After the insertion is done, the trocar provides enough flexibility so that the pipette can rotate freely about its axis and is protected from concentrated transverse stresses, greatly reducing the chance of breakage along the shaft of the pipette. This is especially critical in freehand operation where forces from the sclera interaction could exceed the breaking force for the pipette shaft.

The ideal injection maneuver is shown in Figure 4.22A. First, the pipette is aligned so that the tip is parallel to the vessel. Next, the pipette is brought down so that the entire tip is lying against the vessel. With the pipette aligned, the pipette is translated along the vessel with a slight downward force to pierce the vessel wall. The pipette must then be kept in the vessel for

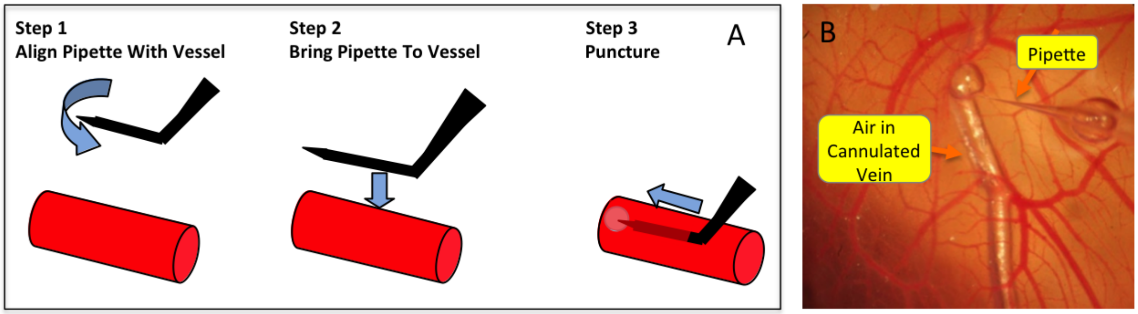


Figure 4.22: Vein cannulation schematic: A) Steps for ideal cannulation of a vessel with bent-tip pipette. B) Pipette injecting air into the cannulated vessel.

one minute simulating drug delivery. In order to facilitate the above maneuver the pipette requires a bend in its tip to account for the geometry of the eye, sclerotomy location, random target vessel arrangement, and a straight tool shaft. Depending on these factors and a location of the vein occlusion a particular pipette tip bend angle is required. For demonstration 45° bent-tip pipettes (MIC SI-45, ORIGIO Humagen Pipettes, Inc.) are chosen due to wide commercial availability. These are originally made for in-vitro fertilization, and have 10–11 μm bevel length, and 5–6 μm interlumenal diameter, and a spike at the tip. In previous trials, mineral oil with an infusion pump has been used to simulate the medication being injected into the vessels. However, the high viscosity and impurities caused numerous instances of clogging of the very small lumen of the pipettes, and because of its clarity, it was difficult to visualize, especially to determine the state of cannulation. Instead, an air-filled syringe has been used to manually inject air when the test subject indicates a successful cannulation, see Figure 4.22B. This proved to be much more effective in determining the state of the cannulation and did not produce “oil slicks” on the surface of the CAM that interfered with visualization.

A JHU IRB-approved experimental protocol involved 11 engineering undergraduate and graduate student volunteers performing freehand and robot-assisted CAM vessel cannulation. All were right handed and had no experience with manipulating micro-scale parts. Upon arrival, each subject was given an overview of the experiment, instructions on using the robot, as well as a brief explanation of the parallel cannulation technique described above. Subjects were allowed a few minutes to familiarize themselves with the setup and trained by tracing circles for both freehand and robot-assisted scenarios. The stereo video microscope adjustments as well as the insertions of the pipettes through the sclerotomy were handled by the experiment moderators.

Each subject completed ten 5-minute continuous cannulation trials, where the first five trials were robot-assisted, and the next five trials were performed in freehand fashion. This order was reversed for each consecutive subject to minimize bias due to learning effects. The subjects were offered rest breaks in between each trial. A 100 μm wire was used as a template to pick out a vessel (regardless of whether it was a vein or artery) to match target retinal vessel sizes. A "successful cannulation" was defined as maintaining the pipette within the vessel for 60 seconds, immediately after which the trial is completed. Trials were aborted for the following reasons: 1) broken pipette 2) robot malfunction 3) request of volunteer. Aborted trials were not included in the data analysis. The statistical and qualitative differences between the two methods were compared on the following metrics: the elapsed *time to guide to and align* with the vessel from the starting position; the elapsed *time to first cannulation* from the starting position; the elapsed *time of maintained cannulation*, i.e. the time pipette was inside the vessel; and *total time* for each cannulation trial. Other measured metrics included the *success rate* as the percentage of trials in which the subject successfully cannulated for 60 seconds; *number of cannulation attempts* is the number of times the pipette entered a vessel and injection was observed; *number of broken pipettes* by the subject during the trial; and *trauma level* corresponding to vessel damage (torn vein and attendant hemorrhaging) assessed visually. A feedback questionnaire was also collected to record subjective experience of the test subject. The data in Table 4.2 shows the absolute times for the steps in cannulation trials for all subjects. However, these results can be misleading due to variability in micro-manipulation skills of each subject as well as differences in task learning. To address this, each subject's performance was assessed by comparing that subject's robot-assisted vs. freehand trials, i.e. the values from the robot-assisted metrics were divided by those of freehand, as shown in Table 4.3. Tests for significance use null hypothesis: mean = 1, indicating no difference between robot-assisted or freehand performance. Two values were statistically significant with a $p \leq 0.05$: 1) longer time to first cannulation and 2) longer time of maintained cannulation, both for the robot-assisted scenario[19]. No statically significant differences were found for time to align, total time, and number of attempts per trial.

Metric	Robot-assisted (s)	Freehand (s)
Time to guide and align	35.02±19.74	49.52±53.30
Time to first cannulation	89.95±40.76	64.31±40.27
Longest maintained	54.80±7.58	42.12±18.89
Total Time	178.52±67.91	216.59±63.08
# of Attempts	1.27±0.45	1.57±0.60

Table 4.2: Results compare freehand and robot-assisted cannulation metrics where means and standard deviations are averaged across each subject's trials and then across all the subjects.

The time to first cannulation took significantly longer in robot-assisted trials by a factor of 1.72 with a significance level of 4% for the null hypothesis. This can be attributed to a low top speed of the robot compared to uninhibited freehand motion which directly translated into a longer length of time needed to guide the pipette to the site, align the bent tip with the vessel,

[19] Longer time of maintained cannulation averages include the trials that were capped at 60s.

Metric	Mean of $\frac{robot}{freehand}$ ratios	p-Value
Time to guide and align	1.22±0.83	0.20
Time to first cannulation	1.72±1.25	0.04
Longest maintained	1.26±0.37	0.02
Total Time	0.88±0.37	0.15
# of Attempts	0.91±0.46	0.26

Table 4.3: Comparison of Robot-assisted vs. freehand cannulation results for each subject. Individual subject's ratios are averaged across all subjects to find mean ratio (displayed) for the whole experiment. Results indicate how better or worse each subject performed in robot-assisted over freehand sections of their experiment session.

and cannulate. However, the average total length of the trials was shorter since there were more successful robot-assisted trials, which were terminated before the five minute limit, decreasing the average trial length. Without the five minute limit, unsuccessful trials might require significantly longer time and show larger difference in total trial length.

Figure 4.23 shows the longest times of maintaining cannulation for each subject in ascending order based on Freehand results. For each subject, with one marginal exception (subject #8 showed no difference), the robot-assistance yielded the longest cannulation times, regardless of which type of trials were done first. Robot-assisted cannulation maintenance times were longer by a factor of 1.26 with a significance level of 2% for the null hypothesis. This is directly related to the performance characteristics of the cooperative robot control, mainly stability, tremor elimination and fatigue alleviation. Once inserted, the pipette can be maintained in that position indefinitely without much effort from the test subject. On the other hand, freehand trials were much more fatiguing since the test subjects must actively keep the pipette still, fatiguing the muscles and eyes for as long as the cannulation needed to be maintained. Furthermore, looking away from the dis-

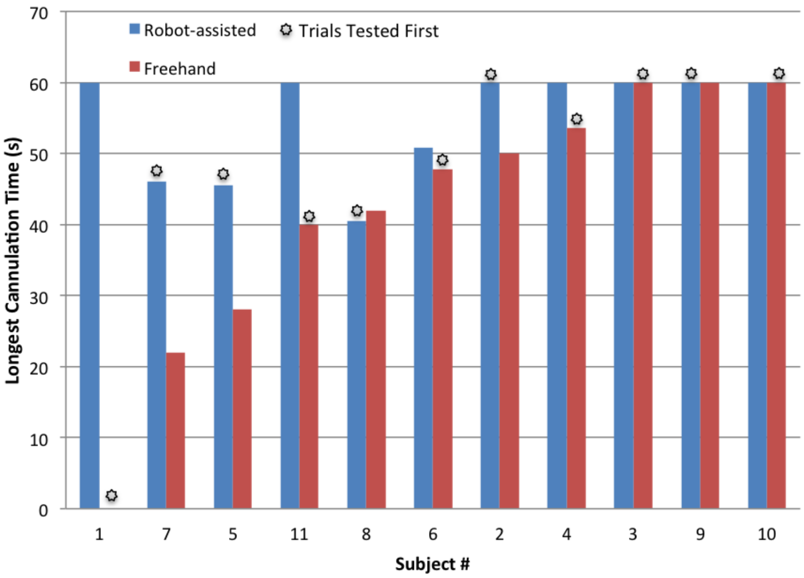


Figure 4.23: Results for longest cannulation time for each subject for robot-assisted and freehand trials in ascending order based on freehand values. Each column in the plot represents the best out of 5 trials. The trial experiment order is also indicated (i.e., which five trials were ran first for a subject: robot-assisted or freehand).

play greatly increases the risk of unintentional drifting since such micron-scale motion cannot be detected through human proprioception. Therefore, freehand trials required more attempts before successful cannulations were achieved. Every penetration and retraction of the pipette increases the risk for bleeding as well as incorrect placement. Subjects often penetrated the tissue surrounding the vessel and consequently causing fluid leaks out onto the CAM. Of course such mistakes are not acceptable in retinal vascular surgery. Though the difference in number of attempts is not statistically significant, it is believed that with more training and more test subjects, the performance will significantly favor robot assistance. Success rate is one of the more clinically relevant metrics that presents the share of trials in which the subject successfully maintained the cannulation of a vessel for 60 seconds. The results are in Table 4.4. The robot-assisted trials were much higher, at 70%, while freehand was only at 44%. The same table shows that that the mean number of pipettes broken is marginally higher for robot-assisted (1.75 pipettes per trial) than for freehand (1.50). During freehand, pipettes were broken when subjects angled the tool improperly, which forced the pipette against sclerotomy constraints, or became distracted and pulled the pipette out of the sclera, consequently breaking the bent tip. The pipette damages from robot-assisted trials were mainly due to robot malfunction, where the pipette was ejected from the end effector, or the

Metric	Robot-assisted	freehand
Success Rate	$70 \pm 29\%$	$44 \pm 38\%$
Pipettes Broken	1.75±0.96	1.50±0.58
Trauma	0.70±1.25	0.91±1.92

Table 4.4: Cannulation results for cannulation success, number of broken pipettes and trauma rates comparing robot-assisted and freehand trial (values per trial). Trauma levels were quantified as : no trauma = 0, moderate = 1, serious = 3.

robot drifted unexpectedly[20]. However, with the virtual RCM functioning properly, subjects were able to operate without breaking any pipettes. Of course, any risk of broken sharps on the retina has to be minimized for clinical applications to be viable.

Through observation, the cannulation technique which worked most successfully involved first pushing down on the vessel at a moderately steep angle, then translating along the axis of the vessel to puncture, and finally lifting up and pushing axially while simultaneously rotating the tool shaft down toward the vein axis in a hooking motion. This results in the pipette tip pressing on the top side of the inner section of vein, with the force along the axis of the vessel translating the tip farther along, rather than puncturing the bottom side. The future injection methods may include a set of wide range of bent-tip pipette angles to accommodate various approaches required to cannulate vessels in different orientations and locations on the retina. Or a different approach can be taken, where a forceps is used to deliver an unconstrained pipette (e.g., with integrated medication injector) to the vessel. Challenges in working with the bent pipette tip include devising a method to determine the exact location of the pipette tip and also in resolving micro-scale fluid dynamics issues.

The results of the survey (Table 4.5) assessing the subject's experience with the robot indicates that subjects generally felt that the robot was intuitive to use and also aided the cannulation in terms of speed, accuracy and general helpfulness. Most subjects responded that at the beginning of the trials, they felt more comfortable using the robot to cannulate and their preference for the robot remained even after completion of all trials (64 % of the subjects). Some felt that that the robot was too sensitive to small motions and force feedback would be helpful. Freehand was described as “tiring” and that the 60 second requirement was much easier to maintain in robot-

[20] This experiment also exposed an unforeseen practical issue with the halogen light source heating up the FT sensor causing incorrect interpretation of strain gauge values, affecting robot interaction. Although in practice the use of such direct light is unlikely, a temperature compensated sensor should be used for safety reasons.

Metric	Value (1-low; 5-high)
Difficult to Learn	1.63
Speed	3.8
Accuracy	4.0
Overall	4.0

Table 4.5: Results of subject feedback on robot assistance in vein cannulation.

assisted trials. While some felt freehand was "easier to manipulate and insert into veins," others felt the robot was easier to move and aided in precision.

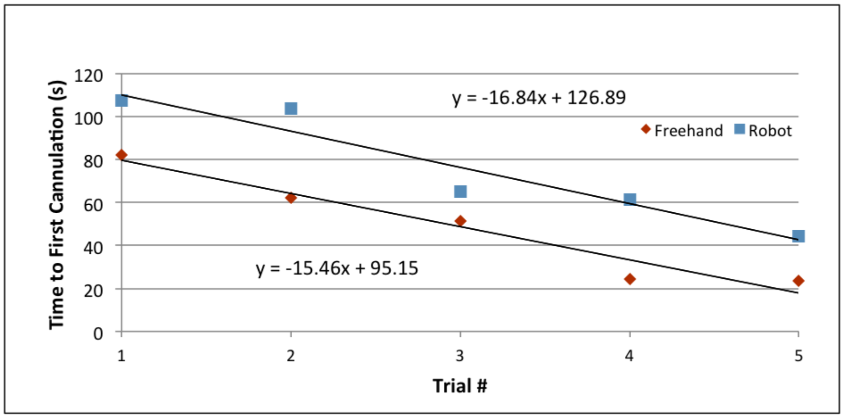
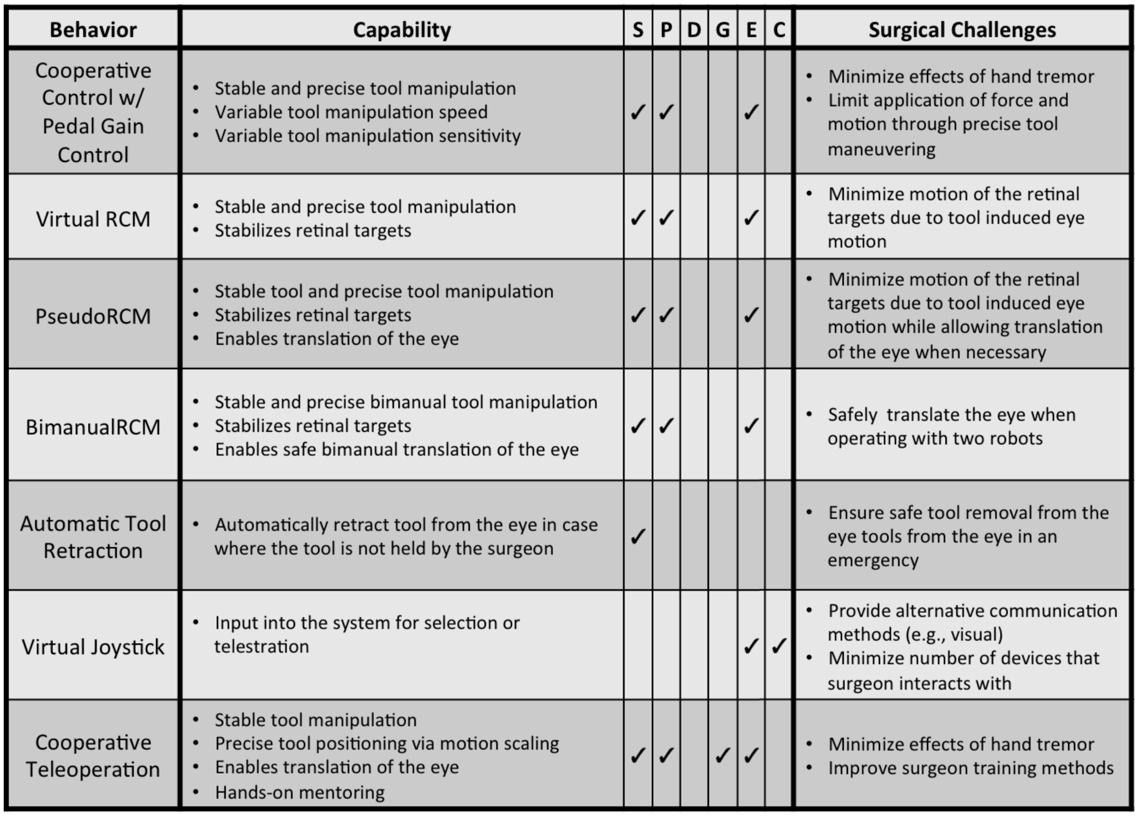


Figure 4.24: Effects of learning based on first cannulation time vs. trial #, comparing the robot-assisted and freehand test scenarios.

Figure 4.24 demonstrates the learning effect by comparing first cannulation time for consecutive trials, with and without robot assistance. The improvement rates for both methods are approximately the same. The likely reason for the constant offset favoring the freehand method is the low speed of the robot in comparison to the speeds of the unassisted hand motions. A faster robot with more transparent control (e.g., user adjusted higher gains) is expected to reduce the time to first cannulation to approach those found in freehand trials. With further training, the robot-assisted cannulation results are expected improve and at some point outperform that of freehand approach due to surgeon's inherent hand tremor magnitude.

4.11 Conclusions and Future Work



Behavior	Capability	S	P	D	G	E	C	Surgical Challenges
Cooperative Control w/ Pedal Gain Control	• Stable and precise tool manipulation • Variable tool manipulation speed • Variable tool manipulation sensitivity	✓	✓			✓		• Minimize effects of hand tremor • Limit application of force and motion through precise tool maneuvering
Virtual RCM	• Stable and precise tool manipulation • Stabilizes retinal targets	✓	✓			✓		• Minimize motion of the retinal targets due to tool induced eye motion
PseudoRCM	• Stable tool and precise tool manipulation • Stabilizes retinal targets • Enables translation of the eye	✓	✓			✓		• Minimize motion of the retinal targets due to tool induced eye motion while allowing translation of the eye when necessary
BimanualRCM	• Stable and precise bimanual tool manipulation • Stabilizes retinal targets • Enables safe bimanual translation of the eye	✓	✓			✓		• Safely translate the eye when operating with two robots
Automatic Tool Retraction	• Automatically retract tool from the eye in case where the tool is not held by the surgeon	✓						• Ensure safe tool removal from the eye tools from the eye in an emergency
Virtual Joystick	• Input into the system for selection or telestration					✓	✓	• Provide alternative communication methods (e.g., visual) • Minimize number of devices that surgeon interacts with
Cooperative Teleoperation	• Stable tool manipulation • Precise tool positioning via motion scaling • Enables translation of the eye • Hands-on mentoring	✓	✓		✓	✓		• Minimize effects of hand tremor • Improve surgeon training methods

Figure 4.25: List of robot-based Behaviors with specific capabilities they provide, and the corresponding surgical challenges that they address. Behaviors are also characterized by capability types: S-Safety, P-Precision, D-Diagnostics, G-Guidance, E-Education, C-Communication.

This chapter focused on the robotics aspect of the *eyeSAW* system and presented Behaviors with capabilities that have been developed for vitreoretinal surgery applications. These Behaviors are summarized in Figure 4.25. Most of the capabilities focus on augmenting surgeons ability to precisely and safely maneuver the instruments inside the eye, i.e., to minimize the effects of hand tremor, provide micron-scale tool positioning, and stabilize the retinal targets by limiting the effects of tool interaction with the sclera, etc.

The current iteration of the EyeRobot has been shown to be an effective aid in microsurgical tasks; however, it still requires more conceptual development and engineering refinement for clinical applications. The following are important requirements and issues that should be considered in the development next generation of the device and associated Behaviors:

- The current EyeRobot is a development platform that is very adaptable and flexible for in vivo animal experimentation. For clinical applications it requires more design iterations to address stringent clinical safety requirements and improve on the human-machine interface, which should be ergonomically comfortable and intuitive to use. The device itself should integrate seamlessly with the operating table and the operating room. For example, the ±45° tilt range might not cover all the extreme maneuvers performed by surgeons but should be sufficient for general vitreoretinal operations. Figure 4.26 shows the EyeRobot2.1 during a typical rabbit experiment.

- Introduction of robotic technologies into the surgical flow adds to the operation time, which was apparent during the rabbit experiments. Yet this disadvantage can be outweighed by the benefits that a robotic assistant provides. Besides reducing hand tremor and improving precision, the robot serves as an excellent platform for communicating information from intraocular sensors and incorporating that information into its control to improve safety or assist in a difficult surgical task. The following chapters present some of these methods.

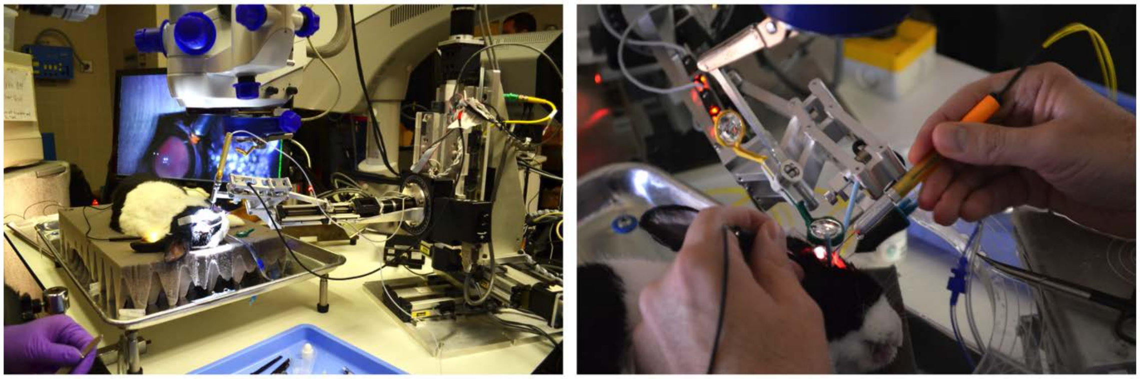


Figure 4.26: JHU EyeRobot2 used in rabbit experiments.

- Through rigorous interaction testing it was found that surgeons desire to always be in control, and any assistive technology has to work in completely predictable ways, whether a robot or any other device.

- Cooperative control transparency can be improved by increasing force control gains to provide a more natural (high transparency) response while ensuring stability through filters and safety checks. This should also reduce hand fatigue by minimizing handle input forces. A new generation of the robot should include a lighter robot body and faster actuators to generate a more responsive manipulation through higher accelerations and velocities[21].

- The system should provide a clear and immediate audio feedback for emergency robot events. e.g., reached end of travel. The surgeons prefer simple audio clips or clear verbal feedback.

- Another complementary and practical behavior would be to provide information regarding the travel limits of the robotic device relative to the surgical workspace. During development the robot was inadvertently positioned so that one of its travel limits would be encountered

[21] For example, the top speed on ER1 was 5 mm/s and in the recent ER designs it is to close to 50 mm/s, which provides a more natural response.

later during the experiment. A visual or audio Behavior could provide a warning regarding such possibility when the procedure is started.

- Symmetrical joint performance is required for intuitive response, e.g., accelerations and gains should be the same for all translational motions.
- An intuitive and quick tool release mechanism is essential for convincing the surgeons about the safety of the robot.
- Design premise of the mechanical RCM is valid. Although, surprisingly the major benefit is in its effect on minimizing translational stage velocities, which improves overall robot stability and speed.
- The Pedal Gain Control has shown to be very intuitive and effective at modulating the force control gains. An pedal-less alternatives, such as the non-linear gain presented by Kazanzides [115] should be investigated in more depth. The variable gain can also be achieved with pressure sensors on the tool handle, where higher finger (squeeze) pressure generates higher gains, or vice versa. Similarly, a contact location sensor embedded in the handle shaft could be used to modulate the gains, e.g., more distal handle contact generates higher gains.
- A pedal-less robot control suffers from undesirable drifting when it is unattended. This can be addressed with a contact sensor on the tool handle so the robot will only move when the tool is held by the surgeon. Another approach uses only the existing force/torque sensor to detect hand-tool contact by continuously analyzing input data for human specific tremor or other distinguishing force profiles.
- The PseudoRCM Behavior is a very simple way to impose a soft constraint on the operator without requiring additional input (e.g., button) to translate the eye. With an external

tracker, or a tool with integrated sclera contact point sensor this behavior response can be further refined to consider the real sclerotomy interaction point. Future work should include comparison of the RCM methods in a simple trajectory tracing (e.g., circle) tasks similar to the ones presented by MacLachlan et al. [78] but with realistic sclerotomy model.

- A 3D tracking system to precisely determine the location of the eye and the surgical instrumentation would enable more safety related virtual fixtures, and potentially for surgical skill assessment. Such systems is currently in development and uses an array of miniature video cameras positioned around the globe of the eye [22].

- A tracking system can facilitate the development of additional safety-orientated virtual fixtures that model geometric safety zones so that the robot prevents the surgeon from colliding with critical structures such as the lens, a common surgical complication that causes cataracts. Additionally, the tool location within the eye can be used to automatically adjust the force control gains, or control robot speed limits, e.g., the tool speed is limited to 1 mm/s when the tool tip is within 3 mm of the retinal surface.

- An alternatives method for adjusting the RCM location is to bias the user interaction to mostly rotations and insertion along the tool axis, but allow manipulation of the RCM position when forces in the transverse plane of the tool exceed a threshold.

- End-point sensing will be essential in active robotic assistance, especially in the cases where the robot is holding long tubular instruments that are susceptible to flexing under natural loading from interaction with tissues. This causes severe errors in tool tip position estimate via forward kinematics calculations. For example, it is not uncommon to notice a 1 mm tool tip error due to sclerotomy interaction forces.

[22] Johns Hopkins University, 600.466) CIS2 course project from 2013 under the guidance of Marcin Balicki.

- Teleoperation setup would require additional slave hardware to actuate the tool axis rotation and also to actuate the tool function. Additionally, a safe, remotely actuated tool retraction/ejection will be required.

- Virtual Joystick is very practical because it is a quick sterile way to interact with the rest of the *eyeSAW* system. In the future, for some scenarios it would be possible to add haptic feedback. The development of the Virtual Joystick also demonstrates how an integrated system facilitates the discovery and rapid prototyping of multiple uses of existing system functionality.

- The robot scripting interface as been very useful during the development of the Behaviors, to control new robot parameters before they were integrated into a GUI, for instrument calibration and for automatic testing, e.g., repetitive tissue loading.

- The surgeon is faced with many parameters and permutations of robot assistance methods. The ability to design the ideal robot configuration for particular tasks before the procedure and quickly switch between them during run-time is essential. Additionally, adjusting the robot control parameters during the task will extend its use in a variety of other surgical applications.

- Freehand cannulation turned out to be more feasible than initially expected with subjects being able to guide the pipette to the indicated vessel and repeatedly enter the vessel. However, the results indicate that robot assistance can make the difficult aspects of the procedure more successful, such as keeping the cannulation. The vein cannulation experiment needs to be repeated with expert users, especially those that have some experience with robotic assistants.

- Future research should investigate supervisory control where the surgeon defines the task the robot executes it in an autonomous fashion.

4.12 Recapitulation of Contributions

EyeRobot (2 and 2.1) (Section 4.3) Significant contributions were made to design and construction of robot prototypes based on new design specifications gathered through extensive use and testing of preceding EyeRobot versions. The salient examples are: a (fixed) mechanical RCM, which stabilizes overall robot motion by minimizing motion of the heavy translational stages; transparent interaction requires large accelerations and velocities, which requires more powerful motors or lighter components; tool exchange and rapid retraction, which is essential for robot acceptance by the surgeons, achieved by a mechanical quick release tool adapter, and automatic tool retraction. The electronics requirements were driven by the fact that the robot controller needs to be both inexpensive, and also compact to facilitate transport. This was achieved by an off-the-shelf Ethernet-based motion controller and a laptop. The software also needed to be flexible to facilitate rapid development of new control methods, and be able to interface with existing and new hardware and software applications. This was achieved by utilizing cisst/SAW component infrastructure and standardizing the eyeSAW system interfaces. In addition, a custom scripting interface (using Python) provided a real time prototyping environment that was used extensively. The platform has been very robust, and can be easily duplicated at a nominal cost, since it is mostly from off-the-shelf parts. It can be used in many different surgical applications, one of which is an ongoing research endeavor to aid cochlear implantation, a high-risk and difficult surgical procedure. **Credit**: This is a full BRP team effort, with specific designs contributed by Dr. Iulian Iordachita, Dr. Russell Taylor, Marcin Balicki, and Dr. Peter Kazanzides, with input from two surgeons: Dr. Handa, and Dr. Gehlbach. Electronics and general robot system design is by Marcin Balicki. Software architecture was developed by Marcin Balicki, while the virtual fixture framework is based on work by Ankur Kapoor. The RCM mechanisms were designed by Iulian Iordachita, Daniel Roppenecker, and Xingchi He. The mechanical quick release adapter was designed by Dominikus Gierlach. Many thanks to Jason Hsu, Robert Romano, Kevin Olds, and many other students for assisting with the assembly of the EyeRobot2.

Variable Gain Control (Section 4.3) An important improvement over the original GUI that adjusted EyeRobot's cooperative (force) control gain. This improvement solves the problem of undesirable robot motion due to gravity or dynamic environmental forces when the tool is not held by the surgeon. By using a foot pedal, the surgeon modulates apparent stiffness and speed of the robot. The variable input pedal provides a simple, intuitive, and direct way for the surgeon to alternate between fine and fast control of the robot. This also instills confidence in the device by having the surgeon choose when the robot is on. Such variable gain control can be applied to other cooperatively controlled robot platforms, and has already been used in otology applications. **Credit**: Marcin Balicki.

Pseudo RCM Robot Interaction (Section 4.5) A novel alternative to a common fixed remote center of motion (RCM) constraint of the surgical robot was created that does not require precise definition and tracking of the sclerotomy locations relative to the robot coordinate system. The Pseudo RCM biases the robot control so that the tool tends to rotate about a point 25 mm from the tool tip. This causes minimal translation of the eye and hence more stability of the retinal targets. However, the surgeons can easily override this "soft" constraint by applying more lateral force on the tool, and translate the eye to change the view of the retina. Such capability can be applied in cooperative control robot applications where the virtual fulcrum needs to be translated often, and is especially applicable where the tool depth inside the cavity is constant. **Credit**: Marcin Balicki.

Bimanual EyeRobot Assistance (Section 4.6) The first bimanual cooperative robot control for microsurgery was demonstrated using two EyeRobots to simultaneously manipulate a simple hook instrument and a light pipe in an eye phantom in a human skull model. In order to provide adequate safety to limit excessive forces on the sclera, an innovative virtual fixture was presented that ensures a constant displacement between the two sclerotomies used to insert the instrument into the eyeball. This approach can be applied in any surgical scenario where motion of multiple instruments is constrained by access ports and a coordinated effort is required to safely adjust the patient's anatomy. **Credit**: Marcin Balicki.

Cooperative Robot Teleoperation (Section 4.9) The first teleoperation of a cooperatively controlled robot for eye surgery was demonstrated, with two operators sharing the control a single instrument. One operator was at a master console, remotely controlling the EyeRobot, while the other operator was at the surgical site, manipulating the instrument jointly with the EyeRobot. In this scenario, both surgeons' relative input to control the instrument can be adjusted, permitting either of the surgeons to have more control, which is useful in an educational setting. This concept combines the strong points of two paradigms into a single surgical system: cooperative control, which is intuitive, fast and provides hands-on-tool safety; and the master-slave system, which is commonly thought to be more ergonomic and provides motion scaling. This paradigm has applications in surgical training, and can enable a specialist surgeon to operate on many patients from a single location. **Credit**: Concept developed by Marcin Balicki. Prototyped with help from Tian Xia to incorporate da Vinci master and Balazs Vagvolgyi to incorporate remote stereo visualization.

Automatic Tool Retraction (Section 4.7) A cooperative robot control method was developed for automatically retracting the surgical instrument from the human eye when the surgeon is no longer holding it. The method does not require any additional sensors and is easily implemented in software. It is considered an optional safety feature and can be used in situations where the proximity of the instrument held by the robot may accidentally collide with delicate human anatomy if left unattended, e.g., cochlear implantation. **Credit**: Concept by Marcin Balicki and Dr. Russell Taylor. Prototype developed by Marcin Balicki.

Virtual Joystick (Section 4.8) A novel dual use of the EyeRobot's handle force sensor was successfully demonstrated. By halting the motion of the robot, the forces applied by the surgeon on the handle can be interpreted as a velocity or position based multi-dimensional joystick. This provides the surgeon with a natural and immediate input into the system. Common uses include telestration, menu navigation, and on-the-fly adjustment of parameters. **Credit**: General concept was developed in collaboration with Anton Deguet. Interaction design and implementation by Marcin Balicki.

Vein Cannulation Experiment (Section 4.10) A study was conducted to explore robotic assistance in retinal vein cannulation. It improved on a prior experiment by considering the sclera constraint, and by using bent-tip pipettes that are more geometrically compatible with the procedure. The results show that robot-assistance significantly increases the time the pipette is held in a vessel, which is important for adequate drug delivery. This indicates that an EyeRobot-like device can be beneficial in delicate tissue micro-manipulation, requiring prolonged periods of tool stability. **Credit**: Designed by Marcin Balicki, conducted by Bryce Chiang, James Gao, Yinfei Xu (JHU CIS2 Course, 600.446).

Chapter 5

Force Behaviors

Vitreoretinal tissue manipulation requires precise manual dexterity, fine visual-motor coordination, and application of forces that are well below human tactile sensation [36, 76]. Unstable manipulation and unknown excessive forces applied to the ocular tissue during procedures such as epiretinal membrane peeling are associated with the risks of retinal hemorrhage and tearing, with potentially irreversible damage that results in vision loss. These can be attributed to physiological hand tremor, surgeon fatigue, poor visual and kinesthetic feedback, patient movement, as well as surgical inaccessibility, tissue fragility and the flexibility of delicate (20–25 Ga) surgical instruments. About 75% of all forces applied during in-vitro retinal manipulation in porcine cadaver eyes are found to be below 7.5 mN and only 19% of the events with this force magnitude can be felt by the surgeons [36]. This indicates that force sensing in microsurgery can provide valuable, safety-critical, yet otherwise imperceptible information to the surgeons. Current practice requires the surgeon to keep operative forces low and safe through slow and steady maneuvering. The surgeon must also rely solely on poor visual feedback provided by a conventional stereo-microscope that complicates the problem, as it takes time to detect, assess and then react to the faint cues, which is especially difficult for novice surgeons.

During membrane peeling, surgeons manipulate the peeling tissue at very slow instrument speeds, observed to be in the 0.1–0.5 mm/s range[1], while visually monitoring local surface deformation that may indicate undesirable forces between the surgical instrument, the membrane, and the retina. A capable surgeon reacts to such cues by retracting the instrument and re-grasping the tissue for an alternate approach. This task is extremely difficult to master due to nearly imperceptible visual cues and a requirement for very precise visuomotor reflexes. Involuntary patient motion, inconsistent tissue properties, high or variable tool velocities, and changing manipulation directions can dramatically increase undesirable forces applied to the delicate retinal tissue. Actively sensing, communicating, and even actively limiting these forces has the potential to significantly improve surgical precision and diminish surgical complications.

There have been some attempts to measure millinewton-scale tool-to-tissue interaction forces with microsurgery instruments that have sensing elements built into the handle [131] and to incorporate sensed tool-tissue forces into cooperative control (e.g., [104, 132]). Handle-mounted tool force sensors are not practical for vitreoretinal surgery because the shaft of the instrument is inserted through the sclera. The friction between the tool and the trocar, and lateral forces from the translating the sclera with the tool can significantly attenuate or distort the propagation of the forces to the tissues inside of the eye. This limitation may be addressed by incorporating force-sensing elements into the shaft section of the instrument, which is typically located inside the eye, as presented in Section 5.1.

Once the interaction forces are measured, there are a number of methods to present the

[1] The peeling speeds were estimated directly from surgical videos which are much lower than the speeds reported by Riviere et al. [130] for other surgical tasks.

real-time information to the surgeon. One way is to incorporate the information into the control scheme of the robotic assistant to communicate the forces at the tool tip. For example, Kumar and Berkelman et al. [104, 132] used such an endpoint micro-force sensor in a force-scaling cooperative control method that generates robot response based on the scaled difference between tool-tissue and tool-hand forces. This was extended by Roy et al. [133] to adaptively estimate the environment compliance during robot co-manipulation tasks to render the exact force reflection and force scaling for user-applied forces. Grace et al. [88] presented various modes (on a 1 DOF master-slave system) to aid in a surgeon's perception of an event in micro-vessel piercing by artificially emphasizing particular "features" of the force signal on the master manipulator. A more common approach is to use a tele-operated system like [85, 86] that measures tool-to-tissue forces at the slave, scales them up by a factor of 100, and relays them to the surgeon via the master manipulator. These robotic approaches combining hand tremor reduction with precise motion control and end-point force sensing feedback are a more complete approach to the problem of fine micro-manipulation of delicate tissue.

Other methods involve sensory substitution where the sense of touch/force is communicated via another sensing modality, such as visualization. In fact, in current practice, surgeons indirectly assess the relative stress applied to tissue via visual interpretation of changing light reflections from deforming tissue. Behaviors involving "visual sensory substitution" are discussed in Chapter 7. However, there are many foreseeable microsurgical scenarios in which the primary visual feedback will be impaired or obscured. Providing the force information using another sensory feedback, such as audio, could be very beneficial. Kitagawa et al. showed that auditory feedback representing force in a complex surgical task improves robot-assisted performance and suggests that continuous and real-time feedback is more effective than discrete, single event information [134].

This chapter presents a device to measure micro-forces applied to eye tissues and behaviors that use this real-time information to comply to the environment and to augment surgical performance by providing intuitive feedback aurally or haptically[2]. The hypothesis is that this will improve the performance of the surgeon and diminish surgical complications.

5.1 Micro-Force Sensor

Vitreoretinal microsurgical applications introduce certain limitations on the exact choice of force sensor by demanding sub-millinewton accuracy required to sense forces that are routinely less than 7.5 mN [36]. Furthermore, miniature instrument size is necessary to ensure proper insertion through a 0.5–0.7 mm trocar, and the force sensor must be able to obtain measurements at the instrument tip, which is below the sclera, inside of the eye. A tool design developed by Iordachita et al. [136] based on Fiber Bragg grating (FBG) sensors adheres to these requirements. It is a 0.5 mm diameter, +30 mm long tube with three optical (FBG) fibers bonded into machined channels along the tool shaft (see Figure 5.1)[3]. FBGs are robust optical sensors capable of detecting changes in strain, without interference from electrostatic, electromagnetic or radio frequency sources. Based on the axial strains due to tool bending (see Figure 5.1B), the instrument senses force at the tool tip in the transverse plane (2 DOF), with a sensitivity of 0.25 mN in the range of 0.25–60 mN. Adding redundant FBGs to the design creates a theoretically temperature insensitive force sensor, but due to fabrication imperfections, temperature effects may be a factor which can be minimized by proper calibration and biasing. A simple version of the sensorized surgical instrument, shown in Figure 5.1D, is a micropick with an ergonomic handle that can be mounted in a robot's

[2] Many of the force-based robot feedback methods have been developed in a collaboration with Ali Uneri [107,135]

[3] Iulian Iordachita developed the earliest prototypes of the tool, and more recently has been working with Xingchi He to develop a 3 DOF version.

quick-release tool holder or used in freehand fashion. Based on this concept, more sophisticated instruments providing forceps functionality are in development [137, 138]. The raw FBG samples are acquired by a commercial FBG interrogator (sm130–700, Micron Optics Inc.) at 2 kHz over a dedicated TCP/IP network. Custom C++ software (SAW component) processes the raw samples and provides the resolved force vector to components in the system. A simple 1 DOF force sensor's dynamic properties were analyzed with a 20 mN step input in air and presented by Sun and Balicki et al. in [139]. The approximate signal rise and settling times are 3.5 ms and 180 ms, respectively and the overshoot was measured to be 6.6 mN or 33%. The dynamic range is 57 dB while the natural resonance frequency is 328 Hz and decay time constant is 47 ms, producing a Q-factor of 24.2. In practice, the negative effects of the natural resonance of the sensor are minimized with a low pass filter and the significant dampening provided by the intraocular fluid and tool-tissue

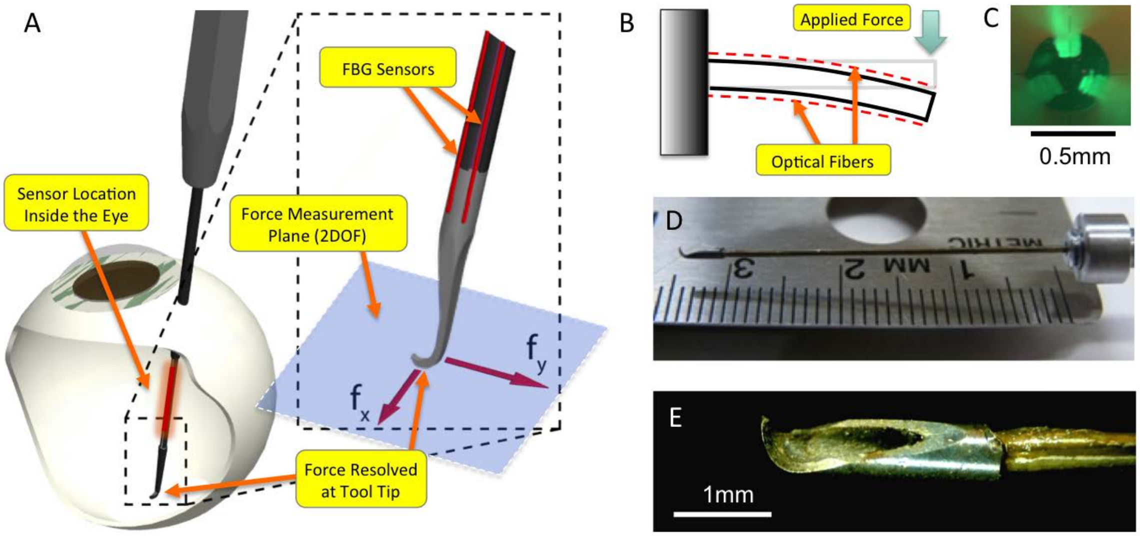


Figure 5.1: A) Location of the sensing section of the FBG when tool is inserted into the eye. B) Sensor is modeled on the strains in a cantilever beam under load. C) Cross-section of the tool shaft showing channels (at 120°) where FBGs are bonded. D) Assembled force tool shaft. E) Magnified view of the hooked micropick prototype.

interaction. The dynamic response of the sensor should be considered when the force readings are used as feedback in robot control.

5.2 Membrane Peeling Forces

To provide beneficial real-time feedback to the surgeon, it is necessary to measure force generated during the surgical maneuver, as well as reference a library of forces that induce injury during that maneuver. The following experiments[4] were performed to test whether the FBG-based force sensing instruments could reliably detect millinewton scale forces and distinguish the forces generated between normal surgical maneuvers and a known complication. They are summarized here and described in more detail by Sunshine and Balicki et al. in [140]. The first experiment was intended to test the sensitivity of the force sensor. It involved peeling the inner shell membrane

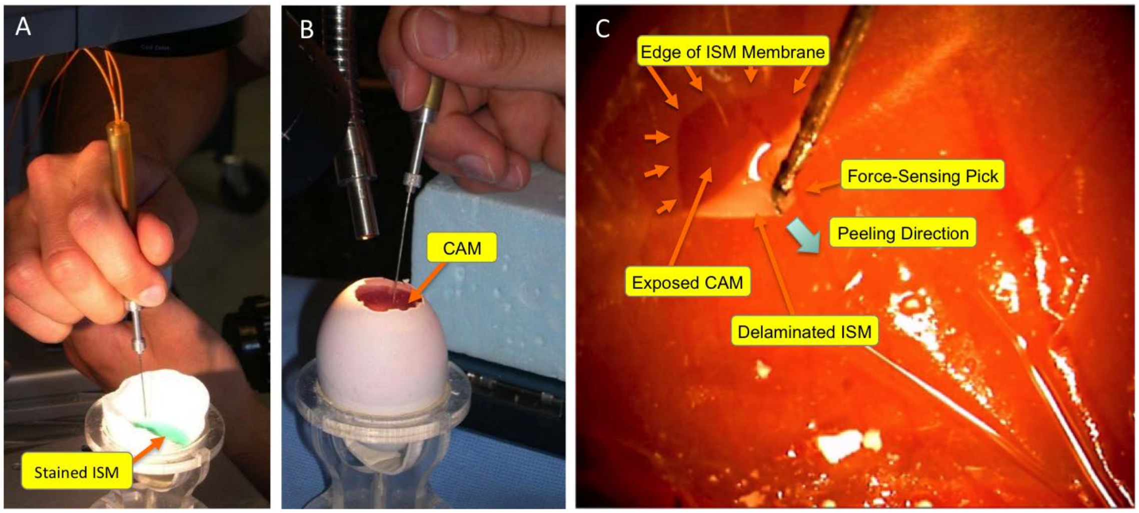


Figure 5.2: A) Force-sensing pick used in peeling of ICG-stained ISM in a raw egg. B–C) Force-sensing pick used in peeling of ISM in a CAM model.

[4] The membrane peeling experiments were led by Sarah Sunshine. The author designed the force collection software, the experimental setup, and assisted with the experiment design and analysis.

(ISM) in a raw egg (see Section 3.3.6). The goal was to maintain a steady force and peeling velocity without tearing the membrane, a maneuver analogous to avoiding a retinal tear. After an initial training period (n = 20 attempts), a single surgeon peeled 11 egg shells. In each trial, the surgeon engaged an egg membrane edge with the micropick (see Figure 5.1E), and delaminated the membrane from the shell while maintaining the instrument shaft in a vertical position relative to the surface of the membrane. This ensured that only the forces in the measurement plane were generated[5]. The minimum force generated during the membrane peeling trials was 0.2 mN, while the average force was 0.5 ± 0.4 mN with range of 0.2–1.5 mN. The standard deviation serves as a quantitative indicator of the variability of forces generated during membrane peeling.

A follow-up experiment was performed to assess the ability of the force sensing instrument to record retinal injury. To simulate a retinal tear, the same surgeon intentionally ruptured the lower chick chorioallantoic membrane, which simulates the neurosensory retina (CAM, see 3.3.7), while peeling the upper ISM, which mimics an ERM. If sufficiently traumatized, the CAM will bleed or tear, which simulates a retinal tear and is easily identified. For comparison, the first set of peels (N = 6) delaminated the ISM without injuring the lower CAM. The maximum force generated during the peeling was 4.1 mN, while the average force was 2.8 ± 0.2 mN in the range of 1.3–4.1 mN.

The second set of peels (N = 6) involved intentionally damaging the CAM during delamination of the ISM by increasing the velocity of membrane peeling. The minimum force that created an injury to the underlying CAM was 5.1 mN, which is significantly greater (P = 0.0001) than the maximum force generated during nontraumatic delamination (4.1 mN). The average force was 7.3 ± 0.5 mN in the range 5.1-–9.2 mN.

[5] With this instrument, we can accurately measure forces within a 1.5% error at a 10° angle between instrument shaft and tissue plane. For a 20 mN force, the error introduced would be 0.3 mN, which is comparable to the 0.1 mN instrument sensitivity.

The above experiments illustrated that very small forces were reliably and reproducibly measured in simulated retinal peelings, as well as that small force differences, which are predictive of tissue injury, are detectable. The sensitivity of 0.5 mN is far below that reported for manipulating the retina (i.e. 7.5mN as reported by Gupta et al. [36]).

For additional studies a more practical force-sensing instrument was developed by He and Balicki et al. [138]. It includes forceps functionality as shown in Figure 5.8A, and has the same 2 DOF force sensing capability and sensitivity as the micropick. This new instrument was used to collect information about the maximum peeling forces in the peeling of ISM in a 12-day-old chicken embryo (see Figure 5.3A). The maximum forces range from 6.07 to 34.65 mN and the mean is 16.11 mN ± 9.14 mN [138]. Figure 5.3B shows a representative force trace of one of the trials. The above results demonstrate that the force-sensing instrument design can reliably detect very small forces during prototypical surgical maneuvers as well as during associated complications. The results also indicate that biological membranes may behave differently during peeling in terms of observed forces during delamination.

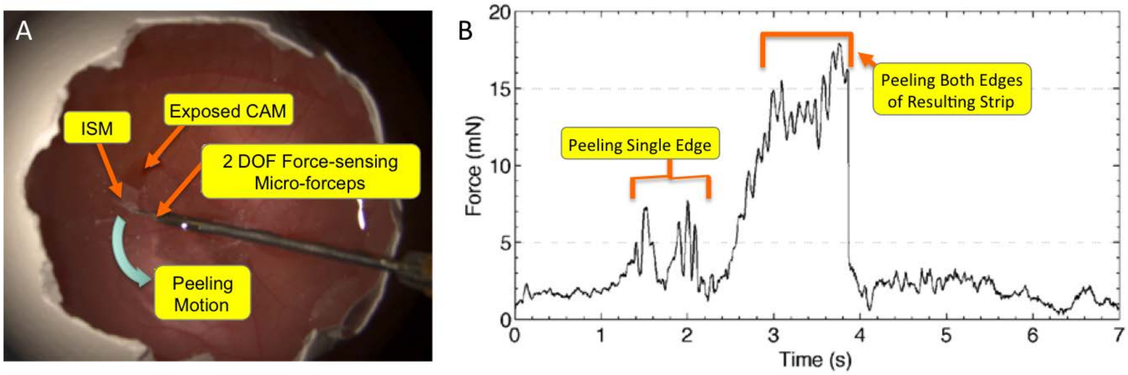


Figure 5.3: A) Circular motions used to delaminate the ISM from the CAM using 2 DOF force-sensing forceps B) Sample force profile from a single peeling trial showing two force regions: one when a single edge of the peeled section is delaminated and a second, with higher forces, where the delamination is produces a strip.

In vivo rabbit eyes were used to test the sensor's feasibility and reliability to measure tool-to-tissue interaction forces in a clinical environment. Specifically, the goals were to demonstrate that contact with the sclera did not interfere with force measurements, and to measure the forces during 1) safe delamination of the posterior hyaloid from the retina, and 2) creation of a retinal tear in the posterior segment. The description of the setup and the general rabbit experiments is in Section 3.3.9). The surgeon performed a standard 3-port vitrectomy and lensectomy after which a force-sensing instrument was inserted through a standard 23 Ga port, then re-biased to minimize any effects from significant temperature change or slight tool deformation. To create intentional posterior hyaloid detachment and a retinal tear, the trained surgeon was instructed to manipulate the tissue at a constant rate, and in the force-sensing (X–Y) plane of the instrument. The recorded video and force data were synchronized with timestamps and analyzed in the *cisstDataPlayer* software described in Section 3.4. The results reported by Sunshine and Balicki et al. in [140] indicate that the minimal forces to successfully detach the hyaloid were 6.7 ± 1.1 mN (N = 3, smallest chosen from 12 trials). Figure 5.4A shows a typical force profile, with a gradual increase in force until a plateau phase, followed by a rapid decrease when the hyaloid was separated from the optic

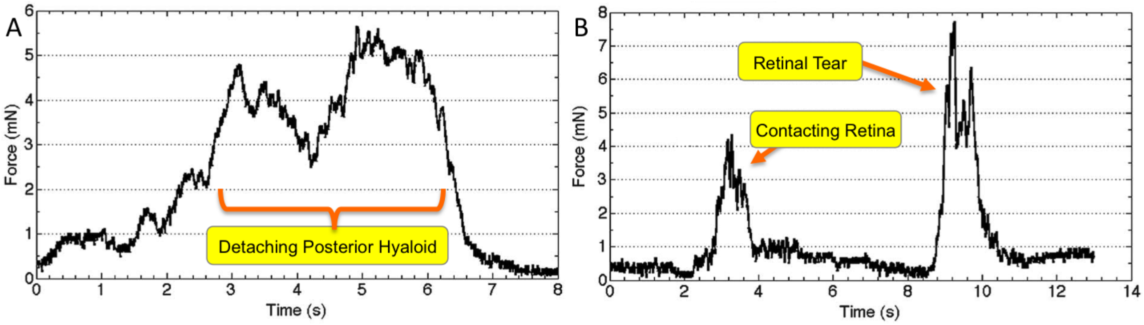


Figure 5.4: In vivo force sensing: A) Representative trace of the raw forces measured by the force-sensing micropick during posterior hyaloid detachment. B) Representative tracing of the forces measured during creation of a retinal tear [140].

nerve. The smallest forces to produce retinal tears were similar: 6.4 ± 1.4 mN (N = 3, smallest chosen from 25 trials)[6]. Figure 5.4B shows a typical retinal tearing force trace. The tear is created between 8–11 s and is characterized by a rapid increase in force, a short plateau phase, and a rapid decrease in force when the retina yielded.

In comparison with detaching the posterior hyaloid, the retinal tear forces increase and decrease much more rapidly, with a steady plateauing of force until the retina is torn. This is likely due to the compliance of the hyaloid attachment architecture to lower the pressure by distributing the forces over a larger area of tissue. In the case of the retina tearing, the force is concentrated in a small region and results in rapid loading of the tissue to a breaking point. Furthermore, it is believed that high velocity manipulation and high frequency force loading of the retina can cause micro-tears, making them more prone to tears during peeling, additionally complicating the already challenging task.

5.3 Audio Feedback Behaviors

Audio sensory substitution provides clear and objective feedback to the surgeon about the forces measured at the tool tip using auditory representation. Kitagawa et al. showed that auditory feedback representing force in a complex surgical task improves robot-assisted performance and suggests that continuous and real-time feedback is more effective than discrete, single-event information [134]. After experimenting with many different feedback modes[7], the following audio sensory substitution methods were developed.

[6] Many of the unreasonably high forces (in excess of 30 mN) recorded are attributed to collisions of the sensitive sensing element of the tool shaft with the sclera at the sclerotomy site when the tool was pulled away from the retina. This often happened without performing any surgical maneuver.

[7] Other feedback modes included: spoken number (e.g., "five" at transition from 4 mN to 5 mN) generated each time the force reading crossed a mN digit; increasing pitch of a looping audio clip with increasing force; and for fun, playing "ouch!" or "ow!" when the force exceeds some threshold

The "four-zone" mode developed by Balicki et al. [135] is depicted graphically in Figure 5.5. The tempos of three different "beep" audio clips with different tonal frequencies are modulated and played back at subject-adjusted volume. They are turned on and off based on four force level zones which represent force-operating ranges selected for their relevance in typical vitreoretinal operations. The audio is silent until 1 mN (f_1) or greater force is measured. From 1–3.5 mN force ($f_1 - f_2$), which is designated to be a "safe" force operating zone, a constant slow beeping is emitted. After 3.5 mN and until 7 mN ($f_2 - f_3$), a "cautious" zone has been designated and has a proportionally increasing tempo of beeps. Any measurement beyond 7 mN falls into a "danger" zone that generates a constant high tempo beeping[8].

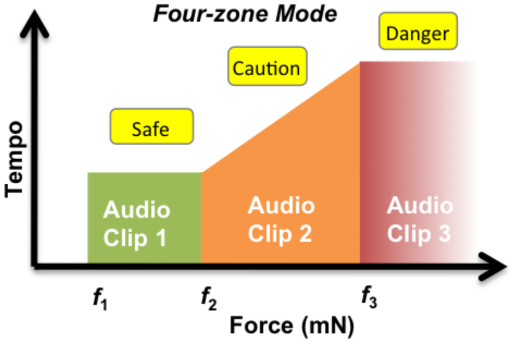


Figure 5.5: Audio force feedback. "Four-zone" mapping algorithm with four regions: $0 - f_1$ silent, $f_1 - f_2$ slow tempo, $f_2 - f_3$ proportionally increasing tempo, and $> f_3$ high tempo.

A preliminary experiment was conducted with a single subject to compare the effectiveness of the above audio feedback method in decreasing mean and maximum peeling forces while minimizing time taken to complete a simulated freehand membrane peeling task. The 10 mm Band-Aid membrane peeling phantom, described in Section 3.3.3, is set up so that the test subject can visualize only the distal tool shaft, which eliminates force cues from tool bending. The test subject

[8] The audio beeps are short audio clips, less than 500 ms long. A different audio clip is used for each feedback zone

is then instructed to engage the 2 DOF force-sensing pick with the perforated membrane, fold it over itself, then peel in a straight line while keeping the tool tip at 1.5 mm above the unpeeled membrane. The test subject was directed to peel the membrane steadily, and as slowly as possible without stopping, in order to minimize the forces applied by the tool. The orientation of the handle was perpendicular to the peeling direction to capture all forces, while still remaining comfortable to the operator. No visual magnification was provided to the test subject. Days prior to the experiment, the test subject was trained for an hour both with and without audio feedback. Five trials of freehand peels and five trials with additional audio assistance were conducted in random order, with five-minute breaks between trials.

The mean force and peeling time results are displayed in Table 5.1 while the typical force profile of a single peeling is shown in Figure 5.6A. In all freehand trials, considerable high force variation was observed, which is attributed to physiological hand tremor. The mean force applied was above 4 mN, with the maximum near 8 mN. Audio feedback helped to reduce mean and large forces but significantly increased task completion time as shown in Figure 5.6A. This indicates that the user consciously attempted to use audio feedback to reduce the forces applied to the sample by moving more slowly when the audio feedback indicated higher forces. During operator training, a

Forces (mN)	Freehand	Freehand + Audio
Mean	4.11	3.80
StdDev	0.97	0.59
Max	7.85	6.21
Time (s)	93.03	125.25

Table 5.1: Force and time results of single subject simulated membrane peeling for Freehand (FH) and Freehand + Audio (FHA) trials.

significant improvement was observed in both the task completion times and force variation decrease in freehand peeling as shown in Figure 5.6B. In comparison, the robot-assisted training[9] did not improve the quality of the peel or affect the rate of peeling due to the lack of objective information regarding the force behavior of the peeling membrane. This must clearly be attributed to audio feedback alone.

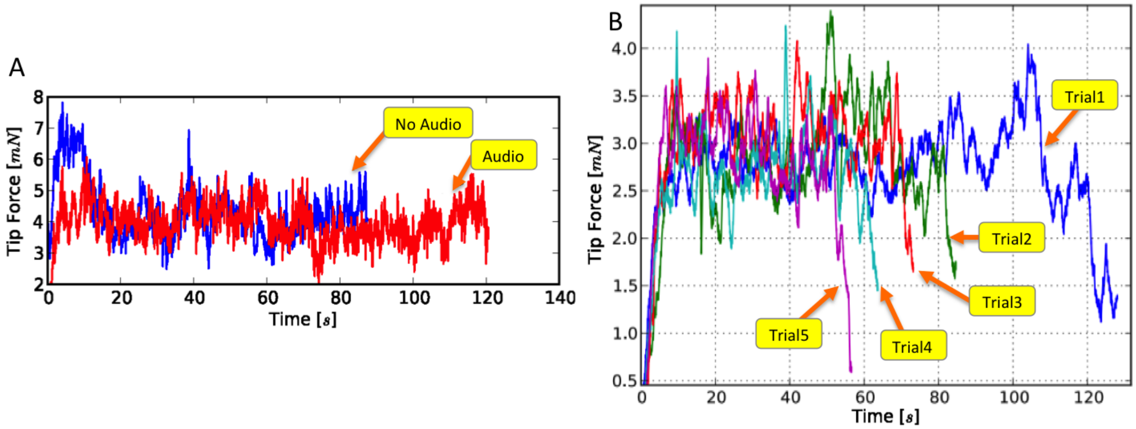


Figure 5.6: Audio force feedback results: A) Example force profile for freehand trial and audio assisted feedback trial. B) Force profiles for five consecutive trials from a freehand training session with audio feedback.

The results from this single-user pilot study provided the basis for a multi-user, IRB-approved study by Cutler and Balicki et al. [141][10]. The human subjects varied in their surgical experience from medical students to expert vitreoretinal surgeons. The general experimental setup (see Figure. 5.8B) was the same as above with the exception of a 2 DOF force-sensing forceps tool [138] shown in Figure 5.8A; and a standard stereo operating microscope for visualization.

For this experiment, the four-zone audio sensory substitution method was simplified to create two new feedback modes. The two-zone "alarm" mode produced no audio feedback until the

[9] The plot of training data for robot-assisted peeling is not shown. Anecdotally, the force profiles vary wildly in completion time and force magnitude due to lack of objective feedback.

[10] The multi-subject study was conducted by Nathan Cutler, while the design of the system and the testing phantom was done by Marcin Balicki.

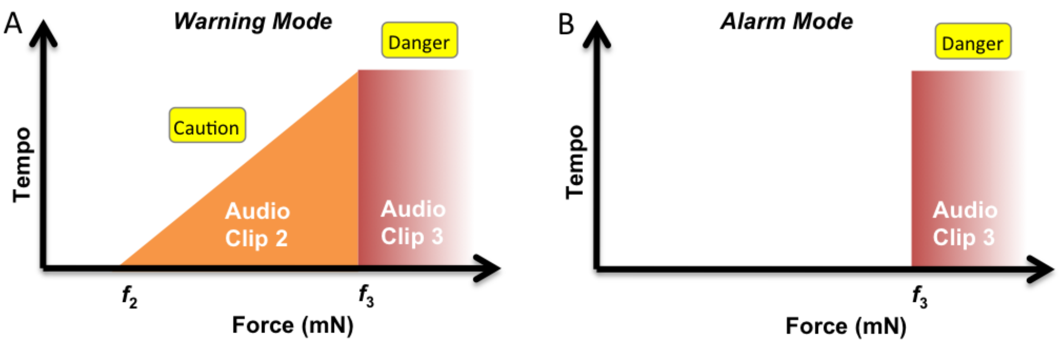


Figure 5.7: Audio force feedback. A) “Warning” mapping algorithm with three regions: $0-f_2$ silent, f_2-f_3 proportionally increasing tempo, and $> f_3$ high tempo. B) “Alarm” mapping algorithm with two regions $0-f_3$ silent, and $> f_3$ high tempo.

tool tip force reached $f_3 = 9$ mN (see Figure 5.7A). At forces of 9 mN and above, a continuous stream of high-pitched “beeps” indicated excessive and potentially damaging forces that were being applied to the simulated tissue. The three-zone “warning” audio feedback mode (see Figure 5.7B) provided more information on a range of forces: below 3 mN, the audio was silent; at $f_2 = 3$ mN the system emitted beeps at a frequency of 1 Hz that proportionately increased in frequency as the force rose from $f_2 = 3$ mN to $f_3 = 9$ mN. The escalating frequency of warning sounds alerted the test subject when the force application was approaching the upper limit of safe forces. At a force of $f_3 = 9$ mN and higher, the audio response was the same as in the “alarm” mode. Each participant peeled a total of 30 Band-Aid membranes switching between three testing scenarios: no audio feedback, warning audio mode, and alarm mode.

The results reinforce those from the prior single-subject study: force-audio feedback reduces the forces generated with improved precision during simulated membrane peeling. Surprisingly, this was true regardless of surgical experience. Although the more experienced surgeons performed the tasks better than those with less surgical experience, they still achieved greater

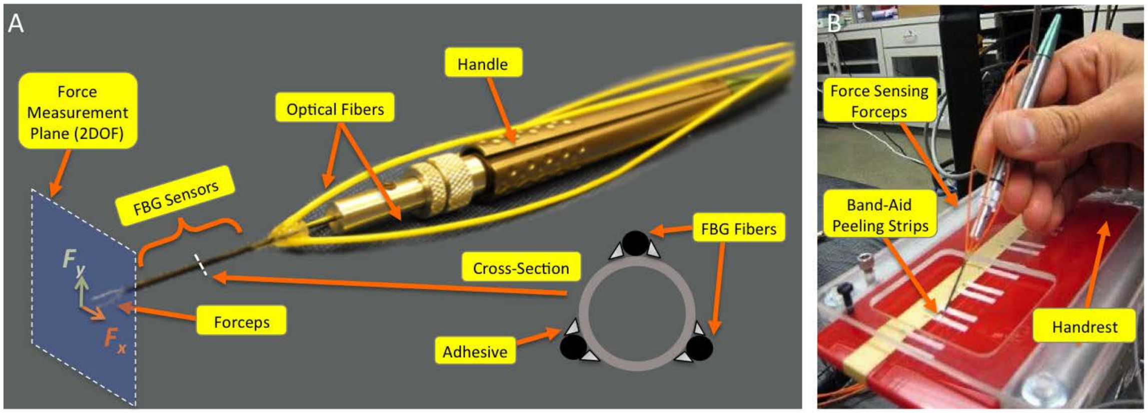


Figure 5.8: A) 2 DOF Force-sensing forceps B) Human subject peeling a Band-Aid membrane peeling phantom with force-sensing forceps.

precision and control with audio sensory substitution. With the exception of maximum force, peeling with audio feedback reduced all performance parameters: average force; standard deviation of forces; and force × time above 9 mN. In a comparison between the two modes, the peels that were performed with the warning audio feedback had significantly lower maximum force and standard deviation of the forces than those performed with the alarm mode, for all test subjects.

Other performance parameters (average force, peeling completion time, time above 9 mN, and force × time above 9 mN) reveal significantly different results when comparing audio feedback mode to no-audio feedback for each test subject. The results in the warning feedback mode produced greater differences, outperforming no-audio feedback in nearly all comparisons. The alarm mode only sometimes achieved significantly better performance than no audio feedback. Surgical experience levels did not correlate with any significant differences between the two audio feedback methods. Additionally, a short training period with audio feedback improved subsequent peeling performance, even when audio feedback was turned off after training, for all measured parameters. Audio feedback resulted in a significant increase in peeling completion time, which was expected since lower force application requires slower peeling rates. Therefore, this results in longer time to completely delaminate a given length of the peeling phantom while minimizing the forces applied to the tissue. In practice, the surgeon should be able to use the force feedback to peel faster when the forces are low and slow down when they are high. This will result in the minimum amount of time required to safely delaminate the membrane.

A qualitative survey showed that 75% of test subjects, independent of training level, preferred the warning audio feedback mode to the alarm mode, since it allowed more precise peeling at a consistent pace, and provided an opportunity for the surgeon to react to the rising forces. Many of the test subjects thought the proportionally modulated sounds of the warning mode instilled confidence, and appreciated the additional quantitative real-time feedback. For others, the alarm mode was sufficient, and less distracting than the virtually constant audio response present in the warning mode. Many commented that having a force-audio feedback system available for residency training would be very valuable. This affirms the above results that showed that even short training sessions with audio force feedback can improve subsequent peeling performance without the assistance. As a result, even if an audio force-feedback system was not available during actual surgery, training with the system in a lab could improve surgeon performance and thus surgical outcomes.

5.4 Tool Compliance Correction

The force application in typical tissue manipulations can exceed 15 mN. This force has the potential to significantly deflect the surgical instrument, as much as 1 mm at the tip, as depicted in Figure 5.9A. In general surgical practice, the surgeon compensates for this disparity between the expected and the actual tool handle-to-tip transformation via the visual feedback loop. However, for some robot feedback control methods, such as virtual fixtures (geometric safety barriers), it is desirable to estimate the actual tool tip location and velocity when the tool is attached to the robot. This is especially important in cases where the robot is used as a position tracking device, e.g., to characterize mechanical properties of the tissue. Using a micro-force sensing instrument, it is possible to provide the exact location of the tool tip under load, under the assumption that the deflections from the trocar interaction are negligible or known. This is typically not an issue in bench-top experiments, and can be addressed with additional sensors on the shaft and the use of an RCM constraint.

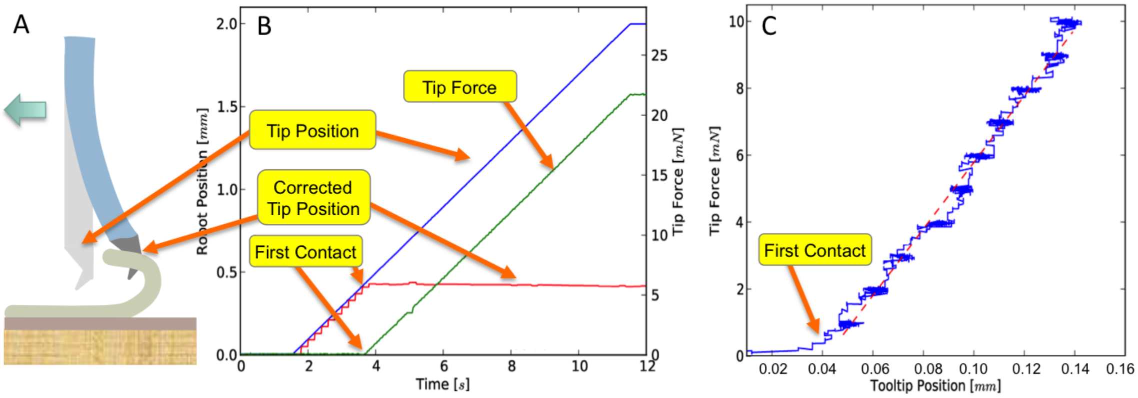


Figure 5.9: A) Schematic of tool bending under load. B) Tip position calculated through forward kinematics is erroneous due to the deflection. C) Verification of correction using a known spring.

The tool compliance was empirically characterized in 1 DOF by laterally translating the tool with the robot for a distance of 2.0 mm, effectively pushing the tool tip against a rigid surface. The corresponding position and reaction force is displayed in Figure 5.9B. The resulting deflection response is linear, following Hooke's law, generating a tool stiffness of 13.76 mN/mm. This stiffness constant (K_t) is used to correct for the compliance in the following fashion.

$$\Delta x = \frac{f_{tx}}{K_t} \tag{5.1}$$

$$\Delta y = \frac{f_{ty}}{K_t} \tag{5.2}$$

$$\Delta z = l_t - \left(\sqrt{l_t^2 - \Delta x^2} + \sqrt{l_t^2 - \Delta y^2}\right) \tag{5.3}$$

where $\mathbf{f}_t$ is the force at the tool tip and l_t is the tool length. Since the amount of correction along the tool axis (Δz) is two orders of magnitude smaller than the transverse plane correction, the tool is approximated as a rigid body pivoting around its base, e.g., handle[11].

To verify the above correction to the position, the same lateral loading test was performed against a spring of known stiffness. Figure 5.9C shows the resulting Force-Strain curve, which is linear, and measures the stiffness of the test spring with 2.8% error. In the case of cooperative control where the user applies a disturbance force to the robot in order to achieve motion, resulting in additional end-effector position errors, the above tool compliance correction method can be combined with work by [142] that addresses the compliance of the rest of the robot body.

5.5 Tissue Force Characterization

One future application of this platform is assessment of soft tissue characteristics through vitreoretinal tissue manipulation. This information may enhance surgical technique by revealing factors that contribute to successful surgical results, and minimize unwanted complications. This capability may also be applied to the dynamic updating of virtual fixtures in robot assisted manipulation.

A preliminary set of experiments conducted by Uneri and Balicki at al. [107] has been

[11] Sclerotomy forces are not considered in this case

carried out to assess the capability of this system in characterizing tissue reaction forces through controlled motion and high resolution sensing. For this, a constant-force robot control mode has been implemented that commands the robot to exert a desired force at the tool tip as described by:

$$\dot{q} = J_t^{+} K_p (f_d - f_t) \tag{5.4}$$

where the error between the desired force f_d and tip force f_t is scaled by a proportional gain K_p and multiplied by the inverse Jacobian of the end effector tip to generate robot joint velocities. Note that f_t has to be transformed into the handle coordinate frame using an adjoint transformation.

After attaching the surgical micropick to the internal limiting membrane of a 12-day-old chicken embryo (see Section 3.3.7) a desired constant force was set on the robot while the corrected displacement of the tool tip was recorded. The robot translated laterally to apply a desired force at the tool. The desired forces were incremented by 1 mN, with a 10 s delay between each increment, up to 10mN. For these trials, the surgical micropick was first attached to the intact tissue and force was applied until failure. The membranes exhibit an average tearing force of 10 mN, after which, continuation of the tear is accomplished with lesser forces (∼6 mN). Figure 5.10B shows a sample force profile for only the initial phase of inner shell membrane loading (up to 10 mN). This is comparable to the freehand peeling forces described in Section 5.2. The characteristic curve obtained reveals a similar pattern to those seen in fibrous tissue tearing [143]. The toe region of the curve, the shape of which is due to the recruitment of collagen fibers, is a “safe region” from a surgical point of view and is followed by a predictable linear response. Yielding occurs as bonds begin to break, resulting in a sudden drop in resistive forces due to complete failure (not shown in the figure). In the surgical setting, this is analogous to the undesirable tearing phase of a membrane being peeled.

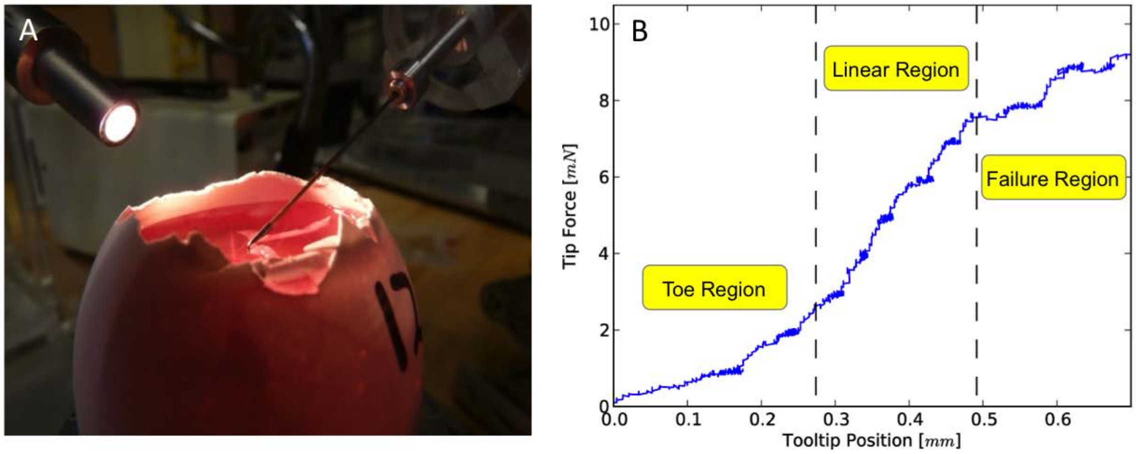


Figure 5.10: A) Robot setup for tissue characterization of chicken embryo inner shell membrane. Robot motion is parallel to the surface and in the sensing plane of the tool. B) Sample force profile showing the initial phase of tissue loading.

5.6 Force Scaling Behavior

The earliest surgical application of micro-force sensing in cooperative robot control is described by Kumar and Berkelman [104, 132]. This approach maps, or amplifies, the human-imperceptible forces f_t sensed at the tool tip to handle interaction forces f_h by modulating robot velocity v as follows[12] :

$$v = K_p(f_h - \gamma f_t) \tag{5.5}$$

This control scheme has been implemented on the EyeRobot2 as one of the available behaviors by Balicki et al. [135]. Prior applications used $\gamma = 25$ and $\gamma = 62.5$ scale factors [104, 132], which are low for the range of operating parameters in vitreoretinal peeling. A scaling factor of $\gamma = 500$ was chosen to map the 0–10 mN manipulation forces at the tool tip to input forces of 0–5 N at the handle, while the force control gain for experimentation was set to a low $K_p = 1$ to produce the very stable motion required for membrane peeling.

[12] The force scaling mode can be easily incorporated into the Virtual Fixtures Framework with $v \to \Delta x$ as the incremental motion objective.

5.7 Force Limiting Behavior

The force limiting behavior developed by Balicki at al. [135] increases maneuverability when low tip forces are present. The method incorporates standard linear cooperative control (see Equation (4.2)) with an additional velocity constraint that is inversely proportional to the tip force. With such scaling, the robot response becomes very sluggish with higher tool tip forces, effectively dampening manipulation velocities. The most applicable constraint parameters were chosen empirically to be $m = -180$ and $b = 0.9$. To avoid zero crossing instability, forces lower than $f_1 = 1$ mN in magnitude do not limit the velocity. Likewise, to provide some control to the operator when tip forces are above a high threshold ($f_2 = 7.5$ mN), a velocity limit ($v_2 = 0.1$ mm/s) is enforced. For demonstration purposes, the constraint is applied directly to the desired velocity **v** rather than incorporated as a constraint in a virtual fixtures framework:

$$v = \begin{cases} V_{lim}(f_t) & \text{if } (f_t < 0) \land (-f_h < V_{lim}(f_t)) \\ V_{lim}(f_t) & \text{if } (f_t > 0) \land (-f_h > V_{lim}(f_t)) \\ K_p f_h & \text{otherwise} \end{cases} \tag{5.6}$$

where $V_{lim}(f_t)$ is the function described graphically in Figure 5.11.

5.8 Comparison of Feedback Methods in Membrane Peeling

The objective of the following experiment, conducted by Balicki et al. [135], is to compare the effectiveness of robot assistance and audio sensory substitution in decreasing mean and maximum peeling forces while minimizing time taken to complete the peeling task. The experiment setup was the same as the one described in Section 5.3 with the robot positioned so the force sensing micropick is ∼1.5 mm above the peeling surface. The instrument is rigidly attached to the handle

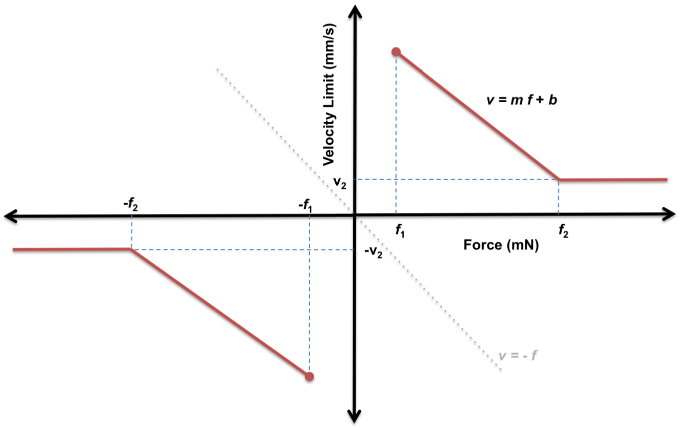


Figure 5.11: Velocity limiting function $V_{lim}(f_t)$. The function is symmetric about the line $v = -f + 0$.

of ER2 and is calibrated in order to transform it into the robot coordinates. To simplify the experiments, the robot motion was constrained to Cartesian translations only. Comparison experiments (not shown here) showed no noticeable difference between trials with and without rotational DOFs. The test subject was thoroughly trained in using the robot (> 3 hours) prior to the trials.

Five trials of each of the following robot feedback control modes were performed with audio feedback, and five without: PV/PVA - conventional cooperative control where the robot moves with *Proportional Velocity* to the force applied at the handle (see Equation (4.2)) with a gain[13] of $K_p = 1$; FS/FSA - *Linear Force Scaling Control*; and VL/VLA - *Velocity Limiting Control.* For comparison, the *Freehand* (FH/FHA) experiment from Section 5.3, where the same test subject peeled the sample without robot assistance, is included. The results are presented in Table 5.2. Figure 5.12 shows typical force profiles during peeling for the various modes in this experiment. In every method tested, audio feedback decreased the maximum tip forces, as well as tip force variability. It significantly increased the task completion time for Freehand and Proportional Velocity control

[13] $K_p = 1$ translates handle input force of 1 N to 1 mm/s tool velocity.

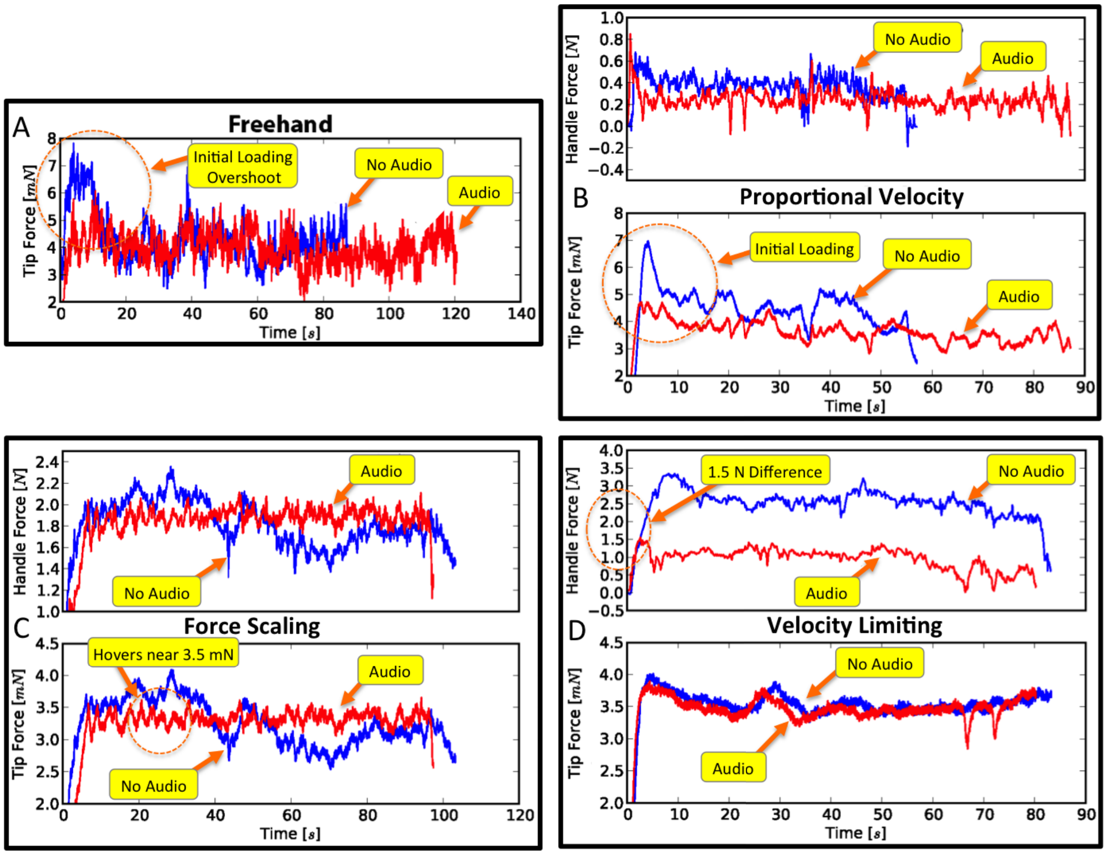


Figure 5.12: Example peel force plots in membrane peeling experiments with audio feedback in A) Freehand peeling; B) Robot-assisted force scaling with associated handle forces; C) Robot-assisted proportional velocity control with associated handle forces; D) Robot-assisted velocity limiting with associated handle forces.

Forces (mN)	FH	FHA	PV	PVA	FS	FSA	VL	VLA
Mean	4.11	3.80	4.20	3.64	3.34	3.22	3.58	3.45
StdDev	0.97	0.59	0.95	0.51	0.54	0.40	0.36	0.33
Max	7.85	6.21	6.93	4.74	4.10	3.59	4.03	3.83
Time (s)	93.03	125.25	62.30	85.98	103.80	96.80	88.67	80.58

Table 5.2: Results of simulated membrane peeling comparing *Freehand* (FH/FHA), conventional cooperative control with linear gain: *Proportional Velocity* (PV/PVA), *Linear Force Scaling Control* (FS/FSA), *Velocity Limiting Control* (VL/VLA).

mode trials, while it decreased the time slightly for the others. The test subject was naturally inclined to "hover" around the discrete audio transition point corresponding to 3.5 mN, which was observed in all cases except Freehand. This was particularly prominent in Force Scaling, where the operator appears to rely on audio cues over haptic feedback (see Figure 5.12C, in 60—80 s time range). In velocity limiting trials, the addition of audio feedback reduced mean input handle forces by 50% without compromising performance. This indicates that the operator consciously attempted to use audio feedback to reduce the forces applied to the sample. The audio feedback trials exhibited a reduction in initial loading where the peeling force increased above the desired level, as shown in the 0–10 second range in Figure 5.12A and B. This difference was less noticeable in the Force Scaling and Force Limiting assistance trials as shown in Figure 5.12C and D.

Proportional Velocity control mode performance benefited from the stability of robot assistance and resulted in a smoother force application, while the range of forces was comparable to Freehand tests. Likewise, additional feedback from audio sensory substitution caused a decrease in large forces, but increased time to complete the task. Force Scaling control mode yielded the best overall performance in terms of mean forces with and without audio, though the average time to completion was longest with the exception of the Freehand with Audio Feedback trials. The longer peel time in that case results from slower and steadier tool motions that generate lower forces applied to the membrane. Velocity Limiting control mode resulted in a very smooth response except for the section that required higher absolute peeling forces at the limited velocity point. This had an effect of contouring "along" a virtual constraint. Due to matching thresholds, audio feedback had very little effect on the performance. Overall Force Scaling control mode with audio feedback provided the most intuitive response, and the greatest force-reducing performance in the simulated membrane peeling task, where the goal is to apply low and steady forces to generate a controlled delamination. However, the handle input forces were high enough (∼2.5 N) to cause fatigue, and had the effect of decreasing precision due to prolonged strain. Increasing the admittance control gain, and lowering the force-scaling factor will lower the handle input forces, but may compromise sensitivity, since force-sensing resolution of a human finger can be as large as 0.5 N [144].

A follow-up experiment involved peeling the ISM without breaking the CAM using the force-sensing micro-forceps as described by Kuru and Balicki et al. in [137]. The following four cases were studied: Freehand peeling with auditory force feedback; Robot-assisted peeling with auditory force feedback; Robot-assisted peeling with force scaling (scale: $\gamma = 300$); Robot-assisted peeling with auditory force feedback and force scaling. The assessment was based on the applied forces during the delaminating motion, i.e. separating ISM from CAM. Sample peeling profiles in Figure 5.13 show that high frequency oscillations in delaminating forces are reduced when the robot is used, with force scaling further smoothing out force variations. Auditory feedback is beneficial for keeping the forces within the "safe" operation zone, i.e. below 7.5 mN.

A similar experiment was repeated by Gonenc and Balicki et al. [145] using the Micron3D instead of the EyeRobot2, and a force-sensing micropick. It assessed the effects of tremor can-

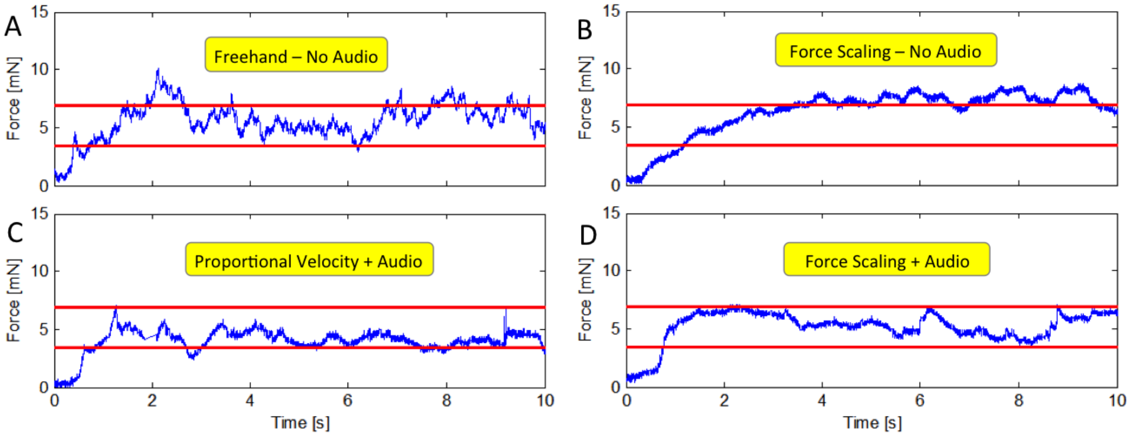


Figure 5.13: Membrane peeling forces on chicken embryo: A) completely unaided; B) robot assisted (force scaling); C) robot assisted (proportional velocity) with auditory force feedback; D) robot assisted force scaling with auditory force feedback

cellation capability of the Micron3D in a Band-Aid membrane phantom peeling task; both with and without audio sensory substitution. The results showed that unaided membrane peeling forces (Micron's active tremor cancellation was disabled) exhibited two main types of variations during the delaminating period: low frequency changes due to varying peeling speed, and high frequency oscillations due to hand tremor. Upon activation of tremor cancellation these oscillations were significantly reduced but the forces above 7 mN were still observed. When auditory sensory substitution was used in freehand mode (Micron3D tremor cancellation disabled), almost all forces were kept below the "danger zone". However, high frequency oscillations appeared both in tip speed and force profiles. The combination of active tremor canceling and auditory sensory substitution produced the best performance, where peeling forces were kept within the safe operation zone, free of high-frequency oscillations. The force frequency analysis (see Figure 5.14) clearly shows that Micron's active tremor attenuation reduces high-frequency force application corresponding to involuntary tool motion from 2 Hz until 20 Hz [146]. The magnitude of forces is greatly reduced in both the absence and presence of auditory force feedback. In this range, the recorded reduction is 60%- 90% with auditory sensory substitution, and 70%-90% without auditory sensory substitution. The prominence of the 10 Hz peak in both figures primarily stems from postural hand tremor as it occurs at a frequency between about 8-10 Hz in normal humans [146]. Below 0.5 Hz, relatively stable hand-eye coordination becomes more dominant [146] which explains the flat region from 0.0 Hz to 0.5 Hz. This is also the range that is affected by the audio feedback, as illustrated in Figure 5.14B by lower force magnitudes for audio assisted trials.

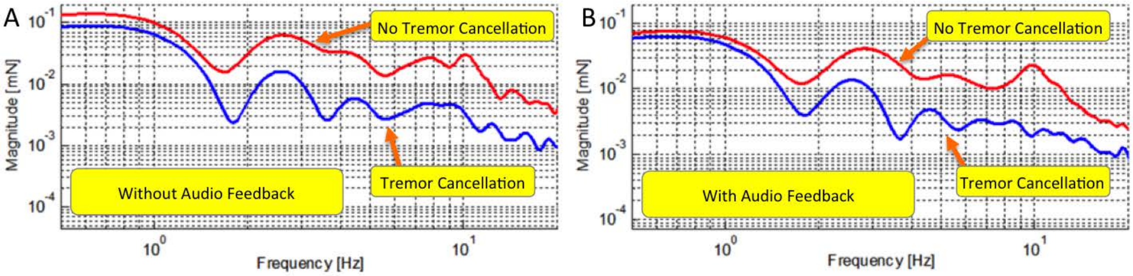


Figure 5.14: Frequency analysis of tool tip forces during phantom membrane peeling with Micron3D: with and without active tremor reduction and A) without auditory feedback and B) with audio feedback.

5.9 Force-guided Peeling Behavior

The following behavior has been developed to aid in challenging retinal membrane peeling procedures to assist the surgeon in manipulating tissue in the direction of least resistance. It extends the force-scaling behavior described in Section 5.6 by limiting maximum forces applied to the tissue and when close to this limit, actively guiding the surgeon towards the direction that generates a lower force. Figure 5.15 shows a typical peeling scenario where the membrane being peeled exhibits anisotropic properties. The method could be broken into two main components as described below.

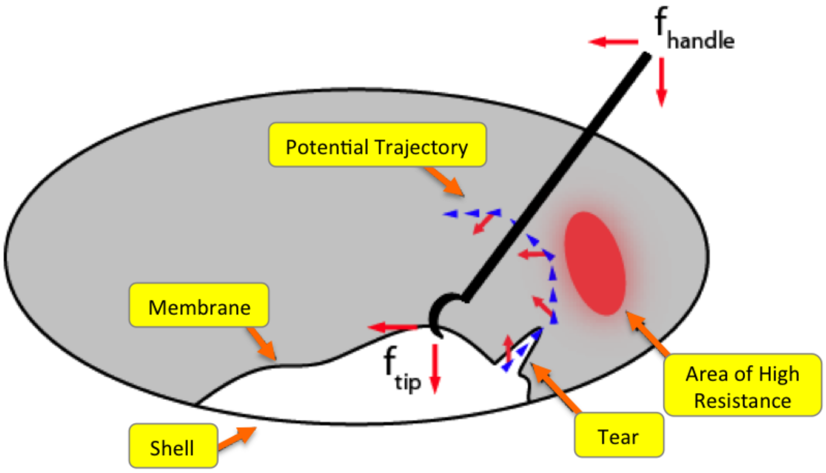


Figure 5.15: Schematic of membrane peeling.

Global Force Limiting: This first layer of control enforces a global limit on the forces applied to the tissue at the robot tool tip. Setting a maximum force f_{max}, the limiting force on each axis would conventionally be defined as

$$\mathbf{f}_{lim} = f_{max}\mathbf{r}_t, \qquad \mathbf{r}_t = \frac{|\mathbf{f}_t|}{\|\mathbf{f}_t\|} \tag{5.7}$$

However, this approach has the disadvantage of halting all motion when the tip force reaches the force limit, i.e., the operator has to back up the robot in order to apply a force in other directions. Distributing the limit with respect to the handle input forces

$$\mathbf{f}_{lim} = f_{max}\mathbf{r}_h, \qquad \mathbf{r}_h = \frac{|\mathbf{f}_h|}{\|\mathbf{f}_h\|} \tag{5.8}$$

gives more freedom to the operator to allow the exploration of alternative directions (i.e. if there exists a maneuver that could generate lower tip forces) even when $\mathbf{f}_t$ is at its limit. Considering the governing control law,

$$\mathbf{v} = k_p\mathbf{f}_h \tag{5.9}$$

the limit is applied as follows:

$$v_{lim} = \mathbf{v}\left(\frac{\mathbf{f}_{lim} - |\mathbf{f}_t|}{l_{lim}}\right) \tag{5.10}$$

Thus the Cartesian velocity is proportionally scaled with respect to the current tip force, where l_{lim} is used to effectively create a spring in order to ensure stability at the limit boundary.

Local Force Minimization: The second control layer guides the operator in order to prevent reaching the limit in the first place. This is achieved by actively augmenting the robot response to lead the operator-controlled tool towards the lower resistance direction. The ratio $\mathbf{r}_t$ is used this time to update the user input in the following fashion

$$\mathbf{v}_{min} = k_p(1 - \mathbf{r}_t s_{min})\mathbf{f}_h \tag{5.11}$$

where s_{min} is the sensitivity of minimization that sets the ratio of the handle force to be locally minimized. Note that if $s_{min} = 0\%$, the operator is not able to override the guided behavior.

For additional safety, the handle input force is monitored to determine if the operator is engaged with the robot handle, e.g., if the operator is not applying any force at the handle (< 0.1 N), the robot minimizes $\mathbf{f}_t$ by “backing up” away from the high force region.

$$\mathbf{v} = K_p\mathbf{f}_t \tag{5.12}$$

This algorithm has been tested on the Band-Aid peeling phantom described in Section 3.7 with a global limit of 7 mN and a minimization sensitivity of 90%. An audio cue was also used to inform the operator when the limit was reached. Figure 5.16A shows the tool tip travel paths, implying the direction of minimum resistance for this phantom. The operator was naturally guided away from the centerline of the tape, following a gradient of force towards a local minimum resistance. Due to mechanical advantage affected by the folding of the tape, this corresponded to peeling at ∼45°to the centerline of the peeling strip.

Algorithm 1 Tip Force-guided cooperative control algorithm.

```
if handle is not engaged then
  do 5.12
else if tip is engaged and tip is opposing handle then
  do 5.11
  if tip is close to limit then
    do 5.10
  end if
else
  do 5.9
end if
```

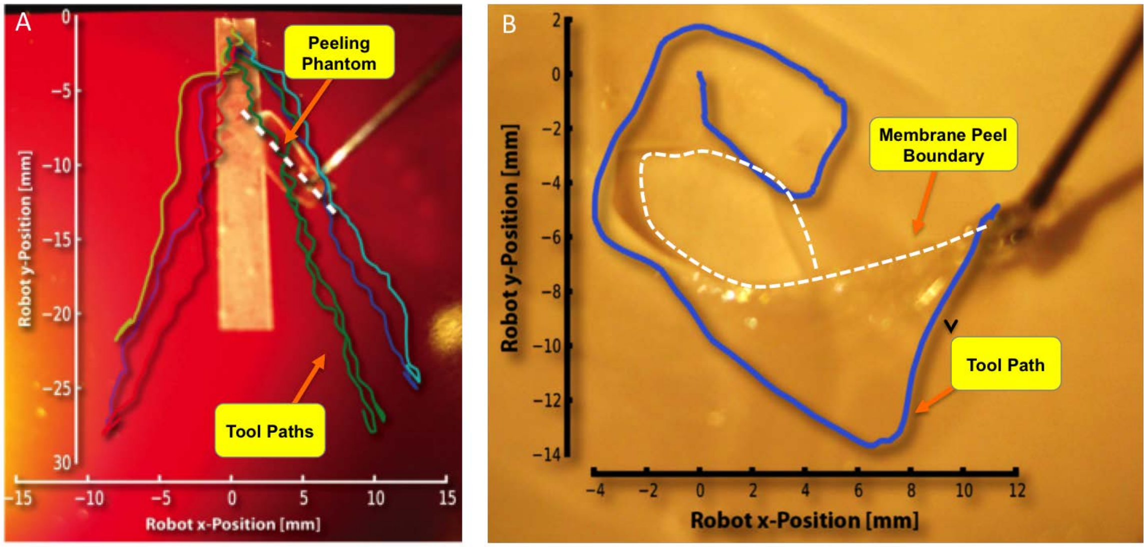


Figure 5.16: Results for the force-guided assistance experiment: A) Tape phantom. B) Peeling Inner Egg Shell Membrane - Diverging and circular robot trajectory overlays. Note that peeled section is half the distance traversed by the robot.

For a more realistic demonstration, this behavior has also been tested on the raw egg membrane peeling phantom, as described in Section 3.3.6. The resulting tool motion indicates a tendency for the operator to peel in spiral trajectories as shown in (Fig. 5.16B). This robot-augmented behavior is consistent with the band-aid trials, but with the added factor of continuously changing tear direction, i.e. tear direction follows the ∼45°direction of force application (Figure 5.15). Effectively, the algorithm was observed to magnify the perception of tip forces lateral to the operator's desired motion. Upon reaching the force limit the operator explored around the boundary in search of points of lower resistance that would enable continuation of peel. This could be achieved smoothly and without requiring the operator to "back up", as the limits on the axes were redistributed based on the operator's application of handle force.

5.10 Cannulation Forces

Another application of force sensing in vitreoretinal surgery is to monitor the process of vein cannulation. Such information can be used to provide real-time, quantitative force feedback aurally, as described in Section 5.3, or haptically, with a robotic assistant, as in the methods proposed by Gracie et al. [88] to communicate or exaggerate boundary transition of force events. These events are generally associated with the three main steps of vein cannulation: 1) insertion of the needle into the blood vessel, indicated by a sharp increase in force; 2) maintaining the needle in the vein, which is represented by a constant force level; and 3) removal of the needle from the vein, characterized first, by increasing force, and then decreasing force when the needle is finally removed. These changes are visible in the force profiles (see Figure 5.18) generated in a preliminarily study[14] conducted by Sun and Balicki et al. [139] that used an early version of a 1 DOF FBG-based force

[14] The experiments were conducted with the help of Brian Hu.

sensor with a straight micro-injection pipette to cannulate a vein in the CAM of a 12-day-old chicken embryo (see Section 3.3.7). The setup is shown in Figure 5.17. The cannulation was performed both with and without robot assistance (EyeRobot1 in basic cooperative control mode) on 100 μm diameter veins. The cannulation force profiles show that there are distinct events that can be extracted from the micro-force sensor readings (see. Figure 5.18). There is little overall difference in freehand vs. robot-assisted force profiles with the exception of the high-frequency oscillations found in the freehand data. The puncture forces (~2 mN) are in the range presented by Ergeneman et al. [147] who used a different experimental setup to measure cannulation force of CAM vessels. They also found that the smallest microneedle size (i.e., 1—2 μm) and bevel needles generated the lowest range of forces (< 1 mN) necessary for vessel puncture. The biological data from the CAM vein cannulation experiments show large force variations for a range of needle diameters and therefore, fitting a predictive model is challenging.

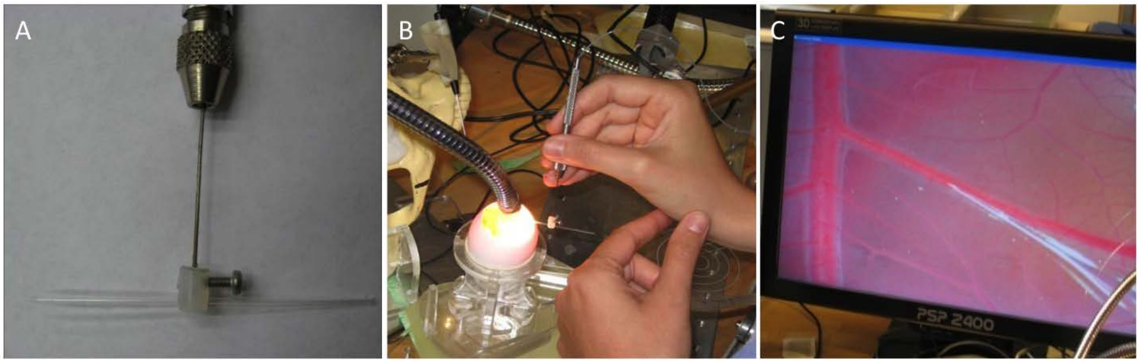


Figure 5.17: A) Force sensor with adapter for straight micro-pipette. B) Cannulation experiment setup with chick embryo. C) Microscope view of the pipette approaching the CAM vessel before cannulation.

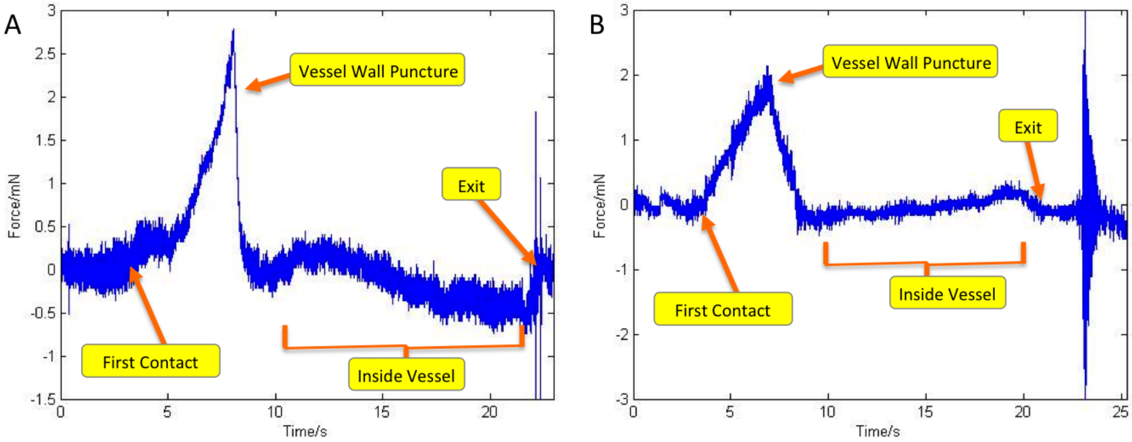
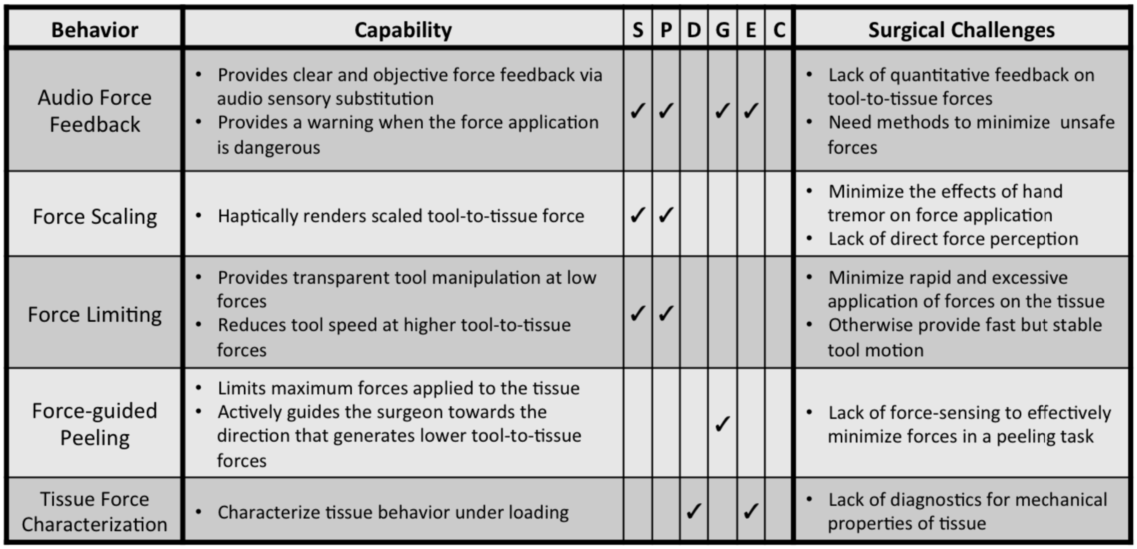


Figure 5.18: A) Force profile in freehand vein cannulation. B) Force profile in robot-assisted vein cannulation. Note: High amplitude and frequency oscillation at the time 23 s is due to accidental perturbation of the pipette outside of the vessel and the large mass of the pipette adapter.



Behavior	Capability	S	P	D	G	E	C	Surgical Challenges
Audio Force Feedback	• Provides clear and objective force feedback via audio sensory substitution • Provides a warning when the force application is dangerous	✓	✓		✓	✓		• Lack of quantitative feedback on tool-to-tissue forces • Need methods to minimize unsafe forces
Force Scaling	• Haptically renders scaled tool-to-tissue force	✓	✓					• Minimize the effects of hand tremor on force application • Lack of direct force perception
Force Limiting	• Provides transparent tool manipulation at low forces • Reduces tool speed at higher tool-to-tissue forces	✓	✓					• Minimize rapid and excessive application of forces on the tissue • Otherwise provide fast but stable tool motion
Force-guided Peeling	• Limits maximum forces applied to the tissue • Actively guides the surgeon towards the direction that generates lower tool-to-tissue forces				✓			• Lack of force-sensing to effectively minimize forces in a peeling task
Tissue Force Characterization	• Characterize tissue behavior under loading			✓		✓		• Lack of diagnostics for mechanical properties of tissue

Figure 5.19: List of force-based Behaviors with specific capabilities they provide, and the corresponding surgical challenges that they address. Behaviors are also characterized by capability types: S-Safety, P-Precision, D-Diagnostics, G-Guidance, E-Education, C-Communication.

5.11 Conclusions and Future Work

This chapter presented Behaviors that rely on intraocular force sensing to provide objective force feedback of normally imperceptible tool-to-tissue forces and also introduced methods for force-based robot assistance to improve precision and safety in delicate tissue manipulation. These Behaviors address surgical challenges with application of milliNewton-level forces which is affected by physiological hand tremor, fatigue, deficient visual and kinesthetic feedback, patient movement, and poor accessibility due to the anatomy of the eye. They are summarized in Figure 5.19.

The addition of force sensing to the surgical toolset has been of great interest to vitreoretinal surgeons and has the potential to significantly alter the current methods in manipulating delicate retinal tissue, e.g., in procedures such as membrane peeling. The associated Behaviors have been shown to be beneficial in prototypical vitreoretinal surgery tasks. The following points describe the lessons learned during the device and Behavior development, the resulting requirements for next design iteration, and proposed future work:

- The current generation of the force sensor instruments is able to measure, with high sensitivity, the forces in two axis, transverse to the tool shaft. However, to cover a variety of surgical approaches and force application directions, a microsurgical forceps and other standard instrumentation needs to also include the axial force measurement functionality. Recently, He and Iordachita have presented a prototype design that provides this functionality [148]. In the future, higher dimensional force measurements could improve overall measurement precision and account for possible torques.

- In contrast to general surgery, the addition of force sensing and feedback to ophthalmic procedures is not one of regaining lost feedback, but rather it is the addition of a sense that was not available before. Presenting this new information intuitively is challenging. One method is audio sensory substitution as presented in this chapter. It provides instant, objective and intuitive force feedback to the surgeon. Despite this, audio may compete with other operating room equipment, which uses audio for alarms or basic feedback on the state of the device. Continuous audio feedback may be disruptive or overwhelming, but occasional and unique sounding aural signals during critical moments can be very effective. Additionally, the alarm-like force feedback sounds may startle or cause anxiety for the patient who is often awake during the operation. A simple solution is to provide a simple wireless earphones that only permits only the surgeon to hear the sounds.

- Preliminary user feedback has indicated that auditory sensory substitution can be very useful, especially when combined with cooperative robot control. This concept has potential to be a very useful training tool in providing immediate and objective tissue manipulation force information, which can be correlated with other cues. As surgical training programs continue to adapt more stringent curricula hours, training with force-sensing tools that provide quantitative feedback could help new residents acquire microsurgical skills more quickly. The studies shown that force-Audio Sensory Substitution can effectively improve peeling performance when participants are asked to perform an unfamiliar task. Anatomical and pathological variances in every procedure result in a new peeling experience for the surgeon. This illustrates the importance of being able to adapt to new tasks and the potential benefit of force-sensing tools.

- Besides tactile and audio force feedback, other sensory substitution methods, such as visual overlays presented in Chapter 7, may also provide sufficient force representation. Anecdotal experience indicates that perceptions sensitivity (latency and precision) vary with the different methods and a comparison should be investigated in the future.

- Further investigation of the force characteristics of membrane peeling will be necessary to understand the full implications of the forces that are associated with both normal and damaging maneuvers. The results of raw egg ISM peeling tested the sensitivity of the micro-force sensor (force range of 0.2–1.5 mN). The peeling of the ISM from CAM in a chicken embryo (1.3—4.1 mN) was compared to a deliberate CAM injury (5.1-9.2 mN) during the same task to simulate retina tears. These were higher in a follow up experiment (6.07 – 34.65 mN) that used a forceps tool. After careful analysis to make sure the tool is functioning the same, the conclusion is that the egg tissue is highly variable. This is one reason why artificial membrane peeling phantoms were developed - to provide a consistent testing platform for multi-subject tests. With further understanding of the forces required, both to successfully complete a surgical task and those that induce tissue injury, a library of safe and dangerous forces for any intraocular surgical maneuver can be developed that would serve as a framework for developing enabling systems that preemptively warn the surgeon of imminent tissue injury and allow the surgeon to minimize the risk of inducing a complication. For example, retinal tears are a rare but a recognized complication during ERM peeling. Because the forces of typical retinal manipulation and that of creating a retinal tear are both below tactile sensation, quantitative information communicated to the surgeon during this maneuver could prevent a retinal tear during ERM peeling.

- Intraocular cannulation forces are difficult to measure to to lack of adequate force-sensing instrumentation. One approach, proposed by Tameesh et al. [149], uses a flexible thin tube that is handled inside the eye by the surgeon using a forceps. The surgeon then guides tube into the vessel to perform the cannualtion. The force-sensing forceps described in this chapter are compatible with this approach and can also be used with a robotic assistant in force-based Behaviors such as force scaling, etc., to generate a more controlled injection.

- Robot stiffness reduces user perceived level of the forces between tool shaft and sclera and these excessive forces on the sclera can also cause serious complications. Sclera force information can provide vital feedback in robot assisted eye surgery. To restore the perception of these forces a force-sensing instrument was recently developed that has the capability to measure the forces both at the tool tip and along the tool shaft as described by He and Balicki at al. [124]. A more advanced version that is in development uses three FBG sensor sections in each of the three optical fibers embedded in the tool shaft to extract the location and amplitude of the sclera force, which is then used in robot control to establish a force-based dynamic RCM.
- Future in vivo experiments should examine the manipulation forces in various vitreoretinal microsurgical tasks, explore the properties of biological structures, and collect data to determine the safety-critical force threshold for manipulating the tissues in the eye.

5.12 Recapitulation of contributions

Micro-Force Sensor (Section 5.1) The first retinal surgery compatible instrument capable of measuring tissue manipulation forces inside the eye was created. The FBG-based sensor is located at the distal end of the surgical instrument (inside the eye) and can measure sub-mN forces, while avoiding the attenuation from the sclerotomy interaction that affected previous designs. Hook and forceps instruments were also developed. The instrument can be used for basic science to collect manipulation forces, as well as in various force feedback systems, and can be applied in any surgical application that benefits from fine force sensing embedded in long tubular shafts. **Credit**: Developed by Iulian Iordachita, Russell Taylor, Jin Kang, and Xingchi He with significant input from the *eyeBRP* team members. Marcin Balicki assisted with the design and performed initial device testing. Software developed by Marcin Balicki.

Biological Membrane Peeling Force Experiments (Section 5.2) A set of experiments which investigated tissue manipulation forces related to membrane peeling was undertaken. Raw egg and chicken embryo models were used to test the sensitivity of the force-sensing instruments. These were compared to a first-reported in vivo rabbit study to measure retinal tear and hyaloid detachment forces. The results show that the peeling forces ($\sim$ 6 mN) are in the range that is difficult or impossible to perceive directly by humans. This also indicates that the chicken egg and embryo are good surrogates for development and testing of force-sensing instruments. However, artificial peeling models were developed to minimize variability and difficulty of preparation. The results are also used in development of active assistance, which incorporates force-sensing to increase safety of membrane peeling. **Credit**: The membrane peeling experiments were conducted by Sarah Sunshine and Xingchi He. Marcin Balicki developed the force collection software and the experimental setup, and assisted with the experiment design and analysis.

Force-Based Membrane Peeling (Sections 5.6, 5.7, 5.9) In addition to the implementation of the existing force scaling cooperative control algorithm, two new robot control methods were developed to assist in surgical membrane peeling. All the algorithms use the tissue interaction forces and modulate robot velocity. A force-limiting algorithm limits the maximum robot velocity above a force threshold, but increases maneuverability when low tip forces are present. The force-guided peeling extends the force-scaling algorithm by limiting maximum forces applied to the tissue and when close to this limit, actively guiding the surgeon towards the direction that generates a lower force. These can be applied in cases where a cooperatively controlled robot is used with a force-sensing instrument to aid in tissue manipulation tasks. **Credit**: Marcin Balicki and Ali Uneri.

Audio Force Feedback (Section 5.3) The first force-audio sensory substitution for retinal surgery applications was developed and thoroughly tested. It uses three distinct audio clips that are turned on and off based on three force-level zones which represent force-operating ranges selected for their relevance in typical vitreoretinal tasks. Initial freehand membrane peeling experiments showed that audio force feedback helped to reduce mean and large forces, but significantly increased task completion time. A follow-up multi-subject study showed the same results, regardless of surgical experience. This force-audio sensory substitution method can also be used in other micro-surgical applications that involve tissue manipulation forces below human perception. **Credit**: Concept development, system design, and initial experiments by Marcin Balicki with assistance from Mark Finkelstein. Multi-subject experiment was conducted by Nathan Cutler.

Comparison of Feedback Methods in Membrane Peeling (Section 5.8) A study was undertaken to assess the effectiveness of EyeRobot assistance and audio sensory substitution in decreasing mean and maximum peeling forces (on a Band-aid phantom) while minimizing the time taken to complete the peeling task. The results indicate that audio force feedback is the most effective method at lowering the forces, but can be supplemented for further improvement with force scaling or force-velocity-limiting robot control. The test-subject using standard cooperative control and freehand peeling did not perform well without audio force feedback in this study. Confirming results were seen in a follow up experiment with a force-sensing forceps tool used to peel ISM from CAM. A similar experiment was repeated using Micron3D on a Band-Aid membrane phantom peeling task. The combination of active tremor canceling and auditory sensory substitution produced the best performance. This experimental setup can be used to test new micro-force control methods for membrane peeling. The audio feedback is very effective at communicating discrete force levels and can also be applied in other surgical tasks. **Credit**: EyeRobot-based experiment was conducted by Marcin Balicki and Ali Uneri. The chicken embryo experiments were conducted by Ismail Kuru. The Micron experiment was performed by Berk Gonenc and Marcin Balicki.

Tool Compliance Correction (Section 5.4) A method for correcting EyeRobot kinematics by including an estimate of the tip location of the flexible tool shaft was presented. The tool tip position can deviate as much as 1 mm when manipulating tissue. The solution maps the tool deflection to the micro-force sensed at the tip. This correction is important in applications where precise tip position of a flexible shaft instrument is required, such as force-based robot control, or precise tissue characterization. **Credit**: Marcin Balicki and Ali Uneri.

Tissue Force Characterization (Section 5.5) An experiment was conducted to assess the potential of micro-force sensing in characterizing tissue reaction forces during retina-like tissue tearing, using a chicken embryo's CAM as a model. By servoing the robot based on the feedback from a micro-force-sensing tool, a constant force was applied to the embryo's membrane in stages. The characteristic response is a similar pattern to those seen in fibrous tissue tearing. This information can be used in designing force-based assistance to minimize retinal tears, or in other delicate tissue micro-manipulation procedures. **Credit**: Marcin Balicki and Ali Uneri.

Cannulation Forces (Section 4.10) A study was conducted to determine the range of forces during cannulation of 100 μm chicken embryo CAM vessels. The study used an FBG force sensor held by the EyeRobot to assist the operator with the cannulation. A later study by a different group [147] found similar force measurements (∼2 mN) but used a different force sensing scheme. The findings indicate that cannulation forces are below human tactile perception and micro-force sensing may be used in assistive systems to improve the safety and efficacy of such difficult surgical procedures, not only in eye surgery but also in other micro-lumen injection tasks. **Credit**: Marcin Balicki and Brian Hu.

Chapter 6

OCT Behaviors

The previous chapter (5) focused on methods to extend human sensing ability in the realm of application and feedback of micro-forces. Another major challenge faced by vitreoretinal surgeons is the difficulty in locating and clearly visualizing surgical targets during the operation, and the negative effects of physiological hand tremor on safe execution of high-risk surgical maneuvers near the retina. Surgical outcomes (both successes and failures) are further limited by technical hurdles that cannot be overcome by current instrumentation. A typical vitreoretinal instrument has a simple and singular mechanical functionality, e.g., forceps for grasping tissue, and does not provide any physiological or even basic interpretive information that is surgically relevant, e.g., the distance of the instrument from the retinal surface, or the depth of instrument penetration into the retina. The following chapter presents Behaviors that rely on Optical Coherence Tomography (OCT) imaging to provide diagnostic information via real-time intraocular imaging of the retina, as well as surgical safety and intervention assistance capabilities that leverage the integrated nature of the *eyeSAW* ecosystem[1].

6.1 OCT Technology

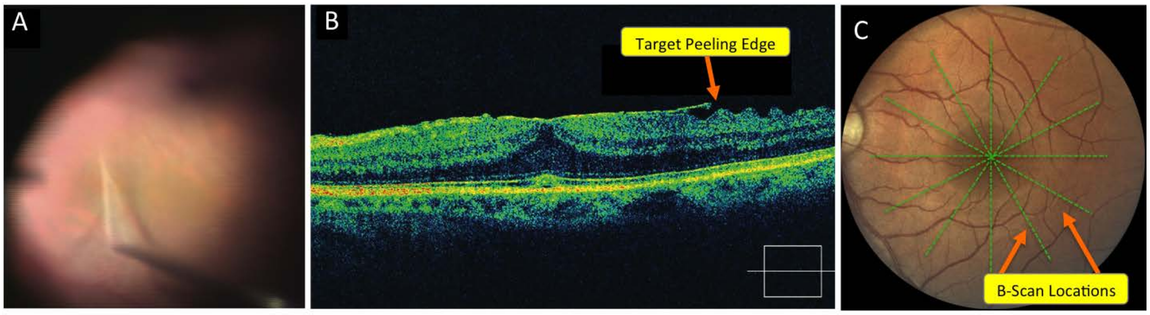


Figure 6.1: A) Microscope view of ERM peeling and B) corresponding preoperative OCT B-Scan of the retina [29]. C) Locations of typical preoperative B-Scans on a fundus image.

OCT provides very high resolution (micron-scale) images of anatomical structures within the tissue. Within Ophthalmology, OCT systems typically perform diagnostic imaging through specialized optics to provide 2D cross-sectional images ("B-Scan") of the retina. Figure 6.1B shows a typical OCT B-Scan of an epiretinal membrane (ERM). In OCT, ERMs are recognized as thin, highly reflective bands anterior to the retina. A potential dissection plane between the ERM and the retina is clearly visible in the preoperative scan (pointed to by the arrow in Figure 6.1B), but is invisible to the surgeon through an operating microscope, even with very high magnification. Figure 6.1A shows the membrane peeling process where the already delaminated ERM shows sufficient contrast to be detected visually through the microscope.

Until recently, OCT was purely a preoperative imaging technology used for diagnosis and treatment planning, and in a few cases, for optical biopsy and image-guided laser surgery

[1] Dr. Kang's optics group provided all the OCT expertise and built most of the OCT hardware described in this chapter. Early help was from Jae Ho Han, Yi Yang and Kang Zhang. The latest incarnations were developed by Xuan Liu.

[29, 150–152]. Work by Fleming et al. [153] explored the registration of preoperative OCT images to intraoperative microscope images to aid in identifying ERM edges for initiating ERM removal. However, ERMs can grow and further distort retinal architecture. It is therefore unclear whether preoperative images would provide a useful guide if the interval between the preoperative image acquisition and surgery allows for advancement of the ERM. Direct imaging of the ERM relative to the surgeons' instruments would be very useful, either as a replacement for, or supplement to, the preoperative OCT images. In addition, direct imaging of the local anatomy relative to the instruments can be used to provide feedback on tool-to-tissue distances and real-time updates during tissue dissection.

These considerations have led to the development of sub-millimeter diameter microsurgical instruments incorporating OCT sensing capability for directly sensing tissue planes beyond the instrument tip [154]. This approach relies on the use of common path Fourier domain OCT (CP-OCT) sensing through an optical fiber built into the instrument shaft. Fourier domain OCT works on the principle that coherent light interferes with itself after it is reflected and refracted. This interference encodes (in Fourier domain) the distances from the reference plane to the optically opaque layers of the imaged sample. The spectral image can be collected by a spectrometer in a single exposure, and quickly processed into a whole axial depth scan (A-Scan), see Figure 6.2B.

Intraocular CP-OCT is simple, robust, and affordable [155], and also permits an extremely compact tool design. Alternative diagnostic approaches include an intraocular hand-held probe by Han et al. [156] that has a large 21 Ga. diameter and is rather expensive due to its delicate mechanical scanning system. Ehlers et al. [157] reported the use of a surgical microscope with integrated volumetric OCT imaging capability for intraoperative diagnostics. Their system is prohibitively slow; requires ideal optical quality of the cornea and lens, and can be affected by the gel that is

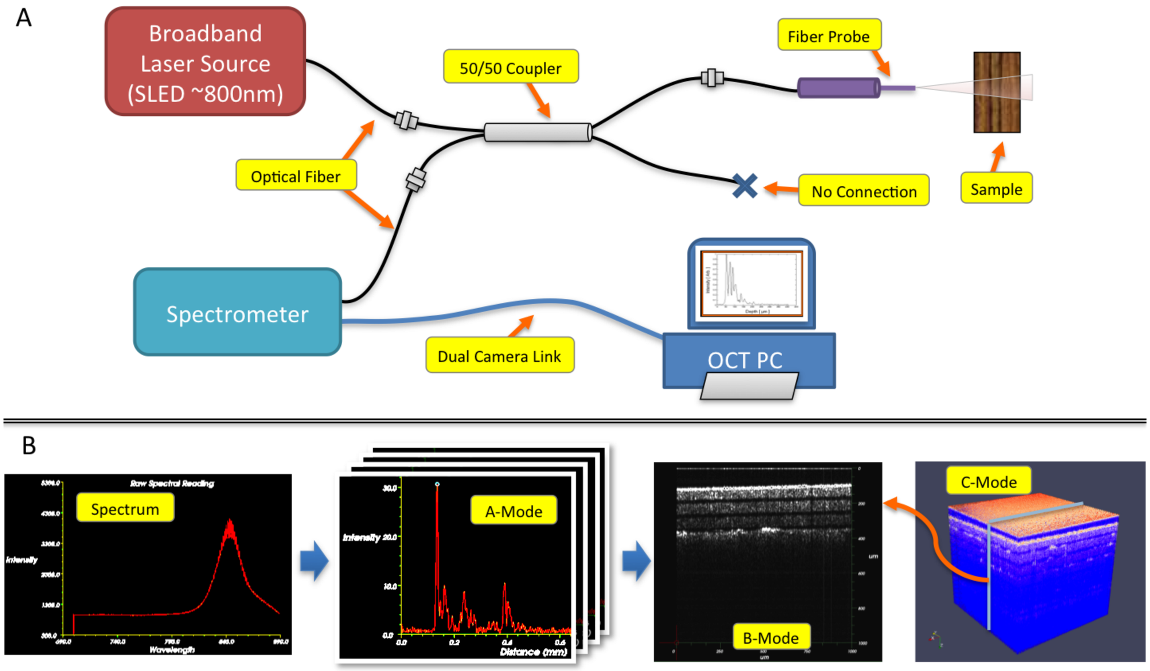


Figure 6.2: OCT technology: A) Schematic of the common path Fourier domain OCT System. B) From left to right: raw OCT spectrum, A-mode single axis scan (A-Scan), B-Mode scan is a planar 2D slice where structural opacity is represented by grayscale values (B-Scan), C-Mode scan builds a volumetric (3D) image (C-Scan) and can be constructed from multiple B-Scans.

often used to keep the cornea moist during the operation. Both of these approaches lack a unified display, requiring the surgeon to look away from the surgical field to examine the OCT image.

The “A-Mode” sensing capability of a common path (single fiber) OCT can either be used with conventional freehand instruments or can be combined with a robotic microsurgery platform. With added instrument tracking, A-Scans can be organized into 2D planar images (B-Scans) and also volumetric images (C-Scans). Without tracking, sequential A-Scans can be combined into basic 2D images called H-Scans. When integrated with a robotic assistant, the OCT instrument may be scanned to produce images of the local anatomy or integrated in various ways to control the robot. This integration allows for a system that can enforce safety barriers that prevent collision with the retina, is be able to maintain a constant distance from the retina during scanning, and promotes accurate targeting of anatomic features within the retina.

6.2 OCT System

Several single fiber Fourier domain OCT systems were designed specifically for intraocular imaging [155, 158–160] presented here. The general CP-OCT design (see Figure 6.2A) comprises a single fiber probe, a 50/50 (2×2) fiber-optic coupler, custom-built high-speed spectrometer, and a low coherence, fiber-coupled, superluminescent diode (SLED) light source providing 50 to 100 nm bandwidth with the central wavelength of about 800 nm. The output of this broadband source is coupled into one input port of the fiber-optic coupler and exits through one of the output ports that is connected to the single mode fiber probe, described in the next section. The reference light from the probe tip, and the light backscattered by the sample, couples back into the fiber probe, where it interferes, and exits the fiber-optic coupler into a spectrometer to be digitized. The spectrometer consists of a collimator, a diffraction grating, an achromatic doublet lens and a pixel line scan CCD detector. The detectors work at very fast line scan rates (e.g., 28 kHz) and have a minimum integration time of 1 μs. The extremely short exposure integration is critical for CP-OCT applications where the OCT probe is moving at fast and varying speeds. Relative motion of the imaged sample during a single exposure can cause phase fluctuations which lead to blurring and averaging of the modulation depth and greatly affect the OCT image quality. By keeping the CCD exposure time as short as 5–40 μs in water (1–5 μs in air), the interferograms are acquired with minimum phase fluctuation to obtain a stable and strong OCT signal[2]. Figure 6.3A shows a portable version of the OCT system that includes a rack-mounted multi-core computer at the base.

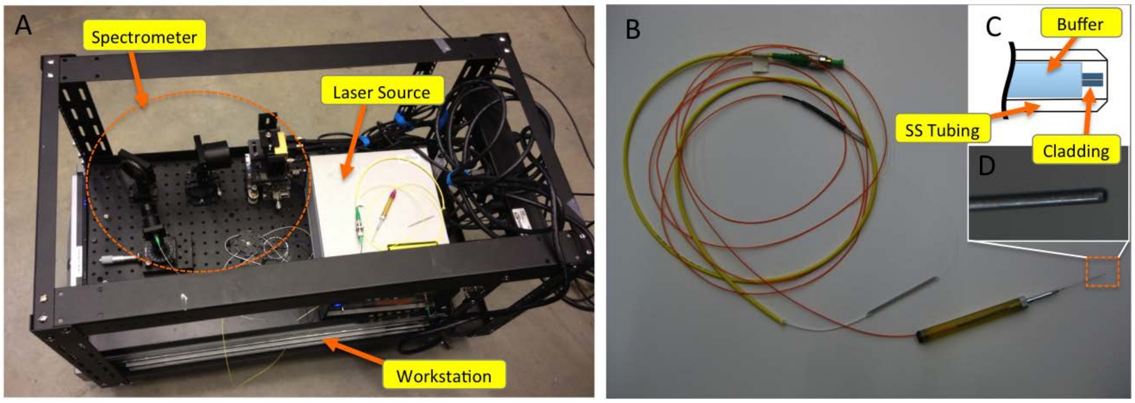


Figure 6.3: A) Portable OCT System: rack-mounted computer at the base, laser source, and the spectrometer. B) Cross-section schematic of OCT probe assembly showing optical fiber stripped of plastic cladding and bonded into the lumen of the beveled stainless steel tubing. The tip of the fiber is recessed slightly (up to 250 μm) for protection from collisions. C) Detail view of the probe tip.

Vitreoretinal procedures require the image resolution to be high enough to image surgically relevant structures that are routinely less than 10 μm. The axial resolution of the CP-OCT varies with the particular configuration of the spectrometer and the bandwidth of the light source. The high-resolution configuration has about 2.3 μm (full width at half maximum) inside the aqueous

[2] Integration time is a function of the laser power, the optical quality of the probe, and the reference signal. Motion artifacts in FD-OCT with long integration times result in A-Scan signal strength degradation.

eye environment (3.0 μm in air) with theoretical maximum imaging depth of 1.2 mm in tissue (1.6 mm in air). The most recent configuration [160] features a theoretical axial resolution of 4.8 μm and a practical imaging range of approximately ~2 mm in water. The OCT configuration is typically a trade-off between the axial resolution and theoretical imaging depth of the system. Later configurations forgo the higher resolution to attain deeper imaging depth so the probe can be further from the surface of the retina, which is considered safer. The typical imaging depth is 1–1.5 mm in tissue and somewhat less in the wax sample materials (see Section 3.3.4).

6.2.1 OCT Instruments

The probe arm of the CP-OCT is a single mode optical fiber (SM800-5.6-125, Thorlabs Inc.) which has a polymer buffer coating diameter of 245 μm, a cladding diameter of 125 μm and core diameter of 5 μm. The numerical aperture is ~0.14. This miniature form factor allows integration with vitreoretinal surgical instruments which are typically long (30 mm) and tubular (0.5 mm dia.). To provide a strong Fresnel reference reflection light, the fiber tip is cleaved at a right angle. The resulting OCT beam divergence in water or tissue is ~12 degrees and ~16 degrees in air. This translates to an effective 20–30 μm imaging width of each A-Scan at 0.5 – 1.5 mm imaging depth [159]. It should be noted that the quality of the OCT images is also a function of the angle between the sample being scanned and the probe axis. In controlled bench-top experiments this is not an issue, but in the relatively small rabbit eye, as well as with sclerotomy constraints, the probe-surface angle can be greater than 20 degrees from the normal which causes significant degradation in the sharpness of the OCT images. This can be addressed with a sapphire ball-lens probe design [161], which provides a more focused "sweet spot" at a given distance from the probe, or an angled viewing probe [162] that is tailored for the procedure to image peripheral regions of the retina.

The basic imaging OCT instrument used for many bench-top experiments is built with the probe described above stripped of the polymer buffer (~100 μm distal section) and bonded with cyanoacrylate adhesive inside a protective 40 mm section of stainless-steel tubing with 250 μm ID, and 500 μm OD. The fiber is recessed to protect the brittle glass cladding and fiber core from accidental collisions with metal instruments or tissue. The edges of the tubing are beveled (see Figure 6.3C-D). This "lens-free" design leads to a very simple, inexpensive, and robust probe. The tubing is then bonded to a handle that compatible with the EyeRobot's quick-release tool adapter as shown in Figure 6.3B.

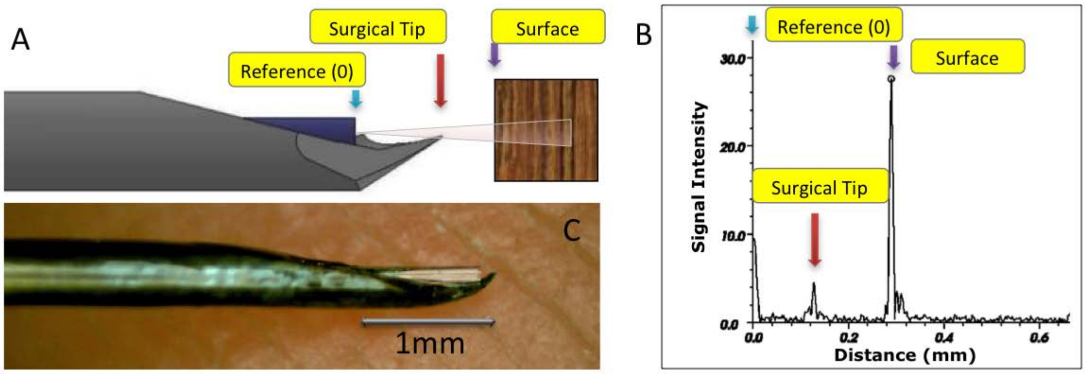


Figure 6.4: A) Side view computer rendering of microsurgical pick with integrated fiber optic OCT probe. B) An A-Scan showing peaks corresponding to the tip of the pick and the surface of the sample. C) Physical prototype.

Vitreoretinal micro-picks picks are commonly used to incise taut surface membranes and to engage and elevate membrane edges for ERM peeling. The simplicity of this basic tool permits the surgeon to quickly craft a sharp-edged pick from an appropriately sized surgical needle (ranging from 25–20 Ga (0.5 mm to 0.9 mm diameter)) by bending the beveled tip with a needle holder. The surgeons routinely use this method to create such picks. Balicki at al. [154] adapted this method to build an OCT-integrated pick to enable simultaneous A-mode imaging and tissue manipulation.

The prototype OCT micro-picks were constructed from 25 Ga, 38 mm long surgical needles with 0.25 mm ID and 0.5 mm OD. The beveled point was bent approximately 200–300 μm from the tip so that the tip intersected with the central axis of the needle lumen. A cleaved optical fiber with ∼2 mm of the plastic jacket stripped to glass cladding was inserted through the lumen, and bonded in place approximately 135 μm from the tip. It is shown in Figure 6.4. One of the main advantages of this probe configuration is that the instrument position is inherently registered to OCT by the fact that the surgical tip is visible in the OCT image. Figure 6.4C shows a sample A-Scan with the zero reference point located at the edge of the fiber probe while the tip of the micro-pick and surface of a sample are simultaneously present in the image. The relative distance of the tool to the tissue is calculated from the distance between the two peaks[3].

6.2.2 Software

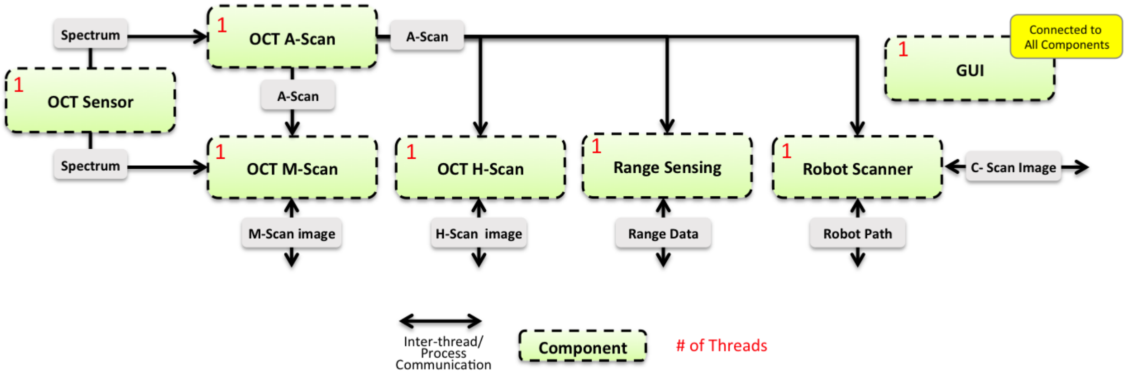


Figure 6.5: Schematic of the OCT software module architecture showing the main components, associated threads, and connections. All of the modules are running asynchronously.

[3] For the initial prototype the location of the probe relative to the pick's tip was mechanically calibrated. Other methods could involve a robot pressing the pick into a deformable material (e.g., wax) and then scanning the resulting cavity with the robot to recreate the geometry of the pick in 3D.

The most basic function of the OCT software module is acquiring the raw spectral data from the frame grabber and processing it to form A-Scans (1024 pixels). A high level schematic of a common OCT software architecture is shown in Figure 6.5. The OCT Sensor component acquires a single spectral image continuously from the frame grabber storing it in a triple buffer. The A-Scan module fetches the latest raw spectrum and immediately converts it into an A-Scan, and both are stored in a large circular buffer with fast, non-blocking read access. These are then accessed by various signal processing components working in parallel to provide reduced OCT data with low latency for display, feedback and synchronization with probe positions for scanning application. Currently, the update rate is $\sim$ 4.5 kHz which is limited by the frame grabber capture method (single frame at a time) and bus speeds.

Other components in the same process access these A-Scans for range sensing, and composing 2D and 3D OCT images. The output of range sensing algorithm provides physical range data that can be used by robotic devices or communicated directly to the user. The average latency is one “frame” ($\sim$ 0.3 ms), which is more than sufficient for most applications presented here that require sensor feedback. The application also contains the graphical user interface for engineering visualizations, testing and calibration. The underlying software framework (*cisst-SAW*) facilitates controlling the operational modes as well as retrieving processed data from these components. The GUI, M-Scan, and H-scan components generate small compressed 2D images (e.g., A-Scan plot, H-Scan, M-Scan preview) at 20–30 Hz that can be accessed by any component in the system using standard component interfaces. The creation of one of these images is followed by a broadcast of an event that distributes the image content to the listening components in the system. In addition, the application includes robot assisted scanning behaviors that are discussed in this chapter. VTK was used for rapid development of XY-plots, and render volumetric OCT images [163].

The OCT application generates more than 200 MB of data per second, which is difficult to archive in real-time due to limited write speeds of the standard hard disks. To circumvent this limitation, each component is responsible for saving data that is deemed relevant for a particular task. It is up to the system developer to ensure that sufficient resources are available during the execution of the experiment. For example, a 10-second M-Scan image is generated on demand when the OCT probe is swept over a section of the retina. The OCT M-Scan component creates an M-Scan object by collecting the required A-Scans and building 2-D image. Once the scan is complete, the M-Scan image is sent out to other components (e.g., the Visual M-Scan, see section 7.10) as an event, and another thread is spawned automatically to archive the full resolution M-Scan object. This includes the following data: a reference spectrum (1024 x 1 px), spectrums (2024 x 45000 px), raw A-Scans (1024 x 45000 px), and a compressed thumbnail (640 x 480 px) and full resolution images (1024 x 45000 px). This data could be as large 0.5 GB for the 10 second scan. The same approach is implemented for robot assisted scanning.

A special experimental version of the application was developed to provide bulk scanning capability that allows the frame grabber to collect data at 30 kHz. This approach relies on the frame grabber to buffer data into a large frame (up to $\sim$ 8 kSamples), which, once full, is transfered immediately to the main memory. The batched nature of the spectrum transfer does not lend itself to real-time feedback (e.g., range sensing) and has only been used for short but high speed robot-assisted volumetric scans, e.g., 0.5 seconds, where high bandwidth scanning is required to capture sufficiently dense images.

6.3 Range Sensing

Range sensing using OCT involves the tracking of the tissue surface visible in the OCT A-Scans relative to the distal end of the surgical instrument[4]. For range sensing to be effective feedback for surgeons, especially when used in robot control algorithms, it needs to be accurate ($\sim 10\mu$m), stable, and have high-frequency and low-latency updates (e.g., 100 Hz and 20 ms respectively). The simplest approach is tracking the brightest layer of the sample represented by a maximum peak in an A-Scan, as shown in Figures 6.2B and 6.4B. This works surprisingly well in bench-top experiments on artificial phantoms, in air, and, especially, when the OCT probe is normal to the surface. In these cases the highest intensity area of the A-scan represents the surface of the sample. A median filter (window size = 40) on the peak estimate is implemented to remove the occasional floating debris artifacts, and to prevent the loss of surface signal due to high frequency, high amplitude vibrations, or other environmental noise.

To track the surface of biological samples and more complex artificial phantoms, a more advanced algorithm has been developed. The layers found in the OCT images of natural heterogeneous tissue, especially in an aqueous environment, have more variable structures, exhibit an unknown number of visible layers, and an unpredictable order of layer (peak) intensities (e.g., the surface layer is not always the most prominent) as shown in Figure 6.7A.

The surface tracking algorithm shown in Figure 6.6 works on the most recent set of A-Scans to provide robust and immediate results. The first step involves pixel-wise averaging (moving average) across a window ($w = 15$) of most recent A-Scans[5]. This minimizes tissue-related speckle and device noise present in each pixel, varying from A-Scan to A-Scan. The resulting averaged A-Scan (1024 pixels) is smoothed by convolving the A-Scan pixels with a Gaussian kernel (window

[4] For this demonstration, the distal end of the tool is considered to be the same as the OCT reference (0 μm).

[5] The A-Scan capture and processing rate is 4.5 kHz with minimal latency.

= 20, σ = 3)[6]. The smoothed A-Scan is thresholded to remove the baseline noise as well as the DC component found near the reference (position 0 μm) or the tool tip visible in the A-Scan. [7]. The thresholding values are calculated at the time of the re-biasing calibration step, which is typically performed immediately before the scanning session with the OCT instrument located in the scanning environment, e.g., inside the eye, and far (∼5 mm) from any tissue.

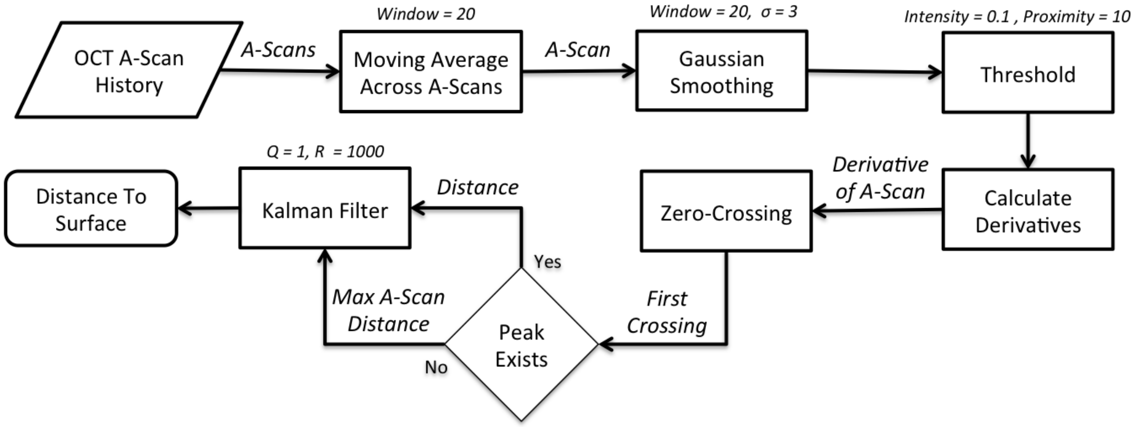


Figure 6.6: Flowchart of the OCT surface tracking algorithm.

The next step identifies the A-Scan peaks corresponding to significant tissue layers by a positive-to-negative zero-crossing of the A-Scan derivative calculated with a finite forward difference method. The first peak found is then used as input into an adaptive (Kalman) filter, otherwise, if no peak is found, the maximum A-Scan depth is used[8]. The 1 DOF Kalman filter [164] is used to smooth the noisy numerical derivative data, and produces a smooth transition between possibly discontinuous surfaces. This can be used as a stable input for robot control with OCT feedback described in later sections.

[6] In some experiments an alternative smoothing method is used (Savitzky-Golay filter) to preserve peak locations, amplitudes and width.

[7] The thresholding step should also consider the instrument tip, if it is visible in the A-Scan.

[8] When no peaks are detected the algorithm returns maximum distance. The CP-OCT configuration used in the scanning mediums, such as air and water, does not detect any samples at maximum theoretical pixel depth.

When tuning the averaging window size w parameter, the scanning frequency and robot control loop should be considered to produce a stable and correct response, with lowest possible lag, e.g., for 4.5 kHz sampling the window should be 10–20. The parameters for the Kalman filter are the measurement noise variance ($R = 1000$) and process noise variance ($Q = 1$) which were tuned on data from previous experiments. The other parameters are empirically tuned, and sometimes adjusted during runtime. This algorithm runs on its own thread, effectively in real-time, with minimal latency outside of the effects of the filtering.

To demonstrate the surface tracking algorithm, the three most commonly used phantoms were scanned in an aqueous environment. The hand-held OCT instrument was swept across the sample surface at rate of $\sim$0.5 mm/s for 5 seconds, generating 25000 A-Scans. The scan of the tape phantom described in Section 3.3.4 is shown in Figure 6.7A. The random axial displacement of the surface in the OCT image is mostly attributed to natural hand tremor. The quality of the image is a function of the OCT hardware configuration and the angle of incidence between the OCT beam and the sample[9]. A one-second section of the scan (5000 A-Scans) is processed and shown in Figure 6.7B. The blue line represents the automatic segmentation results of the first opaque layer, while the red line shows the filtered estimate of the surface. Notice that the tracker "snaps" to the floating debris that is above the surface. For a typical tool motion instability that is generated by hand tremor, the filter lag is about 20 ms with 5 px displacement error, which corresponds to $\sim$20 μm. Figure 6.7C-D shows similar tracking results for scans of the silicone ERM model in the Eye Phantom.

The same algorithm is applied to in vivo OCT data collected in a rabbit experiment (see Figure 6.8A). The surface is not as sharp as the artificial samples, due to the large angle of incidence

[9] The OCT images are collected with the probe at 15–30 degrees to the surface normal. This causes significant blurring of the sample details, e.g., surface, but is representative of realistic conditions in which this system is used.

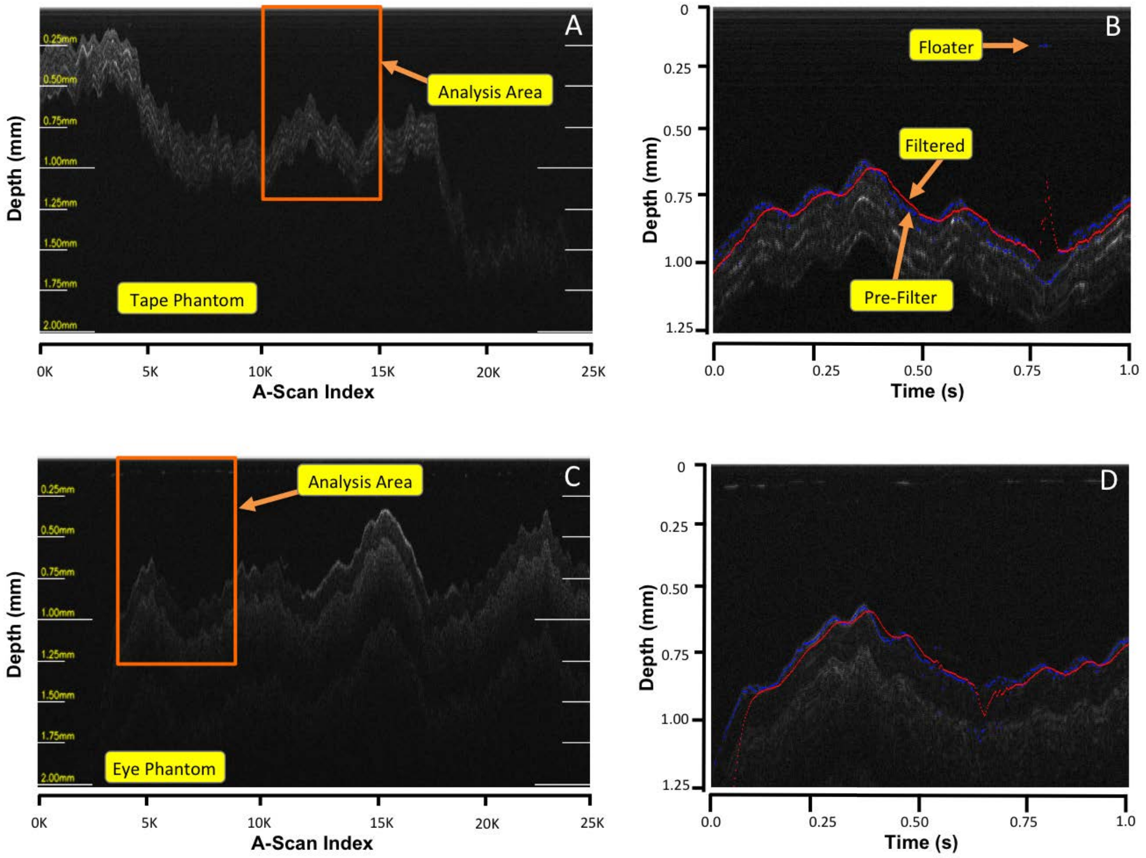


Figure 6.7: A) Five-second freehand OCT scan of the tape phantom. B) One-second section of the scan with corresponding surface tracking results depicted by the red line. The blue line represents the surface segmentation before the Kalman filter is applied. C) Five-second freehand scan of the latex retina insert in the Eye Phantom. The top layer is silicone ERM, the two lower layers are latex coats. D) Selected section showing surface tracking results.

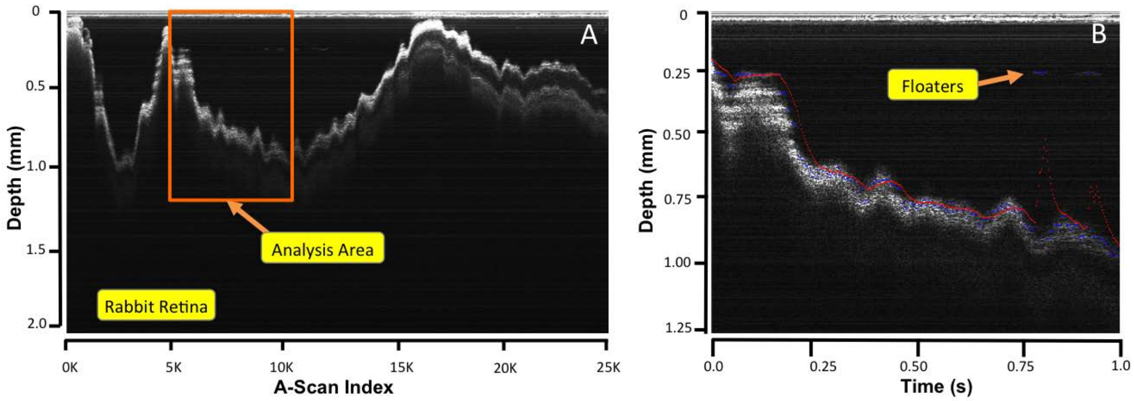


Figure 6.8: A) Five-second freehand scan of the rabbit retina. B) One-second section of the scan with corresponding surface tracking results depicted by the red line. The blue line represents the surface segmentation before Kalman filter is applied.

(probe axis to the surface normal) of the OCT beam, which is common when imaging such a small eye with an endoscopic probe. Despite this, the tracker is able to extract the surface with ∼8 px displacement error on average, which corresponds to $\sim 32\ \mu$m. However, as in Figure 6.7B, the algorithm considers small "floaters" as a surface layer, causing a significant jump at time = 0.8 s. Depending on the application, rapidly detecting any material that may come into the OCT field of view may be a desirable behavior.

By adjusting the Kalman filter gains it is possible to further smooth out the response and minimize the effects of such "floaters" as shown in Figure 6.9A. As with any low-pass filter, this causes large delay. An effective alternative is to apply a median filter instead of a Kalman filter. The results are shown in Figure 6.9B. This tends to remove the artifacts caused by small debris, while minimizing signal latency. A middle-ground approach applies the median filter first, followed by the normally tuned Kalman filter. The results are shown in Figure 6.9C. This can be adjusted based on the context. For example, in the case of range sensing as input in robot feedback, it may be desirable to have a less filtered approach since the robot motion is inherently very slow, and natural displacement of the retina is expected to be rather smooth. The over-filtered range sensing is applicable in Behaviors that do not require exact and real-time or extremely precise feedback such as audio sensory substitution. An alternative is to offload additional filtering onto those Behaviors, since they are more "aware" of the context in which they are operating and may have information regarding the type of sample that is being imagined.

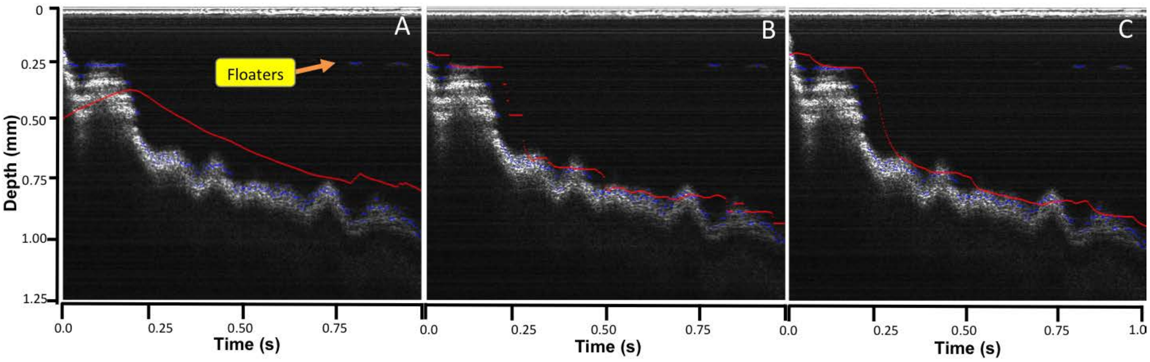


Figure 6.9: A) Highly filtered surface tracking of a rabbit retina (Kalman filter: $R = 1000$, $Q = 0.001$). B) Filtered surface tracking with median filter ($w = 200$). C) Surface tracking with median filter ($w = 100$) followed by a Kalman filter with parameters ($R = 1000$, $Q = 1$).

The above algorithm is sufficient for bench-top experiments and is currently limited by the resolution and sharpness provided by the available OCT system configuration and the high incidence angle of the OCT beam with the tissue. Further improvements in the endoscopic OCT imaging technologies (e.g., narrower OCT beam, side viewing or focused probes producing more contrasting layers) will provide more precise data and result in more precise range sensing.

It should be noted that other approaches to retinal layer segmentation in OCT [165, 166], especially the surface layer [167], are not designed to function in real time and consider the whole 2D or 3D OCT image. These preoperative images have a very high contrast and are A-Scan aligned, a luxury stemming from application specific hardware optimization and a stable (preoperative) OCT imaging scenario. Furthermore, the algorithm presented here can be extended to track other layers in the OCT images in real-time by classifying the layer correspondence of each segmented peak using Kernel Density Estimation [168], following an assignment of each peak to a particular Kalman filter corresponding to that layer. These layer "features" can be used to align the consecutive A-Scans to provide a more realistic B-Scan image that considers the typical topology of the retinal layers. This is an alternative to the Normalized Cross Correlation (NCC) approach described by Huang et al. [160][10].

6.4 Audio OCT Feedback Behaviors

The audio sensory substitution methods developed for force feedback described in Section 5.3 have been adopted for representing OCT range data aurally. One major addition is a mode that interprets the range distance values in reverse fashion compared to force. I.e., high values are considered safe (e.g., >1000 μm), while small values (<300 μm) indicate that the tool is in close proximity to a structure and an alarm should be sounded. In freehand manipulation, the tool-to-tissue range changes continuously and sometimes very rapidly due to hand tremor, hence the preferred feedback scheme (presented in Figure 6.10B) offers two distinct regions with different audio "beeps". The "warning" zone modulates the playback frequency of the audio clip so that a higher frequency corresponds to closer proximity and more intense warning. The "alarm" zone is reached after that threshold is surpassed, at which point a high-pitched, constant, high-frequency feedback is generated. The ranges greater than d_2 produce no audio feedback. The typical values for these transitions are $d_1 = 300$ μm, $d_2 = 1000$ μm.

The scheme in Figure 6.10A provides more discrete feedback by adding an additional

[10] Another approach is to use surface tracking for rough A-Scan alignment and then use the NCC method to fine tune it. This can also allow for incorporating larger weights in NCC to the bottom layers rather than the widely-varying surface.

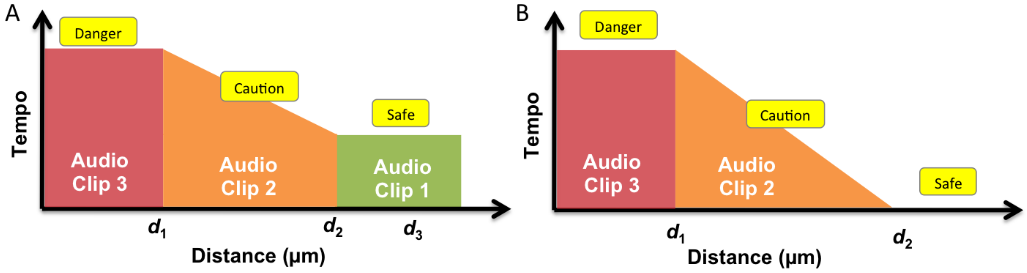


Figure 6.10: Surface proximity (OCT range) to audio feedback: A) mapping algorithm with four regions: $0 - d_1$ high tempo, $d_1 - d_2$ proportionally increasing tempo with the decrease in distance, and $d_2 - d_3$ slow tempo, and silence when the distance was above$> f_3$ high tempo; B) mapping algorithm with three regions: $0 - d_1$ high tempo, $d_1 - d_2$ proportionally decreasing tempo, and $> d_2$ silence.

feedback region, and can be used in cases where a particular "safe" standoff distance is required. The surgeon can use the discrete audio transitions at the boundary of d_2 as feedback to servo at the preset distance[11]. The typical values for the boundaries are $d_1 = 300$ μm, $d_2 = 500$ μm, and $d_3 = 1000\mu$m. These parameters are surgeon and task specific, and are predesigned before the operation or testing and tweaked in real-time as necessary.

This sensory substitution was found to be very helpful during OCT-based Behavior development, serving as an immediate feedback method during scanning tests and providing general safety information in cases of possible collisions with the artificial samples, and in vivo retina imaging experiments. The audio feedback was supplemented with visual OCT feedback, such as overlays described in Chapter 7, and has been used extensively to supplement the haptic virtual fixtures described below. It provides a richer, more intuitive communication, especially in cases where the OCT-based haptic feedback might be surprising or counter-intuitive, e.g., when the robot does not allow the surgeon to move the tool close to the retina because of close proximity. The distance-to-surface audio feedback reminds the surgeon that the instrument is dangerously close to the surface

[11] The d_3 transition of the first feedback method can be used also but having the continuous audio feedback provides more immediate and bounded information.

and the robot is actively preventing the collision.

6.5 OCT – EyeRobot Behaviors

The feedback from the OCT instruments can be communicated to the surgeon with audio feedback as shown above, or it can be incorporated into the behaviors of a robotic assistant as a real-time haptic feedback modality. This can improve robot precision by providing end-point sensing to correct for deflections in the kinematic chain, much like the correction of deflection compliance using the micro-force sensor in Section 5.4. I.e., by measuring the environmental properties (e.g., distance) relative to the "business end" of the instrument, the deflections of the rest of the instrument, and of the robot, are not as significant in determining the location of the instrument that is interacting with the tissue. This is especially important in automatic or semi-automatic servoing to a target (e.g., cannulation, or membrane peeling) where the robot control algorithm needs to know the exact location of the instrument relative to the location of the target.

The use of OCT to adapt to the varying local environments has first been shown by Balicki et al. in [154], where a combination of OCT and robotic assistance is demonstrated in the context of three sample tasks: enforcement of safety barriers; "tracking" to maintain a constant distance from a surface; and accurate placement of the probe on targets identified in a scanned OCT image. These are summarized in the following sections.

6.5.1 Safety Barrier Behavior

One of the first capabilities developed by combining OCT and robot functionality is the Safety Barrier Behavior, where the system prevents the instrument from approaching the target surface closer than a pre-specified threshold distance (κ). This leverages the OCT Range Sensing module to provide the distance (d_s) from the instrument tip to the surface of the tissue. The instrument is held by the robot, and controlled directly by the surgeon through cooperative control. The commanded velocity along the tool axis, $v_{\hat{Z}_t}$, generated by the standard force control law (see (4.2)) is altered based on the following criteria:

$$v_{\hat{Z}_t} = \begin{cases} \max\left(-K_B\epsilon, K_p f_{\hat{Z}_t}\right) & \text{, if } (\epsilon < 0) \\ \min\left(-K_A\epsilon, K_p f_{\hat{Z}_t}\right) & \text{, if } (\epsilon > 0) \wedge (f_{Z_t} < 0) \\ K_p f_{\hat{Z}_t} & \text{, otherwise} \end{cases} \tag{6.1}$$

where $\epsilon = d_s - \kappa$. If $\epsilon < 0$ the probe is too close to the surface (in danger zone B, see Figure 6.11C), and the robot will either automatically move away from the surface, along $\hat{Z}_t$ with gain $-K_B$, or the operator is already commanding the robot away from the surface ($f_{\hat{Z}_t} > 0$) with velocity greater than $-K_B\epsilon$. If $\epsilon > 0$, the probe is out of the danger zone. However, when approaching the surface ($f_{\hat{Z}_t} < 0$) the maximum velocity towards the surface is capped as a function of the distance to it, ϵ, so that the robot will at maximum move with velocity of $-K_A\epsilon$. Since this virtual fixture is only active along the tool axis, the operator is still able to use cooperative control ($v = K_p f_h$, see (4.2)) to translate the probe along the surface, or angle it relative to the surface.

The above constraint is effectively a proportional controller with gains (K_A, K_B, and K_P) tuned empirically, so the maximum expected velocities provide smooth but rapid response when the operator is driving the robot towards the surface, and also when the surface approaches the instrument while the robot is not being force-controlled (but the control pedal is pressed down). This is especially important in cases where the imaged surface is moving unexpectedly, such as during involuntary patient motion during surgery.

The following experiment demonstrates this Behavior in a bench-top setup shown in Figure 6.11A–B. The imaging instrument is mounted on the linear stages of a modified EyeRobot2. The

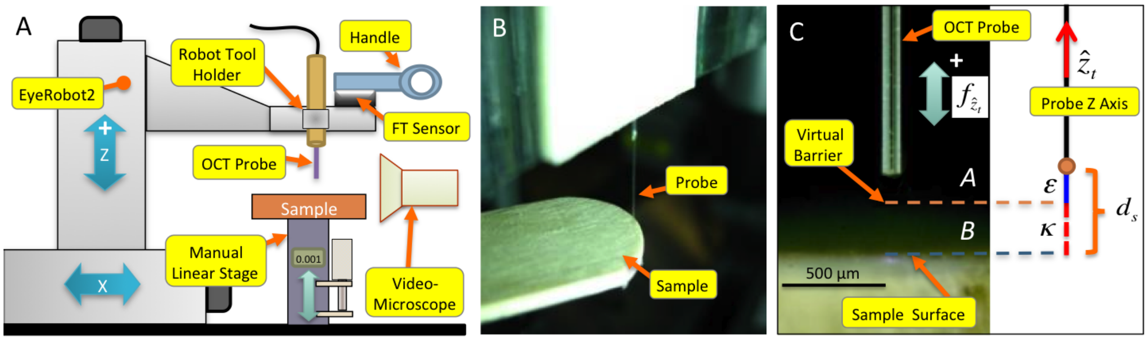


Figure 6.11: A) Experimental setup for OCT-Based robot Behavior tests. B) Photo of OCT probe and the tape phantom. C) High magnification (90×) view from the inspection microscope of the bare OCT probe near the surface of the phantom. Includes a schematic of control zones for the Safety Barrier Behavior.

.

positioning resolution is verified to be 1 μm and measured repeatability is about $\pm 1\mu$m for the range of motion required for this experiment (< 2 mm). The robot moved freely within the 1D workspace to comply with forces exerted by the operator on the control handle, with the exception of the forbidden boundary sensed via the OCT range sensing module. This "virtual wall" was reached when the tip of the probe was located $\kappa = 150$ μm from the tape phantom surface described in Section 3.3.4. A bare optical fiber was used as a probe. An inspection video microscope (E-ZVU/V15, Omano Inc.) is positioned to simultaneously image the side view of the sample and probe at 90× magnification.

Five trials were performed with different robot velocity limits: 100, 200, 300, 400, 500 μm/sec. Relatively low velocities were chosen based on surgical preference when operating close to the retina, and the limited movement in the eye during surgery due to the constraints resulting from the insertion of tools through the sclera and the effects of anesthetics.

The inspection microscope videos are hand-segmented and analyzed. The results for the safety barrier task are shown in Figure 6.12A. The observed over-shoot into the "unsafe" zone was

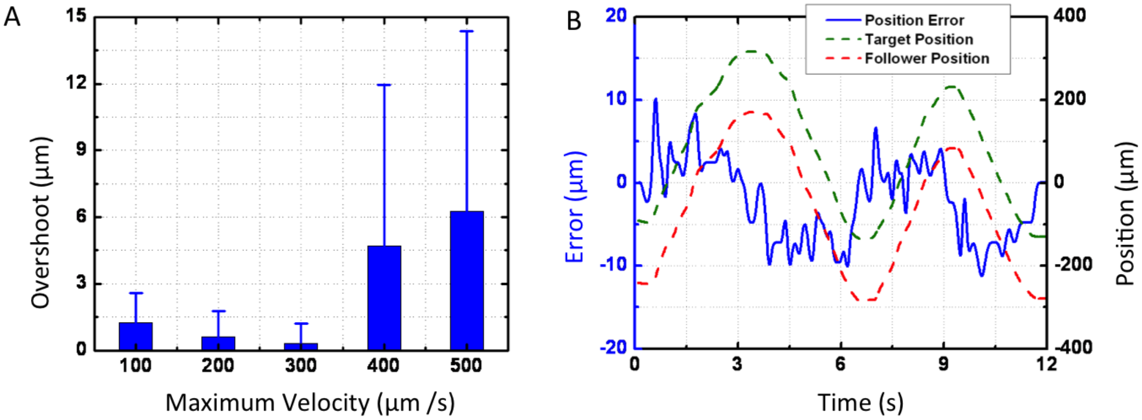


Figure 6.12: A) Results for Safety Barrier Behavior test: overshoot error vs. maximum allowable velocity. B) Results for Surface Tracking Behavior with 150 μm standoff goal.

negligible for robot speeds up to ~300 μm/sec and still quite acceptable for speeds up to ~ 500 μm/sec. Note that these errors also include the effects of robot deflections generated by the operator applying significant force (up to 5N) on the handle. Despite this, the major contributors to this larger error at higher speeds is the low-performance of the OCT system [155] which, due to the high integration time, and low serial sampling rate of the CCD elements, fails to capture reliable OCT images at high velocities. The subsequent generations of the OCT system and more dynamic robotic manipulators address these issues and allow for faster interaction and response.

6.5.2 Surface Tracking Behavior

There exists a system capability that benefits from the ability of the robotic assistant to keep the instruments at a constant distance from the surface. One application addresses the limitation of the simple OCT probe that lacks dynamic focusing, where a constant distance to the tissue provides continually sharp images. Additionally, such functionality can dynamically rectify the undesirable axial displacement between A-Scans caused by hand tremor, which results in jagged images of the retina that are difficult to interpret. This can also be addressed in software, as shown by Liu et al. [169], by vertically aligning similar adjacent A-Scans, but is only applicable to small variations and does not address the sharpness issue.

In the surface tracking task tape phantom samples are mounted on a separate, manually-actuated micrometer stage placed beneath the probe, as shown in Figure 6.11A. The goal was for the robotic control to maintain a constant OCT-reported distance of 150 μm from the sample surface, which was moved up and down manually. One intention for the surface tracking is to correct for retinal motion due to respiratory function, hence the sinusoidal frequency of the input motion is around 0.2 Hz, with a magnitude of $\sim$ 600 μm that encompasses the expected ranges of retinal motion.

The Surface Tracking Behavior algorithm modifies the basic cooperative control ($v = K_p f_h$, see (4.2)) by ignoring the force input along the tool axis ($f_{\hat{Z}_t}$) and servoing along this axis to the offset distance (κ) from the surface located under the tool. A proportional gain K_0 was empirically tuned and used in the following velocity controller:

$$v_{\hat{Z}_t} = -K_0 \epsilon \tag{6.2}$$

where $\epsilon = d_s - \kappa$, and d_s is the distance from the tool reference to the surface provided by the OCT Range sensing module.

The results of the dynamic response of the system when tracking sinusoidal surface motion in 1 DOF are shown in Figure 6.12B. The robot was able to keep the tool tip within about 10 μm of the desired 150 μm standoff from the target surface while the surface moved at about 200 μm/sec.

6.5.3 Servo-To-Target Behavior

The Servo-To-Target Behavior explores the capabilities of the system to enable user identification of a subsurface target in a scanned image followed by automated placement of the instrument tip on the target. For this targeting task, the robot was instructed to scan the OCT-pick in 2 μm increments laterally across the sample surface of the Wax Phantom (see Section 3.3.4), creating a B-Mode image. The evolving B-mode image was displayed continuously to the user, who could use a mouse at any time to click on the target within the image. The robot would then interrupt its scan, and automatically move back to a position over the identified target. It would next use the Surface Tracking Behavior to servo the pick tip to the surface, within 20 μm, followed by a maneuver that creates a puncture of the membrane by moving the tool towards the target. The last step was performed in an open loop fashion based on the location of the pick tip and the depth of the designated target in the same A-Scan, collected right before the maneuver began. The probe was then withdrawn by the robot and a second automatic B-Scan was taken to observe the created defect in the sample surface.

The results of the targeting task are illustrated in Figure 6.13. The subsurface bubble ($\sim 150\mu$m dia.) with 30–50 μm thick membrane and the user specified incision location are shown in Figure 6.13A. The post-intervention B-Scan with an overlay depicting the approximate orientation and location of the instrument tip at the target is shown in the Figure 6.13B. The defect clearly resembles the geometry of the tip, as well as showing good alignment with the planned incision position.

This demonstration assumes that the sample does not move relative to the base of the robot. This contrived scenario is unlikely in a clinical setting but can be addressed with external tracking methods that locate the relative position of the instruments in the surgical microscope,

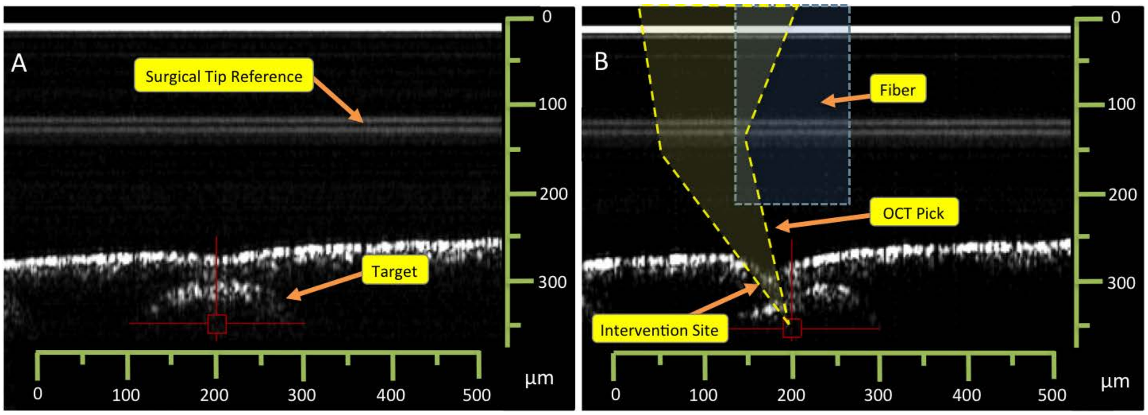


Figure 6.13: OCT Servo-To-Target Behavior test: A) Initial B-Scan of the wax phantom with user input indicating the target bubble. Only a section of the B-Scan is shown. B) Post-intervention B-Scan of the same region showing the indentation caused by the instrument. The overlay shows the estimated position of the instrument tip at target depth.

for making the necessary depth corrections, for producing B-mode images from the tracked tool positions, and for generating registered overlaid displays on the microscope. This is discussed in the following Chapter.

6.5.4 Automatic B-Scan Behavior

Intraocular B-Scans can be automated using an EyeRobot with virtual fixtures. This enables a linear B-scan that simplifies mental registration between the content in the OCT and the location of the scan, similar to the preoperative scans in Figure 6.1B-C. The other benefit is the ability to control the distance between the probe and the surface of the retina. Besides safety and stable motion provided by the robot, the constant offset improves the OCT image quality, and is especially applicable in cases where the OCT probe suffers from significant signal decay with increased imaging depth, e.g 1000 μm in the case of the lens-free probe [12]. The imaging distance

[12] In a probe without a focusing lens at the sample arm, signal decay is due to the light beam exiting the fiber tip diverging significantly as it propagates further.

can also be optimized for a particular probe such as the ball-lens probe, where a specific distance, not necessarily close to the sample, will provide the best imaging quality.

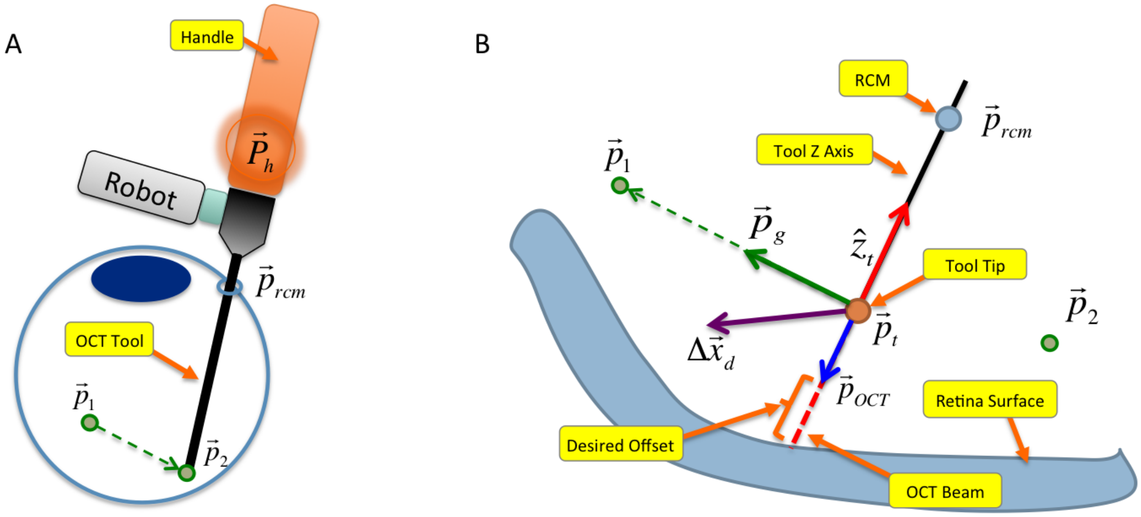


Figure 6.14: Automatic B-Scan Behavior: A) Surgeon uses the instrument to define the B-Scan line by specifying the end (P_1) and start (P_2) locations B) The B-Scan is executed automatically from P_2 towards P_1 while the motion component along the tool $\hat{z}$ is servoed to keep a constant distance from the retina based on the OCT range signal. RCM constraint is also enabled. B) Schematic for constructing the B-Scan behavior.

The scanning protocol requires the surgeon to first define the path that will be used to generate the automatic B-Scan. The surgeon guides the robot so the instrument tip (P_t) is located near the end of the scan (P_1) and then another point (P_2) near the beginning of the scan as shown in Figure 6.14A. The surgeon then presses a pedal that triggers the robot to perform an automatic scan from P_2 to P_1 while observing the RCM constraint and simultaneously servoing based on OCT range feedback to keep a constant proximity to the surface, similarly to the method in Section 6.5.2. This effectively sweeps the probe tip in an arc that lies in the plane created by the three points P_{rcm}, P_1, P_2 while adjusting the radius of the arc in real-time.

The following objective for constrained control optimization combines the desired motion towards the goal P_1 and the motion to achieve the constant distance from the surface of the retina. In order to control the lateral scanning speed, the incremental motion P_g in the direction of P_1 is calculated by taking the minimum of $||\frac{(P_1-P_t)}{||P_1-P_t||}S_{max}||$ and $P_1 - P_t$, where S_{max} is the maximum scanning speed × control loop period. At each iteration of the robot control, the Behavior requests the latest distance to the surface (d_s) from the OCT range sensing module. Using d_s, the desired incremental motion (P_{oct}) of the tool tip (P_t) along the main tool axis is calculated as follows:

$$P_{oct} = -(d_s - \kappa)\hat{z} \tag{6.3}$$

where κ is the desired offset constant[13]. The desired incremental motion (Δx_d) is created by combining the two motion vectors and subtracting the component of P_g that lies along the $\hat{z}$:

$$\Delta x_d = P_g + P_{oct} - P_{g_z} \tag{6.4}$$

The above is incorporated into the objective function with a robot Jacobian resolved at the tip ($J_t(q)$), of which only the first three rows are used[14]:

$$\underset{\Delta q}{\arg\min} \quad ||J_t(q) \cdot \Delta q - \Delta x_d|| \tag{6.5}$$

The scanning is stopped when the tool tip (P_t) is sufficiently close (e.g., 100 μm) to the line created by P_{rcm} and P_1. The objective can be further adjusted to provide faster or slower response of the OCT-based surface servoing by directly adjusting the length of P_{rcm}[15], or by adding a constraint to the optimization formulation. Besides the RCM constraint, additional constraints are added so that the combined velocity and accelerations are reasonably safe, and the motion is stable in all scenarios, see Sections 4.3.4 and 4.4. Figure 6.15 shows a comparison of a scan performed

[13] The desired offset constant, κ, should be sufficiently large to prevent collisions with the retinal structures. This is typically not an issue in the naturally smooth surfaces of the retina.

[14] It is combined with another task objective that minimizes all the joint incremental motions.

[15] The objective could also include damping terms.

in freehand fashion vs one performed automatically by the robot using the above algorithm. The automatic B-scan has less vertical distortion with the exception of occasional deviation due to floating debris.

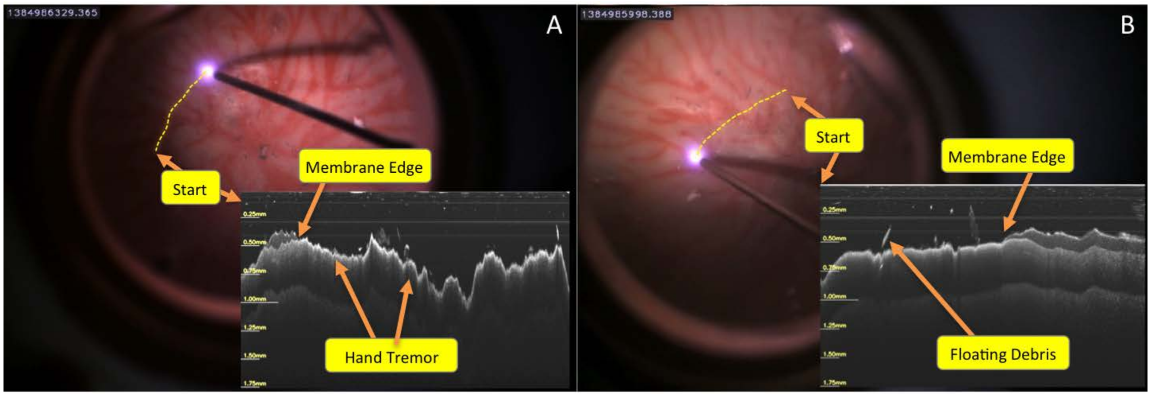


Figure 6.15: Example OCT scans in the JHU Eye Phantom with OCT-only visible silicone membranes: A) Hand-held OCT scan shows significant axial image distortion due to hand tremor. B) Robot-assisted automatic B-Scan has much less distortion, although the floating debris does effect the servoing performance.

In more advanced implementations, the scan location could be defined by centering the tool over an area of interest, where the robot would perform a semi-automatic volumetric scan centered on this point, with a constant distance to the tissue. Another approach is to define the B-Scan path directly in the visualization sub-system by pointing at the desired locations on the fundus or on a real-time feed of the retina. This can also provide additional closed-loop robot control via visual tracking of the surgical instrument and the retina.

6.6 Robot-Assisted Scanning

A-Scans on their own can be used for depth ranging and to a lesser extent for diagnostic imaging. By tracking the position of a probe as it moves across tissue, B-mode and C-mode scans can be generated. This can be performed manually with an external probe tracking device (Ren at al. [170]) or partially through lateral velocity compensation, from feedback obtained from the correlation between consecutive A-scans (Ahmad et al. [171] and Huang et al. [160]). However, the accuracy and repeatability of robotic devices provide a clear advantage for acquisition of such scans. Additionally, as discussed earlier, with the inclusion of an actuated device, the distance of the probe from the tissue can be automatically controlled, which is beneficial both for safety and for image quality.

Special spiral-scanning devices that rely on the resonance of a single fiber inside a catheter have been developed for endoscopic use [172–174]. Helical scans have been used for imaging luminal organs such as the esophagus [175]. These devices tend to be too large for intraocular use. The intraocular hand-held scanning probe by Han et al. [156] has a large 21 Ga. diameter and can only perform B-Scans. These systems tend to be very specialized, only providing the scanning function. By integrating a robotic assistant with the OCT, a number of volumetric scanning Behaviors have been developed and demonstrated here.

6.6.1 OCT Scanning with EyeRobot

The initial integration of the EyeRobot and OCT focused on real-time safety virtual fixtures and planar B-Scans. This section presents the feasibility of volumetric scanning with the same system configuration in a bench-top setting where the robot is run in open-loop fashion relative to the sample which is mounted rigidly relative to the robot base. C-Scans (raster scans) were performed using the setup described in Section 6.5.1, where the EyeRobot2 has a bare fiber probe attached to the CP-OCT system [158] (see Section 6.2). The robot was commanded to move in a planar (X-Y) 1.5 mm x 1.5 mm grid pattern visiting an incremental goal position every 12.5

μm (see Figure 6.16A). Each A-Scan's pixels were transformed into the robot base frame by the corresponding robot-probe transformation.

The resulting raw volumetric OCT intensity point cloud is shown in Figure 6.16A. The robot was placed near the surface of the tape phantom in Figure 6.16A. The phantom has two human hairs ($\sim$ 100 μm diameter) placed in the center of four layers of Scotch tape to simulate a vessel in the retina. The intensity of the raw data points are transformed into a structured grid using inverse distance weighted interpolation [176], thresholded and ray traced using the VTK software library [163] to produce a volumetric image such as the one in Figure 6.16C. The two cross-sectional slices generated from the C-Scan volume are shown in Figure 6.16. The distal one shows a single void of the two OCT-opaque hairs overlapping. The proximal one contains two distinct voids that are formed by the hairs diverging[16]. They are measured to be $\sim$ 100μm in width matching the physical size of the hair.

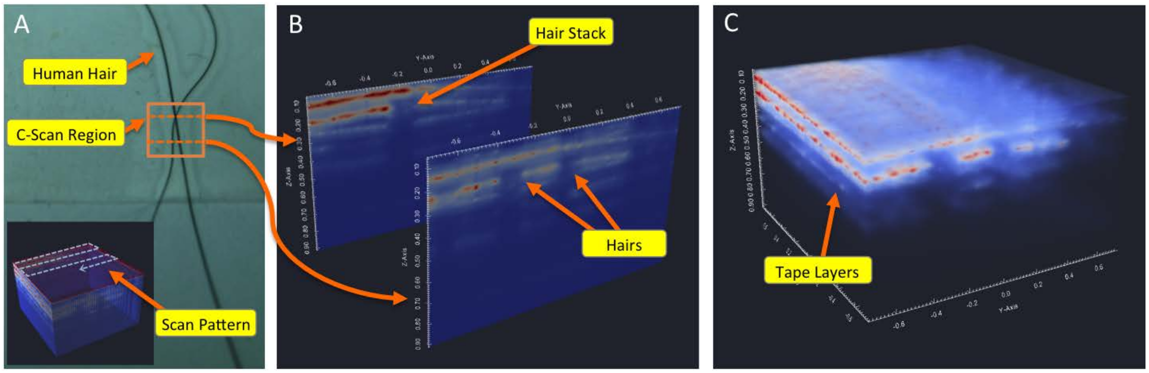


Figure 6.16: Automatic OCT Scanning with EyeRobot: A) Phantom for scanning tests - Scotch tape (4 layers) with two human hairs. Robot scanning pattern. B) B-Scan cross-sections extracted from the C) resulting volumetric C-Scan.

[16] Human hair is very opaque completely blocking the OCT signal. This creates voids in OCT regions located underneath the surface of the hair.

6.6.2 OCT Scanning with Micron

The Micron device provides an attractive platform for scanning the CP-OCT probes due to its small hand-held form factor and dynamic manipulation capabilities. Liu et al. showed that it is possible to use the Micron3D device as a bench-top scanner to make 100 μm wide OCT B-Scans at 10 Hz[17]. The resulting OCT images of ex-vivo porcine retina show the internal limiting membrane (ILM), a top layer of the retina which is 2–3.4 μm thick [11] and is difficult to localize during a vitreoretinal surgery. This section presents the initial integration of the Micron manipulator with OCT to produce a volumetric OCT imaging capability.

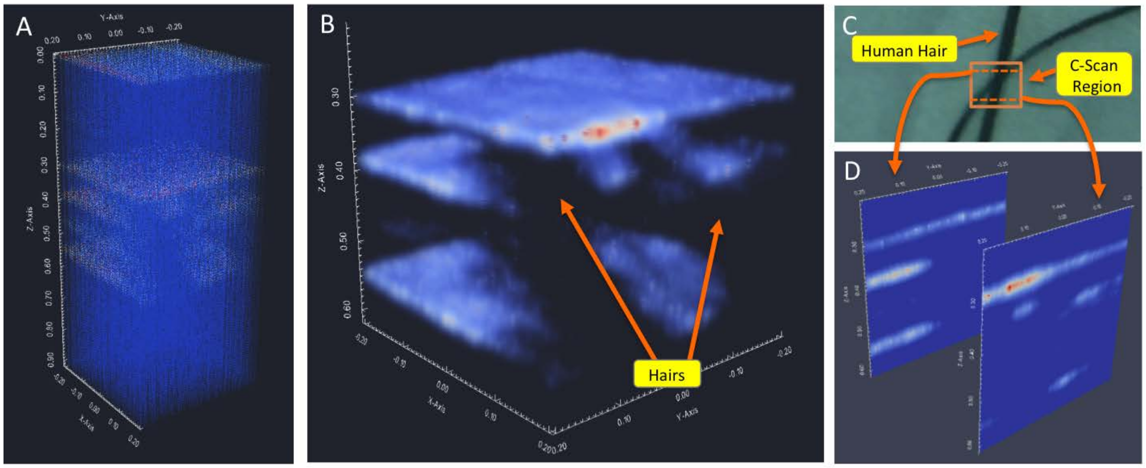


Figure 6.17: OCT raster scan with fixed Micron: A) Raw point cloud of a 0.4 × 0.4 × 1.3 mm Micron3D C-Scan of a phantom made of two hairs between layers of scotch tape. B) Volumetric rendering of the C-Scan. C) Location of the C-scan region and two cross-sections on a photo of the phantom. D) Cross-sectional slices showing the voids corresponding to the hairs.

The raster scanning pattern from the previous section was used to command the point-to-point motion of the Micron3D. The pattern covered a rectangular area of 0.4 × 0.4 mm with way points separated by 7.5 μs. The cubical volume has a depth of 1.3 mm, the range of an OCT

[17] The Micron3D was fixed rigidly relative to the sample

A-Scan signal. The scan took ∼3 seconds and included collection of ∼3K A-Scans transformed by the corresponding pose. The resulting point cloud is rendered in Figure 6.17A. Each A-Scan was acquired when the tool tip was within $\pm 2\mu$m of the target way point. The device was fixed relative to the sample, which was made out of human hairs sandwiched between multiple Scotch tape layers, see Figure 6.17C. The reconstructed C-Scan (Figure 6.17B) clearly shows the voids representing the crossing hairs. This is verified in two slices extracted from the volume shown in Figure 6.17D.

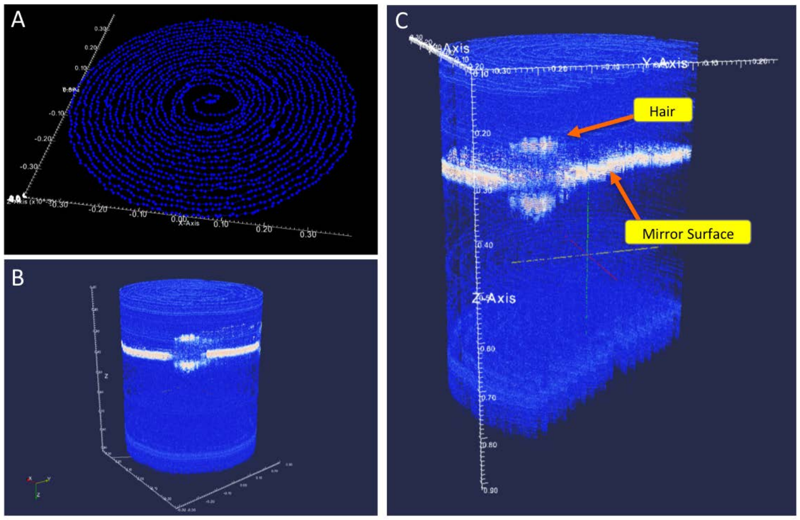


Figure 6.18: OCT spiral scan with fixed Micron: A) Spiral scan pattern (0.4 mm radius, 0.02 mm radial offset) generated with a fixed Micron3D. B) Spiral C-Scan point cloud of a human hair on a mirror. C) Cross section of the spiral C-Scan.

The rectangular waypoint pattern requires many accelerations and decelerations of the instrument, which adds to the overall scan time. One way to minimize these rapid direction changes is to follow a spiral pattern starting from the center, traveling in circles with progressively larger diameters. Figure 6.18A shows the measured locations for a 0.4 mm spiral motion that is constructed by sweeping an arc with a radial displacement of 20 μm at every full turn, and creating a waypoint every 3 μm arc lengths. The time to execute a full scan is ∼1 second while collecting ∼2.5K A-Scans.

The resulting scans of a human hair placed on a mirror surface are shown in Figure 6.18B-C.

Additional Micron-assisted hand-held scanning Behaviors were developed and presented by Yang and Balicki et al. [112]. One is a hand-held spiral C-mode scan that is designed to scan a particular region with minimal input from the operator. It is performed by holding the Micron6D with the OCT probe tip near an area of interest. Upon a trigger event (e.g., pedal or GUI button press), this location is the center of an automatic spiral motion in the plane perpendicular to the shaft of the OCT probe. Micron6D corrects for any hand motion during the scan ensuring that the probe tip is traveling in a spiral[18]. The speed of the probe tip along the spiral trajectory was set at 2 mm/s. The commanded spacing between adjacent cycles of the spiral was 50 μm. The total diameter of the scanned spiral was 1 mm. The trajectory is shown in Figure 6.19A. Because the OCT system[19] sampling rate is higher than the sampling rate of ASAP, linear interpolation between measured poses of the probe was used to estimate the pose corresponding to each OCT A-mode scan or "data point." All pixels in the particular A-Scan were transformed into the ASAP coordinate frame by the corresponding ASAP-to-Micron probe transformations and combined into a volumetric intensity point cloud.

Figure 6.19C shows the result of the spiral C-mode scan over rectangular etchings (group 1, element 4) on a glass plate which is a USAF 1951 MIL-STD-150A resolution test chart (R3L3S1P, Thorlabs Inc., see Figure 6.19B). A false-color representation and thresholding were used to aid visualization using the VTK software library [163]. The spiral contains 17862 individual A-mode scans. This included 7917 poses measured by ASAP; hence, the scan time was 7.9 s. The space between bars in the scanned image was estimated using a measurement widget in the visualization software (ParaView, Kitware, Inc.). The 176 μm space between bars on the test chart is measured

[18] Scans work reliably as long as the device is within its working envelope, and within the tracking space of the ASAP.

[19] The OCT configuration used for the hand-held Micron6D scans are described by Huang et al. [160].

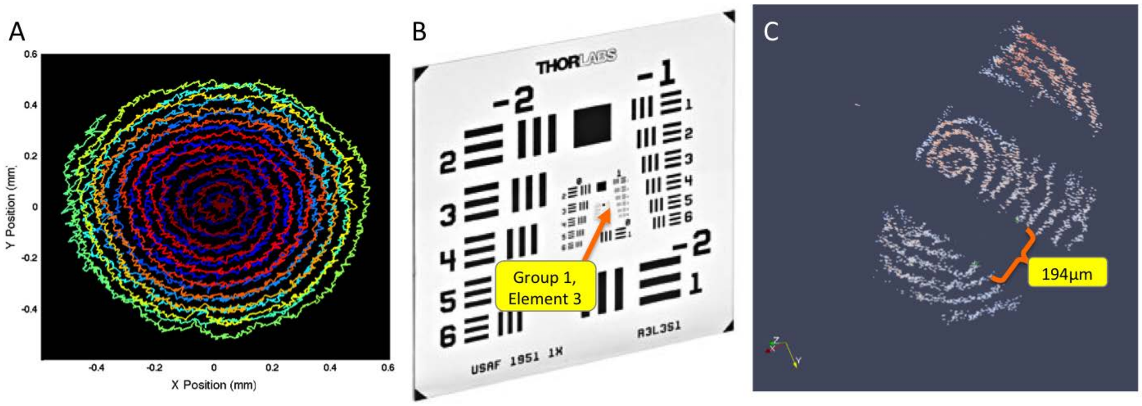


Figure 6.19: OCT scan with hand-held Micron: A) Spiral pattern (1 mm diameter) of the OCT probe generated by the Micron Device. Two passes, forward and reverse motions are visible. B) USAF 1951 MIL-STD-150A resolution test chart. (C) Result of spiral C-mode scan of group 1, element 4. The three rectangular regions representing the element in the chart are visible in the scan. The space between bars, which is 176 μm on the chart, is measured as 194 μm.

to be 194 μm in the acquired scan. A possible source of this measurement discrepancy is the thresholding of the volume which removes noisy edge data that is generated by the conical imaging volume of the OCT beam[20].

The second Behavior, called Z-Scan, creates a volumetric image by creating automatically repeating B-mode scans that are actively translated by the operator[21]. The Micron6D manipulator oscillates the probe along a virtual line in a triangular wave pattern at 5 Hz. The motion of the tip of the probe is 3.0 mm peak-to-peak at a speed of 30 mm/s. During scan execution, the center of the virtual line travels with the Micron6D handle, which is moved manually in a perpendicular direction to the “back-and-forth” scanning pattern, resulting in a zigzag scan pattern with a total of ~50 scan lines (see Figure 6.20B). The scanned material in this case was a stack of three layers of Polyester Tape (8911, 3M). Each layer was 50 μm thick (30 μm adhesive and 20 μm backing).

[20] Note that higher scanning speeds, especially in water, could cause additional positioning errors from tool shaft bending due to drag.

[21] The Z-Scan method was developed in collaboration with Cam Riviere.

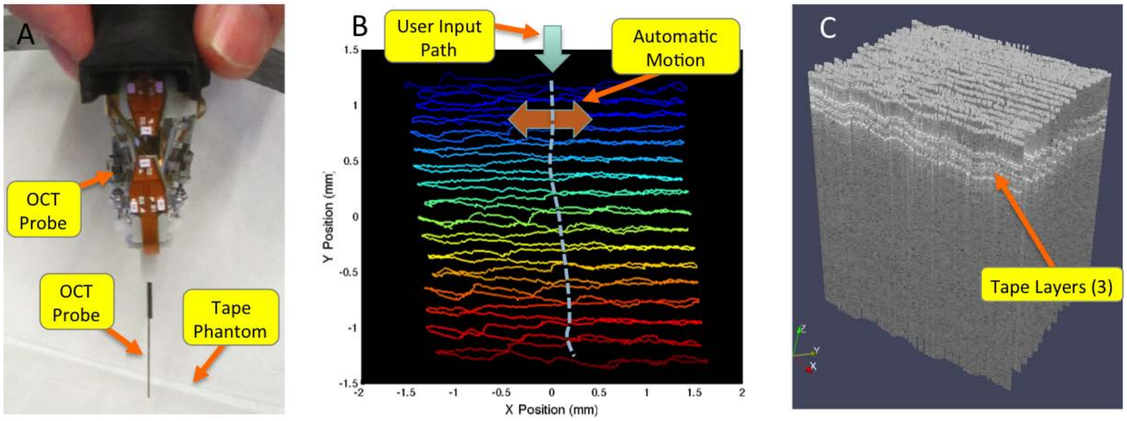


Figure 6.20: OCT Z-Scan with hand-held Micron: A) Micron6D with OCT probe over tape phantom. B) Z-Scan pattern created by translating the Micron6D while it is oscillating the tool laterally. C) Reconstructed Z-Scan image of three layers of tape. The 3 layers are evident near the top of the scanned volume.

The result of the zigzag B-mode scan is shown in Figure 6.20C. The three layers of the tape can be clearly seen near the top of the scanned volume. The total pattern contains 11272 individual A-mode scans. This included 5045 measured by ASAP; the scan time was 5.0 seconds.

The data presented indicate the general feasibility of acquiring surgically useful volumetric OCT scans using a hand-held Micron manipulator. As it is now, the scanning speed is not fast enough to account for motion artifacts caused by a moving sample. This can be partially corrected with post-processing software alignment and refinement of the control system for improved speed and accuracy.

6.7 Automatic OCT Spectrometer Calibration

The spectrometer is a sensitive component of the inexpensive CP-OCT imaging system. The characteristics of the spectrometer naturally change over time, due to both the environmental effects such as temperature and vibration [177], and to poor handling practices, which are common in surgical facilities. This results in significant degradation of OCT system performance in terms of resolution and sensitivity, and also erroneous depth ranging. Therefore, monitoring and recalibration of the spectrometer may be necessary for each session.

Conventional spectrometer calibration methods involve measuring the known spectrum of an external light source [178] or comparing measurements taken by a well-calibrated commercial optical spectrum analyzer [179]. These methods use specialized equipment, are time-consuming, and require specialized measurement protocols in a bench-top testing environment. As a result, they are rather inconvenient in a clinical setting. Furthermore, the calibration may be affected by an unknown or changing refractive index of the medium, thus complicating the determination of the distance between the probe and the structures of the samples being imaged. Since OCT measures optical path length, which is the product of physical distance and the medium's refractive index, an incorrect estimation of this physical distance can cause inaccuracies in imaging, targeting errors, and can lead to robot control instabilities in the case of image-guided, robot-assisted surgery. A simple and automatic OCT calibration protocol that addresses these issues is required.

Automatic spectrometer calibration (ASC) proposed by Liu and Balicki et al. [64, 180] leverages the precision and controllability of the robot to perform on-the-fly calibration during a typical OCT scanning operation[22]. This does not require any additional equipment beyond what is already available in the integrated *eyeSAW* system. The calibration algorithm comprises two parts: A) estimating the mapping from CCD pixel-space to k-space, and B) estimating the physical A-Scan pixel size (e.g., 3.4 μm/px).

In spectral domain OCT, the A-scan is reconstructed by inverse Fourier transformation of the spectral interferogram in wavenumber space (k-space). However, the digital detector array

[22] This concept was developed jointly with Xuan Liu who focused more on the spectrum calibration while Marcin developed the robot based pixel calibration.

of the spectrometer does not sample data evenly in k-space. Therefore, for a correct A-scan reconstruction, it is critical to re-sample the spectral data from pixel space to k-space, based on the known wavenumber values corresponding to each pixel in the CCD array. M. Mujat et al. [177] reported an automatic spectrometer calibration, based on generating a perfect sinusoidal spectral modulation in k-space by inserting a thin glass slide into the optical path. The alternative method, summarized here and described in full detail in [64], forgoes this calibration plate, instead using the specimen itself to generate data to calibrate the OCT system during scanning. As a result, spectral interferograms used in the ASC method can be derived directly from an arbitrary sample, even one with complex internal structures, like the ones found in the retina or other biological samples. Their spectrums do not have a perfect sinusoidal modulation that would be required for Mujat's algorithm [177].

The spectral mapping is achieved by using a zero-crossing detection technique which is based on the fact that zero-crossing points in a spectral interferogram are almost evenly spaced in k-space within a given spectral interval. Although the modulated signals come from different depths and result in different frequency components in the spectral interferogram, the surface reflection of the sample generates a large spectral modulation due to a refractive index discontinuity, and essentially determines the fundamental frequency of the interference fringe. A band-pass filter centered at the fundamental frequency of the spectral interference fringes is used to remove the DC component in the spectrum. Then, standard zero-crossing methods are applied to determine the separation of sinusoidal inflections and used in an iterative process to optimize a polynomial mapping from the spectrum to k-space that results in a sharp A-Scan (tall and narrow peaks in the A-Scan image) using a sharpness metric from [181].

To demonstrate the feasibility of this calibration step, the setup in Figure 6.11 is used with

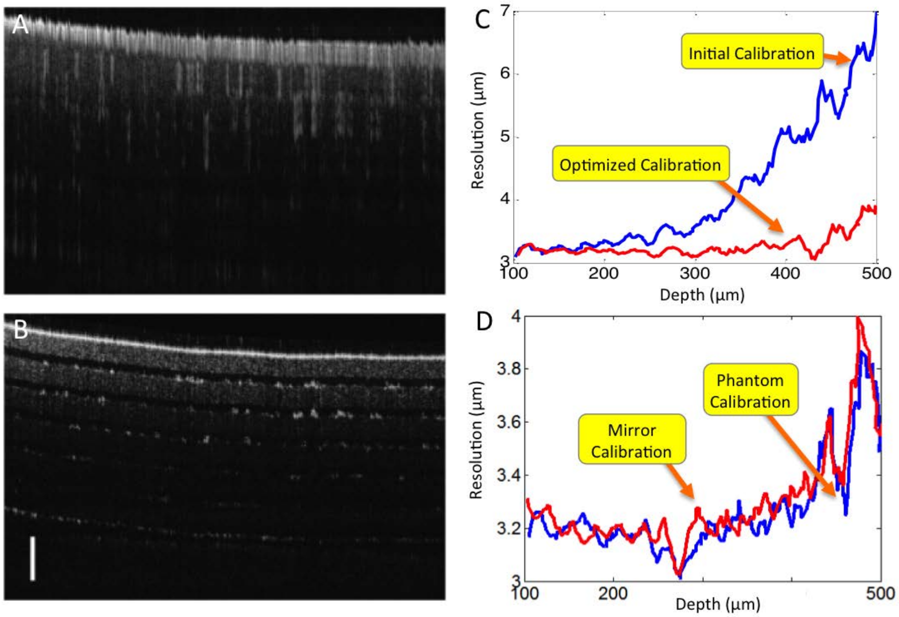


Figure 6.21: Automatic spectrometer calibration. A) B-Scan image of multilayer tape phantom obtained without a calibration. B) B-Scan image after ASC using the scan itself for calibration. C) Comparison of Full Width Half Max (FWHM) resolution at different depths based on initial estimation of polynomial and optimized. D) Comparison of FWHM resolution for standard calibration [177] (red) and one using multi-layered phantom by ASC (blue).

a six layer cellophane-tape phantom (see Section 3.3.4). The spectral data is collected while the robot translated the imaging probe across the sample. ASC is performed using a random spectral interferogram from that set and a B-Scan image is composed from the resulting A-Scans, as shown in Figure 6.21A. The individual tape layers are clearly visible in contrast to the B-Scan in Figure 6.21B, where the OCT image is obtained by directly performing IFFT on the spectral data and the resulting layer structure is hardly discernible. Figure 6.21C shows the significant improvement in axial resolution after iterative calibration (ASC). In comparison to the standard method by Mujat et al. [177], ASC has a very similar performance as shown in Figure 6.21D.

Just as in the method proposed by Mujat et al. [177] the above step does not produce the values for wavenumber limits, therefore the physical pixel spacing (effective width of each pixel) of the OCT A-scan is still unknown after calibration. An additional calibration step is therefore required. A simple approach involves the robot moving the OCT probe axially at known increments relative to the surface of a sample. The sample is static relative to the robot's base. Corresponding spectrums are collected and the resulting A-Scans (after ASC processing step 1), are then processed to detect surface positions in A-Scan pixel coordinates (i_n pixel of the surface for sample n) using one of the algorithms in Section 6.3. The exact pixel spacing K_z is obtained through a least squares linear regression based on the known robot positions (Z_n, where n is the sample number) and corresponding depth ranging derived from the OCT signal using the following set of equations:

$$Z_n = i_n K_z + z_0 \tag{6.6}$$

Another approach is required to perform this second step during a scan, especially if it is performed in an unknown medium[23]. This also addresses the case where the sample might be moving, or when sample topology or relative position of the probe to the sample are unknown.

[23] The pixel spacing calibration is also required for other OCT systems that are imaging in an unknown medium, or to calibrate a new type of probe.

The solution to estimate the physical pixel spacing (K_z) involves separating the known commanded robot motion $Z_0 \cdot sin(\omega t)$ from the unknown motion $Z_s(t)$ of the surface. Both of these contribute to the relative motion of the surface in the A-Scans as shown in Figure 6.22A and are represented by the following equation:

$$Z_r(t) + Z_s(t) = i(t)K_z + z_0 \tag{6.7}$$

The simplest version of the algorithm modulates the robot motion sinusoidally with frequency f_0 along the A-Scan axis of the probe being held by the robot:

$$Z_r(t) + Z_s(t) = Z(0) \cdot sin(2\pi f_0 t) + Z_s(t) = i(t)K_z + z_0 \tag{6.8}$$

This assumes that the motion of the sample is random, or is significantly different than the robot motion. Non-periodic robot motion will also work, and may be required to avoid interference and facilitate filtering in the cases where periodic motion of the sample is expected[24]. The corresponding, time-synchronized A-Scans (after ASC processing step 1), are then processed to detect surface positions ($i(t)$). Applying a band pass filter ($f_0 \pm \epsilon$) on the A-Scan range data removes the components contributed by only the motion of the surface $Z_s(t)$. This leads to a familiar set of linear equations:

$$Z(0)sin(2\pi f_0 t) = i(t)K_z + z_0 \tag{6.9}$$

Thus, these are solved for the physical pixel spacing K_z using least squares linear regression[25].

In order to assess the pixel spacing calibration, a system similar to the one in Figure 6.11 is used with a multi-layered cellophane-tape phantom, which is imaged in air. The phantom is driven

[24] This method assumes that the sample motion is not affected by the motion of the probe itself. This might not be true if the displaced fluid in front of the oscillating probe compresses or translates the sample tissue, especially if the probe has a profile with large surface area and is close to the sample surface.

[25] Another way to arrive at this result is to perform Fourier transform on the range data yielding peak i_f corresponding to the modulation frequency f_0. Same analysis of the robot position data yields a peak (Z_{r_f}) at f_0. The physical pixel distance is calculated by taking the ratio between these two peak values : $K_z = ^{Z_{r_f}} /_{i_f}$. See [180] for more details.

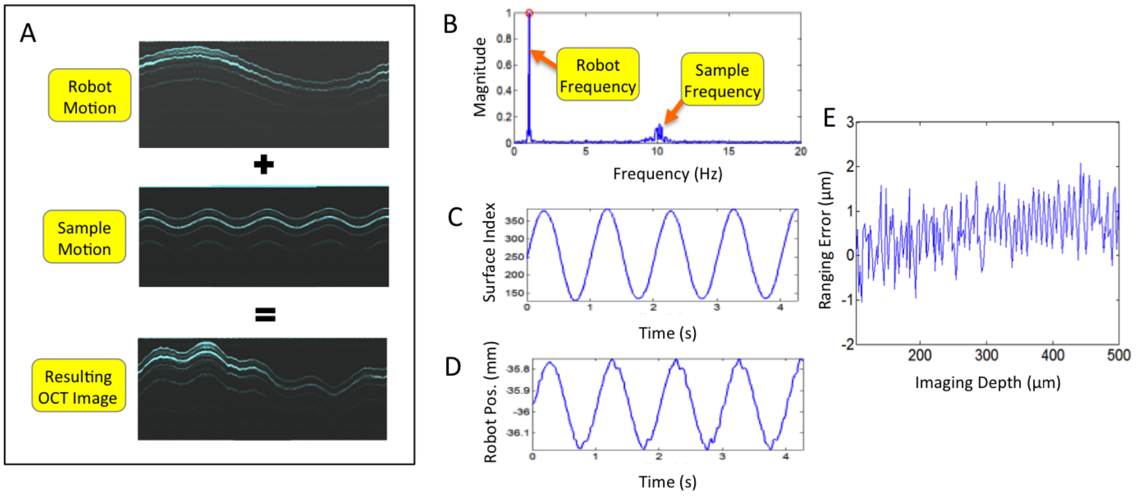


Figure 6.22: Robot-assisted OCT calibration. A) OCT scans of tape phantom for 1) only robot motion, 2) only sample motion, and 3) both robot and sample moving. B) Fourier analysis showing two peaks corresponding to robot (1 Hz) and sample motion (10 Hz). C) The pixel coordinates of the surface location after filtering out the sample motion. This corresponds to the displacement of the surface in the A-Scans due to robot motion. D) Commanded robot position. E) Ranging error after calibration.

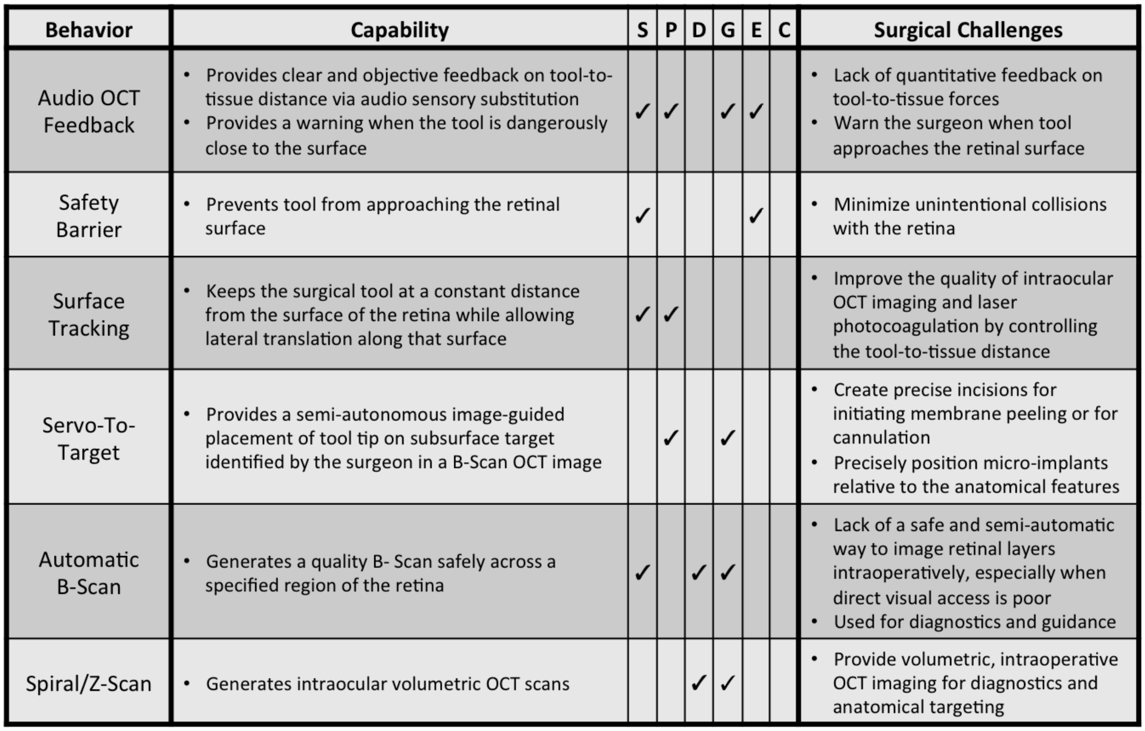
by an actuator in 1 D along an axis that is parallel to the OCT probe (Z axis of the tool) with a 10 Hz sinusoidal wave. The robot was commanded to move only in the Z Axis of the OCT tool with a 1 Hz, 400 μm peak-to-peak sinusoid. One of the interferograms is used to perform step 1 of the ASC calibration. The resulting calibration is applied to the rest of the collected data and processed into A-Scans. For each A-scan, the range tracking algorithm extracts the pixel index of the phantom's surface. Figure 6.22B shows the frequency content of these samples with clear peaks at 1 and 10 Hz corresponding to the motion of the robot and the sample, respectively. Figure 6.22C shows the robot position derived from OCT range signals after band-pass filtering and Figure 6.22D shows the timestamp-synchronized commanded robot position. Using these two data sets, a regression is performed using Equation (6.6). The pixel spacing for this particular OCT configuration is 1.58 μm. Figure 6.22E shows the difference between distance to sample surface measured with the OCT and the measured robot position. These errors are negligible considering the resolution of the robot ($\pm 1\mu m$) and the pixel spacing.

To test this automatic calibration algorithm, the forearm of a healthy human volunteer was scanned. The calibration results have effectively the same performance results as the conventional methods and the resulting B-Scan images are qualitatively identical, see [64]. The proposed calibration is a generic, automatic spectral calibration method that can be implemented in any spectrometer-based FD-OCT system coupled with a precise axial positioning actuator, which is often used in OCT scanners.

6.8 Conclusions and Future Work

This chapter presented Behaviors that incorporate OCT imaging technology to address a variety of surgical challenges in vitreoretinal surgery. These Behaviors are summarized in Figure 6.23. The provided capabilities focus on augment surgeons ability to precisely and safely position the tool relative to the retina surface. Others focus on providing intraocular imaging for anatomical target guidance. These Behaviors address surgical challenges of deficient visual depth perception, lack of detailed information of the cross-sectional anatomy of the retina, effects of natural hand tremor and inherent imprecise tool maneuvering.

Although mechanically challenging, very small instruments (∼0.5 mm diameter) incorporating fiber optic OCT can be constructed and used to identify tissue boundaries beyond the tip of the instruments. As demonstrated, such sensor can provide real-time feedback on the distance from the tool tip to a surface and can be used by a cooperative robot to enforce safety barriers or support surface following. Further, the instrument can be scanned laterally to construct 2D and 3D images from the single A-mode images produced by the OCT sensor. These can be combined



Behavior	Capability	S	P	D	G	E	C	Surgical Challenges
Audio OCT Feedback	• Provides clear and objective feedback on tool-to-tissue distance via audio sensory substitution • Provides a warning when the tool is dangerously close to the surface	✓	✓		✓	✓		• Lack of quantitative feedback on tool-to-tissue forces • Warn the surgeon when tool approaches the retinal surface
Safety Barrier	• Prevents tool from approaching the retinal surface	✓				✓		• Minimize unintentional collisions with the retina
Surface Tracking	• Keeps the surgical tool at a constant distance from the surface of the retina while allowing lateral translation along that surface	✓	✓					• Improve the quality of intraocular OCT imaging and laser photocoagulation by controlling the tool-to-tissue distance
Servo-To-Target	• Provides a semi-autonomous image-guided placement of tool tip on subsurface target identified by the surgeon in a B-Scan OCT image		✓		✓			• Create precise incisions for initiating membrane peeling or for cannulation • Precisely position micro-implants relative to the anatomical features
Automatic B-Scan	• Generates a quality B- Scan safely across a specified region of the retina	✓		✓	✓			• Lack of a safe and semi-automatic way to image retinal layers intraoperatively, especially when direct visual access is poor • Used for diagnostics and guidance
Spiral/Z-Scan	• Generates intraocular volumetric OCT scans			✓	✓			• Provide volumetric, intraoperative OCT imaging for diagnostics and anatomical targeting

Figure 6.23: List of OCT-based Behaviors with specific capabilities they provide, and the corresponding surgical challenges that they address. Behaviors are also characterized by capability types: S-Safety, P-Precision, D-Diagnostics, G-Guidance, E-Education, C-Communication.

to enable surgeon to identify a subsurface target in a scanned image followed by robot-assisted, automated placement of the instrument tip on the target. Many of these capabilities have a great potential to be very useful in future clinical vitreoretinal systems. The following points describe the lessons learned during the OCT Behavior development, the resulting requirements for next design iteration, and proposed future work:

- The various Behaviors presented here were only developed into the prototype stages for use in bench-top setting and in vivo rabbit experiments, and will require more design iterations to create robust versions necessary for clinical trials.

- The current fiber-based OCT imaging technology provides very good results in most scenarios. However, there are still some practical issues to address, including: difficult manual microfabrication processes, optical path occlusion by stray particles in the aqueous environment of the eye, longer imaging range, etc. None of these seem insuperable, though, experimentation and design iterations will be needed.

- Many of the experiments were performed with the probe perpendicular to the sample surface. Although OCT can identify layers while looking obliquely into the tissue at the angles that will be encountered in retinal surgery, it is still necessary to account for approach angle to get correct range data, and also to improve the blurring effects of such angled approach. This can be improved in several ways, such as special optics (angled-viewing lens [162]) or robot pose feedback and tracking of tools in the stereo video microscope.

- Increasing the OCT imaging distance could improve the safety of intra-ocular scanning with a fiber-optic probe. This can be accomplished with a sapphire ball-lens probe design [161], which provides a more focused “sweet spot” at a given distance from the probe. Such lens would also minimize the buildup of biological matter that tended to occlude the small imaging window during in vivo experiments. Furthermore, the sapphire protects the brittle glass of the optical fiber that is susceptible to cracking from inadvertent contact with metal objects, such as the trocar or other instruments.

- The concept of a smart surgical tool described in this chapter with integrated OCT and associated OCT-based virtual fixtures have spawned a parallel development by Kang et al. [182–186]. It focuses on a specialized hand-held tool with 1 DOF actuated axis along the tool shaft to provide constant offset, or collision avoidance and stability based on the integrated enface OCT imaging. However, since it only provides a single DOF for both sensing and actuation along tool shaft, the tremor in the transverse plane is not compensated, which may still cause distortion in OCT imaging, and not address the effects of hand tremor in tissue manipulation. The authors have looked at ways to stabilize grasping using OCT-integrated forceps in anterior eye procedures.

- The OCT-image based servoing (Servo-To-Target Behavior) has also been investigated by the same group as presented by Zhang et al. [183] where the goal is to use a 1 DOF actuated, hand-held, OCT micropick [182] to perform an automatic 100 μm incision into the surface of an intralipid phantom. They showed that this was more accurate than with a freehand approach.

- OCT imaging should be integrated into other common surgical instruments such as a forceps [186], or the light pipe so that OCT-based safety Behaviors are available during the whole procedure with a variety of instruments. Furthermore, incorporating OCT and force-sensing functionality into a single instrument could provide a more complete feedback (position and force) in a particular task.

- OCT Range-audio sensory substitution is useful in warning the surgeon of impending surface contact. Despite the precise range information feedback, the limited manipulation precision by the human hand still requires robot assistance for tool stabilization to provide more effective and precise positioning.

- Microscope-integrated real-time OCT could provide 3D information for targeted treatments of the retina. If prolonged OCT exposure is deemed safe, such modality is a good complement to the intraocular OCT probes, and can also provide excellent feedback for the safety Behaviors such as the Safety Barrier.

- End-point sensing is essential in precise robotic assistance where the tool shaft is flexible and a correct estimate of the tool-tip location cannot simply rely on forward kinematics calculations. OCT range sensing is a good candidate for such feedback. Furthermore, OCT range sensing can provide contextual information for proximity-based cooperative control gain adjustment, e.g., when the tool is close to a surface, a maximum speed limit can be imposed on the overall tool speed or just the speed in the vector normal to the surface.

- The robot-assisted scanning Behaviors show that it is possible to use a multi-purpose robotic assistants to perform volumetric scans using a 1 DOF OCT probe. Whether these will be practical for intraocular scans is still unknown, but in other surgical domains, such as cochlear surgery where direct visual access is very limited, it may be possible to use robotic scanners to provide near real-time imaging and navigation. Future experiments need to explore the effects of aqueous environments on robot-assisted scans, e.g., ideal scanning speed and pattern. Also, it is imperative that these high speed scans are performed with concurrent safety checks, such as collision avoidance with the retina and also the lens, and consider the exposure of the retina to the broadband laser source used for OCT imaging.

- Future work could include incorporating automatic positioning of the tool, similar to the Automatic B-Scan Behavior, to apply treatments in precise locations on the retina. Furthermore, the pre-planning trajectory should be executed by the robot while ensuring a safe distance from the surface of the retina. One immediate application is targeted retina laser ablation or cryotherapy.
- The Behaviors presented here are also compatible with other imaging modalities such as ultrasound, and functional and anatomical sensing of blood vessels, as demonstrated by Kang et al. [181], can be used to further refine the targeting capability.

6.9 Recapitulation of Contributions

OCT System (Section 6.2) Software and visualization methods were developed for a custom Fourier Domain Common Path OCT system. CP-OCT is a relatively inexpensive system assembled from off-the-shelf components which interfaces with a fiber optic based imaging probe. It is specifically designed to provide sufficient resolution and imaging depth for intraocular real-time imaging. The software processes A-Scans at a rate of ~4 kHz, extracts distance to the surface, and composes 2D visualizations at 30 Hz. The system can also be particularly adapted for real-time handheld imaging, and can be used in microsurgical applications where limited access does not allow for conventional imaging. **Credit**: OCT Hardware designed and built by Xuan Liu. Software architecture and visualization methods developed by Marcin Balicki.

OCT Pick Instrument (Section 6.2.1) The first 25 Gauge OCT integrated ophthalmic micro-pick was developed to enable simultaneous A-mode imaging and tissue manipulation. The tool is built using a standard single-mode fiber and a modified hypodermic needle. The instrument tip is visible in the OCT and therefore it is inherently registered to the OCT probe. This provides a simple way to precisely position the surgical part of the instrument based on feedback from the OCT. This concept can also be extended to other surgical instruments, e.g., forceps. **Credit**: Marcin Balicki.

Range Sensing Algorithm (Section 6.3) This algorithm was developed to track the surface visible in incoming A-Scan images in real-time. It improves over previous attempts in that it works on noisy biological data samples and is able to deal with temporary floating obstructions. The algorithm can be applied in any OCT scanners that are used for real-time range sensing feedback. **Credit**: Marcin Balicki.

Audio Range Feedback (Section 6.4) This is an extension of the audio force feedback from Section 5.3, but communicates OCT range-sensing information using distinct audio cues. It was mostly used to supplement visual and haptic virtual fixtures by providing a more complete and intuitive communication, especially in cases where the OCT-based haptic feedback might have been surprising or counter-intuitive. The distance-to-surface audio feedback provides a warning when the instrument is dangerously close to the surface. This can be used in many micro-surgical scenarios where delicate structures may be injured by undesirable collisions with the instrument, e.g., cochlea implant procedure. **Credit**: Marcin Balicki.

Safety Barrier Behavior (Section 6.5.1) This modified cooperative robot control uses range-sensing feedback to prevent the instrument from approaching the target surface closer than a pre-specified distance, with high precision ($\sim 10\ \mu$m). The novel algorithm can be used to prevent accidental collisions with the retina, or any other surgical scenario where delicate tissue can be inadvertently injured by the instrument. **Credit**: Marcin Balicki.

Surface Tracking Behavior (Section 6.5.2) A novel robot control method was developed that uses OCT range sensing and a robotic assistant to keep the surgical instrument at a constant distance from the surface of the retina while allowing lateral translation along that surface. Surface tracking can be used to improve the quality of intraocular OCT imaging, or for precise laser photocoagulation. **Credit**: Marcin Balicki.

Servo-To-Target Behavior (Section 6.5.3) This new application of OCT-based surgical navigation allows for a semi-autonomous robot-assisted intervention. The system enables the surgeon to identify a subsurface target in a B-Scan OCT image, after which the instrument tip is automatically placed on the target by a robotic assistant. Such capability has the potential to aid in creating precise incisions for initiating membrane peeling, or to assist in vein cannulation. In other surgical domains (e.g., otology), it can be used to precisely position micro-implants relative to the anatomical features visible in the OCT. **Credit**: Marcin Balicki.

Automatic B-Scan (Section 6.5.4) This novel method generates a quality B-Scan safely across a specified region of the retina using the EyeRobot and intraocular OCT-based range sensing. To complete the scan, the surgeon first specifies two points on the retina using the OCT instrument. Then the robot travels between these two points while keeping a constant distance to the surface of the retina. These B-Scans require little effort from the surgeon, can be optimized for a particular OCT probe technology (constant offset), and generate higher quality OCT scans than a freehand option. It can also be used in scenarios where conventional tomographical imaging methods are inadequate, and visual access is limited by anatomical constraints. **Credit**: Marcin Balicki

Robot Assisted Volumetric OCT Scanning (Section 6.6) Various methods were developed for creating volumetric OCT images with a 1 DOF OCT probe and robotic assistance. These include automatic raster and spiral scan patterns, which are more suitable for lab-bench imaging, and the Z-Scan, which is acquired by the surgeon translating Micron6D over an area of interest while the Micron is sweeping the probe, creating a zigzag pattern. These methods offer significant improvements over other specialized OCT scanners in that they use multi-purpose robotic assistants and OCT-integrated instruments, and can be used when conventional OCT imaging is inadequate due to visual occlusions. Other applications include imaging the cochlear canal to assist in the placement of cochlear implants. **Credit**: Concepts and software developed by Marcin Balicki. Micron scanning implementation is a collaboration with the Micron team from CMU.

Automatic Spectrometer Calibration (Section 6.7) This novel method was developed to automatically calibrate a spectrometer, the vital component of an FD-OCT system. The method leverages the precision and controllability of a robot to perform on-the-fly calibration in a typical scanning environment. The algorithm estimates both the mapping between CCD pixel-space and k-space (A), and also the physical A-Scan pixel size (B). It improves over existing methods by not requiring any additional equipment beyond what is already available in the integrated *eyeSAW* system. Such calibration can also be performed in any environment. **Credit**: Collaboration with Xuan Liu, who developed method A), while Marcin developed method B).

Chapter 7

Visualization Behaviors

In vitreoretinal surgery, the surgeons primarily rely on visual feedback provided by a ophthalmic stereo microscope that is narrowly focused on the retina. I.e., the surgeon closes the control loop of the surgical instrument by directly visualizing its movements relative to target tissue in the oculars of the microscope and manually adjusting the instruments' position accordingly. There are a number of visualization challenges that make this already high-risk operation very difficult.

Ergonomically, the operating microscope forces the surgeon to maintain an uncomfortable position of the head and neck to achieve and maintain stereo vision through the eyepieces. This leads to fatigue during procedures and even to possible career-ending back pain. While operating on the retina, the surgeon must avoid colliding the instruments with the lens, which can cause cataracts, and, especially, with the retina, which can cause retinal tears. Many surgical tasks require the surgeon to be able to estimate the distance between the instrument tip and the retina within a range of microns. This often involves using a contact lens that significantly narrows the field of view in exchange for higher magnification. The side effect of this is a very narrow field of view ($\sim$5–15% at any one time), and increased cognitive load on the surgeon to localize this small “spot” relative to the much larger retina. Furthermore, the surgeons must be able to incorporate preoperative images and the surgical plan by mentally registering memorized visual features from these often few-days-old images, with the region of the retina they are able to see with the microscope. This task is very challenging, especially when the anatomy has changed due to surgical intervention or the field of view is very narrow and distorted by a damaged cornea, lens aberrations, and normal tissue obstructions. Furthermore, the direct visualization of the retina through the microscope does not provide any structural or advanced diagnostic functionality that may be beneficial during the operation.

This chapter explores the capabilities provided by video microscopy and the power of incorporating computer vision methods inside the *eyeSAW* environment. The visualization subsystem captures stereo video from the microscope, performs image enhancement, does retina and tool tracking, manages annotations, and displays the results on a 3D display. The surgeon uses the video display along with standard surgical tools, such as forceps and a light pipe, to maneuver inside the eye. The system is able to augment surgical feedback with visual sensory substitution, fuse intra-operative sensing information virtually in the field of view, provide a wider and more information-rich map beyond the current field of view to aid in self-localization, and offer intraocular image guidance to aid in localizing retinal targets, such as epiretinal membranes. Additionally, the introduction of 3D displays provides high quality 3D visual access to the surgical site (e.g., view of the retina) for the whole surgical team rather than just the surgeon and an assistant, which is the case in conventional microscope setups. More importantly, the 3D display allows more freedom of movement to the surgeons’ arms, head and neck, thus decreasing the strain caused by many hours of surgery.

7.1 System Design

The major goals of the visualization system include providing ergonomically- friendly and high quality 3D visualization of the surgical site to the surgeon and integrating computer-assistance by processing and augmenting the visual information. Since microscope visualization is the primary method of feedback for vitreoretinal surgeons, it requires a display system that is low latency, high frame rate and resolution, and wide dynamic range. All these affect the surgeon's ability to track tool and tissue motion, as well as visualize nearly-transparent, micron-thick structures, and monitor changes in the tissue throughout the procedure.

There have been very few attempts to provide computer-integrated microscope visualization methods in retinal surgery. One example is Berger et al. [187, 188] who developed a custom video injector containing a miniature cathode ray tube display that injected angiographic images into the microscope view. Another approach used here has been developed by Vagvolgyi and Balicki et al. and presented in [125, 189]. The system uses off-the-shelf cameras and microscope adapters (splitters) for image capture and a 3D LCD with passive or active shutter glasses for displaying the stereo microscope view. The captured stereo-video data is processed using a standard multi-core computing workstation. Figure 7.1 shows one of the two hardware prototypes which are described in Sections 3.1 and 3.2.

The most challenging software engineering aspects involve capturing, transferring, processing, archiving and displaying inordinate amounts of data in real-time. The software design considers these stringent surgical requirements, especially minimizing the display latency and providing high frame rates. Meeting these specifications is challenging while simultaneously executing computer

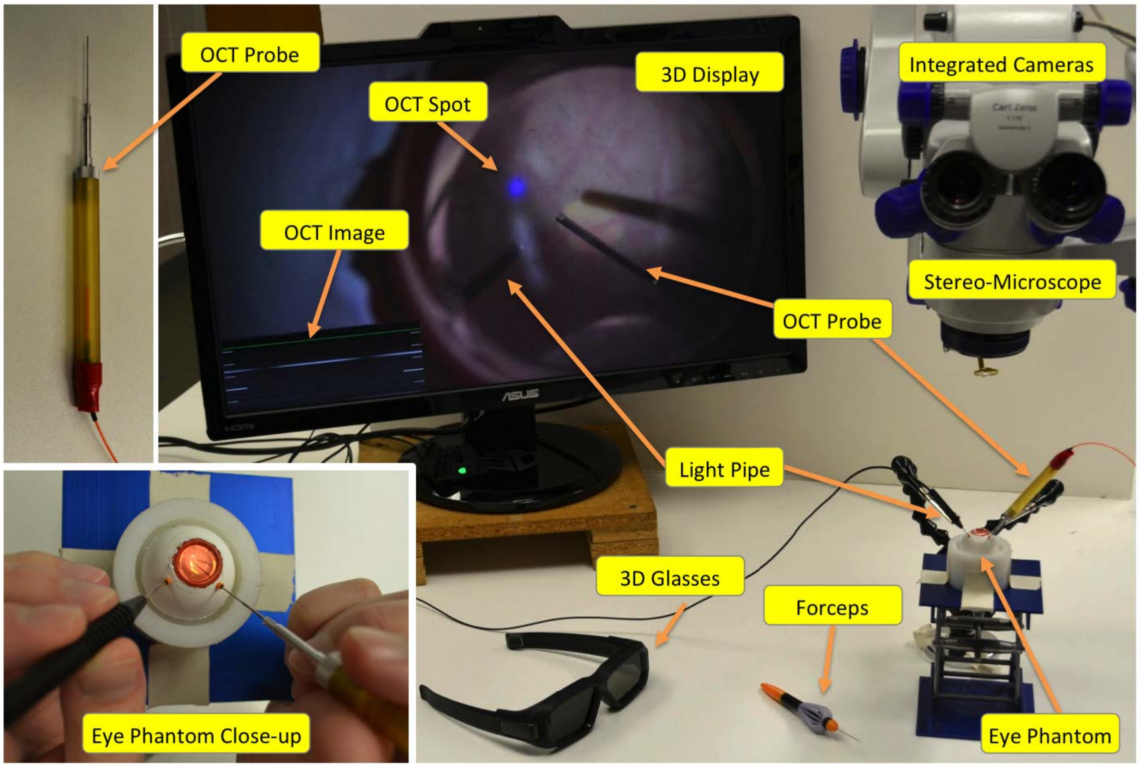


Figure 7.1: Visualization workstation used for M-Scan experiments.

vision algorithms, augmenting the surgical view, e.g., with overlays, and facilitating the exchange of information with the rest of the *eyeSAW* system. Figure 7.2 shows the software architecture of the visualization sub-system. It is built using *cisst* StereoVision library (SVL) [48, 125] and is described in detail in Section 2.5. SVL provides a wide array of highly optimized filters, such as capture device interfaces, image processing (background and tool trackers), network transmission functionality, overlays, archiving, and video formatting for stereo-specific displays. The library is highly multi-threaded, adopting the GPU streams concept [48], where each filter in a *Stream* has access to a pool of threads to process the latest video frame using the CPUs. The typical application runs at 20–30 FPS on a multi-threaded PC displaying 1920 x 1080 px stereo images with 2-4 frame latency.

The extremely large ($\sim$ 400 MB/s) stereo-video data stream has to be processed in real-time on a single machine due to imaging hardware constraints and limited processing and data transfer speeds. The SVL stream architecture allows for video data to travel between filters without expensive data copies. The exception is branching, where the video data is copied to another Stream which has an independent execution and does not directly affect the source (main) branch processing. The architecture lowers data duplication and data serialization/deserialization, and minimizes data transfers on already congested system communication buses. The main branch of the SVL Stream in Figure 7.2 is responsible for displaying the microscope view as fast as possible. This is achieved by performing minimal, display-related image processing on two threads, one per left/right channel. The more computationally intensive tasks, e.g., tool and retina tracking, tend to be slower and are performed in parallel on different branches. The parallel processing results are used by visual Behaviors and incorporated asynchronously into the visualization with overlays, or provided to other Behaviors located in different processes using the command pattern interfaces.

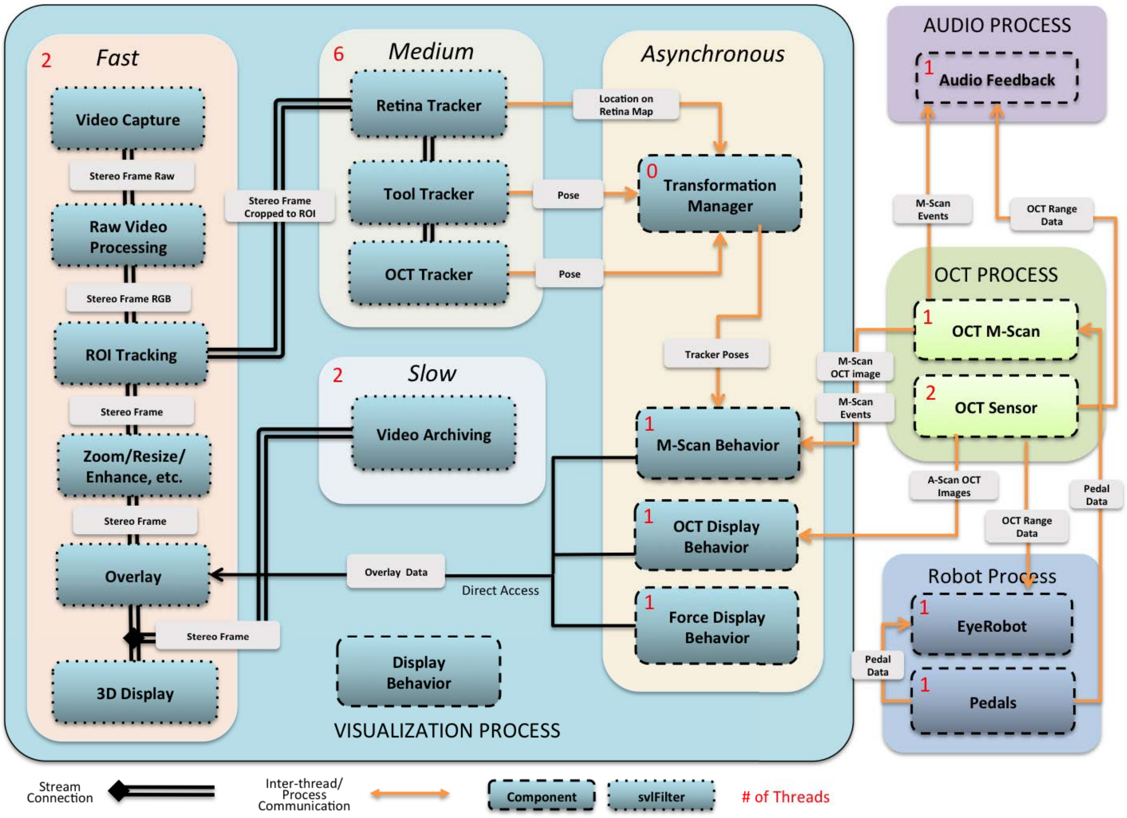


Figure 7.2: Visualization sub-system architecture based on *cisstStereoVision* library showing the branching of stereo-video Streams and connection to Behaviors located on other processes. Note: the Display Behavior has direct connections to the visualization Stream Components (not shown).

The Display Behavior manages the basic functions of the video Stream, such as magnification, brightness, image inversion, tracker properties, etc., providing the surgeon with the appropriate display capability and associated preferences for a given surgical task.

It is important to note that due to the operating microscope's complicated optical path, narrow depth of field, extreme image distortions, and complex illumination conditions, conventional computational stereo methods are very difficult to achieve. Therefore most processing is performed in image space and often on each left/right channel independently. By associating the overlays with particular image features in each channel, the above challenges are circumvented and augmented reality rendering appears very natural to the viewer, e.g., the overlays "attached" to the retina have the correct disparity and appear at the same depth as the retina surface.

7.2 Video Latency

The expert surgeons strongly believe that in order for a surgical video microscopy to be practical it needs to have low latency and high frame rate, approaching the visual acuity provided by a conventional stereo-microscope. To precisely measure the video latency of the *eyeSAW* visualization from the surgeon's perspective, a novel stand-alone hardware module was developed, see Figure 7.3. It is an inexpensive alternative to the common method of video recording both the trigger and the display screen with a single high speed camera and then manually analyzing the video frames.

This custom-built device comprises a standard green LED placed in the microscope's field of view, and a silicon photodiode detector (BPW 34 - OSRAM GmbH) that is attached directly using masking tape to the 3D display over the location of the LED in the image. A microcontroller (UNO, Arduino Inc.) is used to randomly trigger the LED and precisely measure the time for the

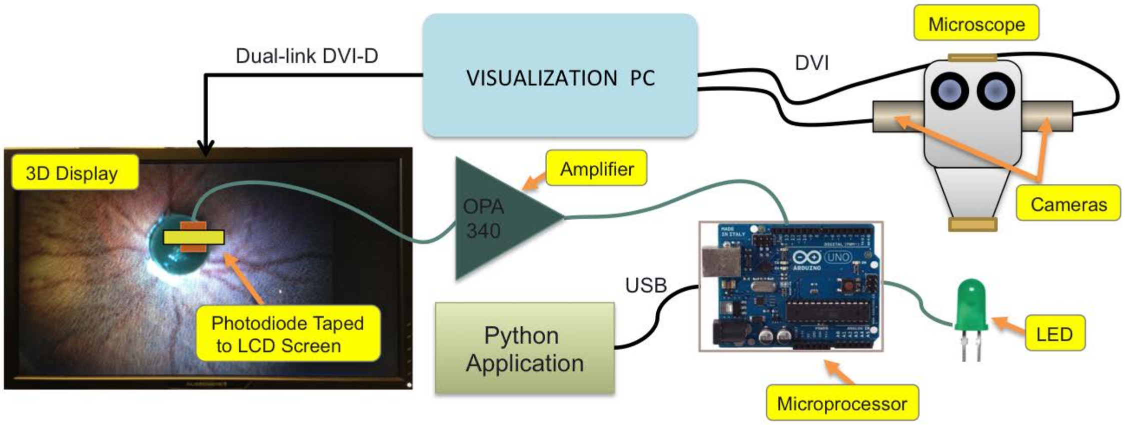


Figure 7.3: Schematic of the Video Latency Tester.

visual event to be detected on the video display with ∼ 250μs accuracy. The results are sent to a custom Python application via USB. For comparison in the initial trial, a simple stereo visualization application without any significant processing was tested. The experiments were executed with minimal overhead room lighting to test the worst case scenario where the camera shutter speed is slow to compensate for the low light.

The measured latency was 108.9 ± 7.5 ms at ∼ 40 FPS (N = 500). The same test was repeated with all the relevant *eyeSAW* components enabled (overlays, visual trackers, recording, etc.) operating under standard lighting conditions. The latency measurements were a bit higher at 147.9 ± 14.3 ms (N = 500) with the video frame rate of the main stereo feed running at ∼ 20 FPS while recording in HD quality at ∼ 17 FPS.

There are many sources contributing to the delay, such as shutter speed (∼ 25 ms), camera digitization, on-board processing and data transfer (∼ 20 ms), PCIe bus and RAM bandwidth (∼ 50 ms), and CPU processing (∼ 10 - 60 ms)[1]. Although any delay can affect surgical performance [190],

[1] The sources of the delays are difficult to measure directly and are only estimates.

the measured display latencies are within the functional range in comparison to tele-surgical systems which show a decreased task performance with delays over 250 ms [191]. With higher video latency (e.g., over 100 ms) surgeons need to adapt to the altered hand-eye coordination, and effectively slow down tool manipulation speeds and anticipate the tissue motion earlier.

This can be addressed through training, but it is expected that the next generation of the *eyeSAW* related hardware (cameras, displays, computer processing and memory bandwidth), and the use of processing parallelization on GPUs, the overall video performance will greatly improve. With target maximum latency[2] of $\sim$ 50 ms, frame rates above 33 FPS, and higher resolution video, the new systems will have to address even greater processing and bandwidth challenges.

7.3 Image Transformations Manager

The video subsystem contains several coordinate frames that are generated by various visual trackers and display configurations. These include the transformations from raw image frame to the display (affected by magnification, translation, etc.), and from the raw image to the region of interest (ROI), which is the bright circular area in the microscope view that is bounded by the iris and is often used by trackers. Other transformations are ROI to retina map, and location of instruments relative to the ROI. Besides providing a convenient, singular point of access, the management of these geometric and temporal relationships have grown complex enough to require a centralized transformations database called the Transformations Manager (TM) [3]. Figure 7.4 shows the typical set of coordinate frames that exist in the Transformations Manager, while Table 7.1 demonstrates the possible transformations between them.

[2] The video latency of the original daVinci teleoperation system is 56.54 ± 4.67 ms as measured with the latency tester.

[3] The Image Transformations Manager was designed in collaboration with Balazs Vagvolgyi.

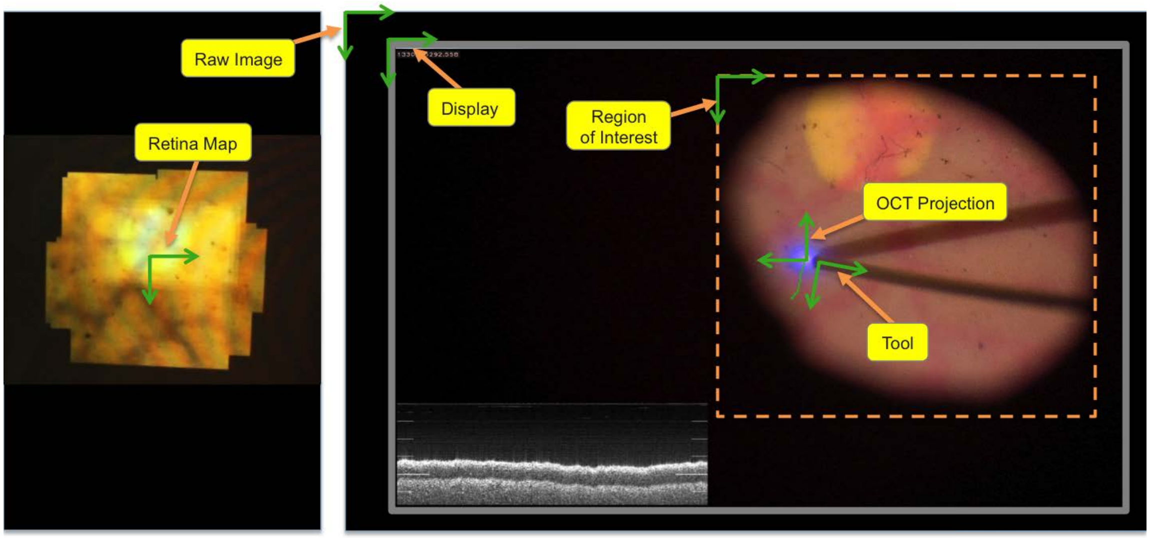


Figure 7.4: Coordinate frames that exist in the Transformations Manager.

The architecture of the TM is particular to a visualization subsystem where only image based transformations are stored and the temporal granularity is synchronized with the main display Stream. These transformations are stored in a table where each row represents the set of transformations to and from any coordinate system in the visualization process. Each transformation in the most recently generated row has a timestamp equal to the current video frame's timestamp. Upon the arrival of a new video frame, the previous transformations are copied over and all of their timestamps are updated (but are not valid). The transformations are typically stored as a pair of 3 × 3 matrices (4 DOF), each corresponding to left or right images, and are updated as various system components process the video frame and generate a transformation (e.g., from raw image frame to region of interest). Whenever a transformation is updated with new content, an event for that particular transformation type is emitted with the new transformation as the payload. This minimizes polling type read requests. In the case of low latency updates (e.g., retina tracker, 10 Hz vs. 25 Hz for video display), multiple rows of the transformation tree are updated. Each transformation object has methods that convert to and from any type of transformation if possible (e.g., 2 DOF to/from 3 DOF), and a flag to indicate if its content has been updated, or if it is an estimate. Mutex-based critical sections for reads and writes ensures a consistent state of the data. The table is implemented as a list of circular buffers where each column is a vector representing a particular transformation type. The vector size is preallocated and is sufficiently large for extended visualization sessions (e.g., 10 minutes). The programming interface allows for retrieval of the latest transformation by name or by nearest frame timestamp. Additionally, a software developer can access a list of a particular type of transformation based on range of timestamps. This is often used to build tool travel paths from a list of transformations, such as in the case of the M-Scan trajectory on the retina.

	Raw image	ROI	Display	Retina Map	Tool Tracker	OCT
Raw image		■	■	■	■	■
ROI	■		■	■	■	■
Display	■	■		■	■	■
Retina Map	■	■	■		■	■
Tool Tracker	■	■	■	■		■
OCT	■	■	■	■	■	

Table 7.1: Transformations provided by the Visualization Transformations Manager.

Particular attention should be considered when accessing the Transformations Manager component to minimize writes and reads that could negatively affect the performance of the system. The current implementation requires the developer to manually extend the coordinate transformation tree when additional transformations are to be added to the system. With this in mind, the TM functionality has been very practical in rapid development of new Behaviors by providing direct access to the various visual transformations in the display subsystem.

7.4 Retina Tracking

One of the most important functions of the visualization system is the ability to track the current view of the retina relative to an internal map of the retina. The map can be built in real-time from already visited regions or adopted from preoperative fundus images. The retina tracking process is typically divided into two sub-processes: frame-to-frame retinal motion tracking, which runs in real-time (e.g., 15 Hz); and global registration, which determines the location of a live retinal video image relative to the wide-area retinal map. This tends to be much more resource intensive due to a larger search window and more sophisticated matching algorithms. Currently, it runs asynchronously at a rate of 1 Hz. The registration is needed to initialize the tracker when the frame-to-frame tracking has failed. The registration feature can also be used to fuse multiple intraoperative retina maps, fundus images, and other similar retina representations[4].

Early attempts to register retinal imagery include the work of Douglas Becker et al. [192], which is based on detection and characterization of vasculature landmarks which are used as features in a global feature set matching process. This algorithm was designed for a non-surgical fundus camera which provides a wide viewing angle and very good retina clarity compared to typical vitreoretinal surgical views. Similarly, Shen and Stewart et al. [193] used vessel bifurications as features to register multiple fundus images offline to create a large retina map or a mosaic. More recently, Brian Becker [109] combined a fast vessel-segmenting algorithm with an occupancy grid

[4] See Brian Becker's PhD Thesis [109] for comparison of registration and tracking methods from other domains that are relevant to the retina tracking problem.

for mapping and an iterative closest point algorithm for localization (eyeSLAM). He was able to track the retina using pre-recorded in vivo data. In another approach, Berger et al. [188] presented a system for direct overlay of previously-stored photographic and angiographic images onto the real-time slit lamp fundus view. The registration algorithm used was developed by Barrett et al. [194]. It uses six single dimensional templates that are strategically positioned based on general retina anatomy. This approach is only able to track minor 2D motions of the retina in the image. Their approach is unique but only works on a fixated eye.

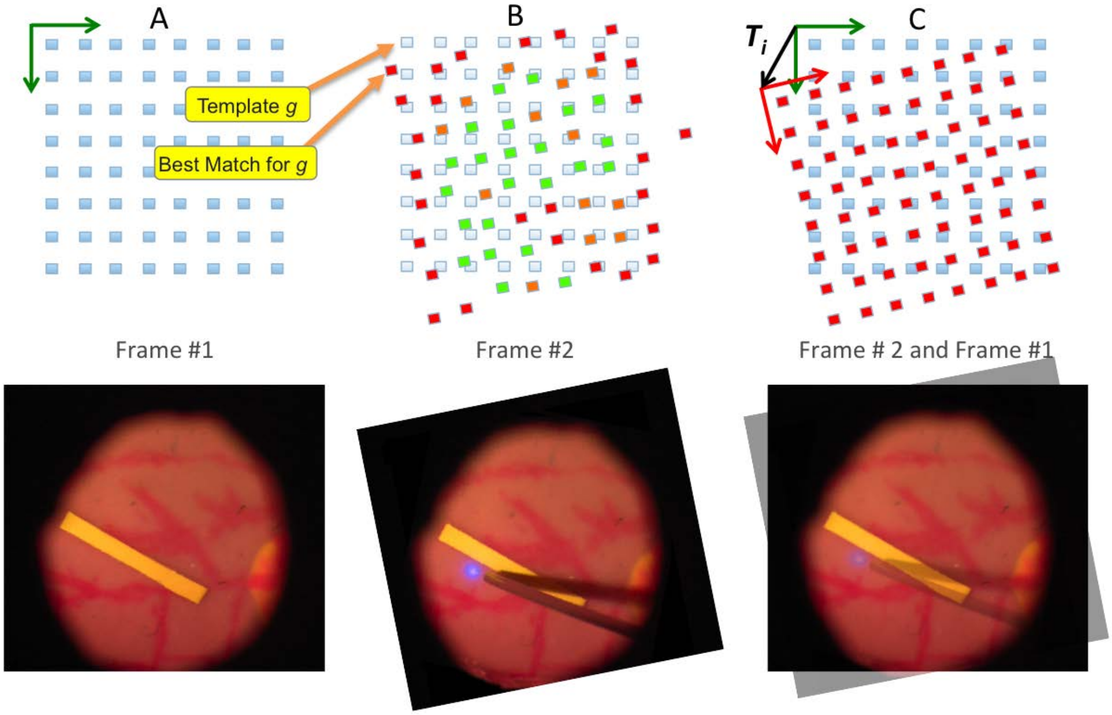


Figure 7.5: A) Structured template grid of retina model. B) Templates matching with candidate image. Colors show level of match confidence: red is low, orange is medium, and green is high. Note: matching confidence is low over the area with the tool and its shadow. C) Back projection of original templates.

The retina tracking method[5] used in *eyeSAW* is described in full detail by Balicki and

[5] The retina tracking method was developed by Balazs Vagvolgyi.

Vagvolgyi et al. in [189]. The algorithm continuously estimates a 4DOF transformation (rotation, scaling and translation) between the current ROI and an internal planar map of the retina, the content of which is updated after each processed video frame. The motion of the retina in the images is computed by tracking a structured rectangular grid of 30 x 30 px templates equally spaced by 10 px (see Figure 7.5). Assuming that rotations and scale are small between image frames, the translation of individual templates (g) visible within the ROI is tracked by a local exhaustive search using Normalized Cross Correlation as the illumination invariant similarity metric (C_{gj}) that operates on the three color channels (RGB).

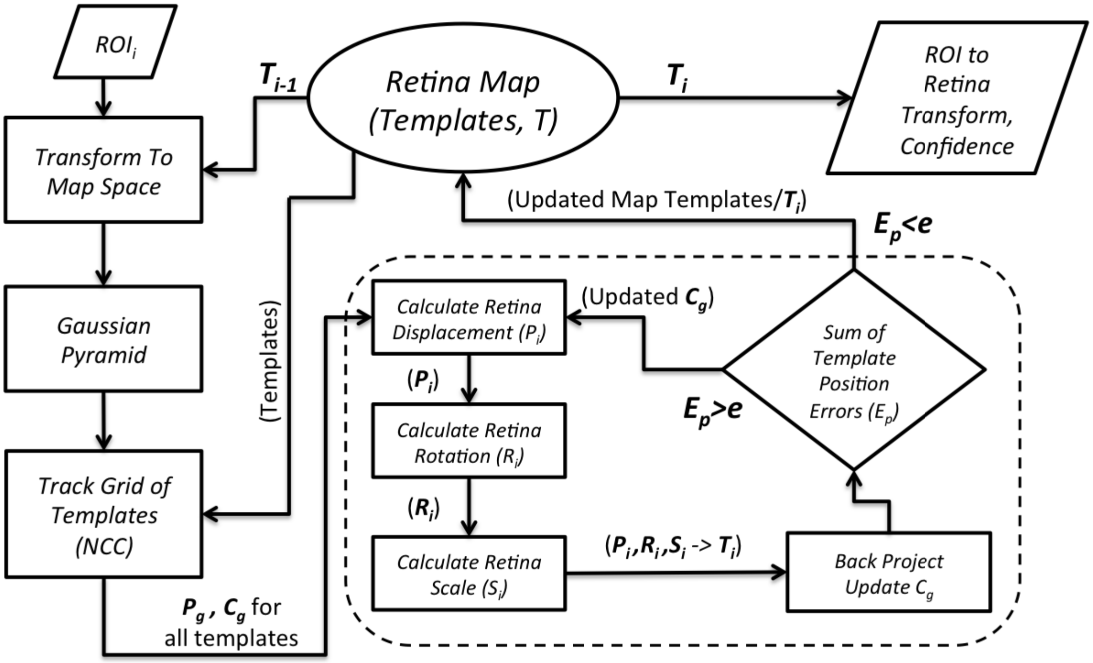

ROI$_i$
Transform To Map Space
Gaussian Pyramid
Track Grid of Templates (NCC)
T_{i-1}
Retina Map (Templates, T)
T_i
ROI to Retina Transform, Confidence
(Templates)
(Updated Map Templates/T_i)
E_p<e
Calculate Retina Displacement (P_i)
(P_i)
Calculate Retina Rotation (R_i)
(R_i)
Calculate Retina Scale (S_i)
(Updated C_g)
Sum of Template Position Errors (E_p)
E_p>e
(P_i,R_i,S_i -> T_i)
Back Project Update C_g
P_g , C_g for all templates

Figure 7.6: Retina Tracker algorithm processing single video frame.

In order to achieve real-time performance and account for scaling caused by zooming the microscope, a Gaussian pyramid is implemented. I.e. for each template (g), each candidate match

(j) in the template's local search area is first matched in the coarsest scale, then that candidate position is refined consequently at higher resolution scales [6]. For each template translation (P_g) there is a corresponding matching confidence which is used to improve robustness when matching in areas of minimal texture variation. The confidence (C_g) for each template match is calculated by:

$$C_g = \frac{max(C_{gj})}{\bar{C}_{gj}} \tag{7.1}$$

Template matching results are used as inputs in the iterative computation of the 2D rigid transformation from the image to the retinal map. The algorithm is shown graphically in Figure 7.6. First it starts processing in the coarsest scale and propagates the results toward finer resolutions. At each iteration, the following steps are executed:

(a) First, the average of motion for all visible templates g, weighted by their respective matching confidence (C_g) is used to determine the gross translation (P_i) relative to the reference templates' positions P_g.

(b) Next, the gross rotation is computed by averaging the rotation (R_i) of the new template locations about the new origin (P_i), again weighted by the confidence.

(c) Finally, the scale (magnification) S_i is computed by comparing the average distance of template locations from the origin of the visible subset of the templates on the retina map and the current image.

At the end of each iteration, the reference template grid locations are back projected using the new transform calculated in the beginning of this registration loop $(P_i, R_i, S_i \rightarrow T_i)$. These are compared with those locations found in the NCC matching step. The overall registration error

[6] Note: for each refinement, the extents of the search window are extended by one pixel (centered on previous scales match position.)

E_p is the sum of squared position errors divided by the number of visible templates. The loop terminates when the sum of template position errors (E_p) is below a predefined threshold e, which was chosen empirically to account for environmental conditions and retinal texture.

The algorithm favors contributions from templates that possess high template matching confidence and conform fairly closely to the regular grid. If the loop continues, the confidences (C_g) for each template are reduced in proportion to the corresponding registration error for that template $C'_g = C_g - E_g$. This effectively removes the outliers in a few iterations. If there are only a few high-confidence templates left, the registration step is reported as not reliable. This decoupled iterative method was found to be more reliable in practice than standard weighted least-squares. The outliers usually occur in areas where accurate image displacement cannot be easily established due to specularities, lack of texture, repetitive texture, slow color or shade gradients, occlusion caused by foreground objects, multiple translucent layers, etc. This also implies that any surgical instruments in the foreground are not considered in the frame-to-frame background motion estimation, making the proposed tracker compatible with intraocular interventions (see Figure 7.5B). In the case of stereo images, rotation and scale of the left and right retina tracker as well as their vertical disparity are constrained to be the same (averaged) at each iteration of the algorithm.

The typical use requires initial exploration of the retina without any instruments in the view to build a clean mosaicked map of the retina. A more sophisticated approach would avoid this step by including tool tracking results to remove the regions that contain the tool from the map building process. The tracking algorithm does not recover from loss of tracking, for example, when the light pipe is turned off and on. This requires an additional registration module such as the one developed by Richa, Vagvolgyi and Balicki et al. [195]. It uses SURF features [196] to build and update the map as it expands. RANSAC [197] is employed for matching features from the feature map to those on the current ROI, enabling tracking to be reinitialized in case of full occlusions.

Anecdotal use in phantoms and in vivo rabbit experiments has shown the tracker to work very well when the translations of the retina were smooth. The tracker tends to favor rich textures which are common in the human retina and is able to handle tool occlusions without affecting tracking performance. The increase in computing power and further code optimization will make the tracker even more robust by reducing inter-frame retina motions. Furthermore, the M-Scan experiment presented in Section 7.10 demonstrates the use of this tracker in a Behavior. The results presented consider the ROI translations to be less than $^1/_3$ of the ROI from the origin[7]. Larger translations create significant distortions due to the spherical nature of the retina being projected onto a planar model. Also lens orientation relative to the imaging axis of the microscope can produce significant image warping.

7.5 Tool Tracking

Visual tracking of the ophthalmology instruments enables a set of Behaviors that provide safety as well as new diagnostic capabilities described in this chapter. The location of the instrument relative to the retina can also be used as a pointer in an augmented reality application, e.g., as an input in surgical planning Behaviors or in visual communication for educational Behaviors. Visual tracking of ophthalmology instruments is challenging due to variability of the tool appearance from visual path aberrations, strong shadows, widely varying lighting conditions, and the reflective surface of the tracking target.

Only a few approaches have investigated visual tracking of instruments in retinal surgery.

[7] The center of the initial reference image ROI represents the origin of a retina map.

They can be roughly categorized into two approaches: color-based and geometry-based. In color-based approaches, tools are detected and tracked based on the color information in the images, and artificial markers are often employed. This method is used by Becker et al. [129, 198], where the tip of the surgical tools held by Micron are colored with contrasting paint colors. This approach is challenging due to concerns of bio-compatibility and sterilizability of color markers, change in tool function such as tool width, and also tool customization which is not always possible.

Geometry-based approaches rely on finding the best visual match for a predefined shape model of the tool. Pezzementi et al. [199] showed how this concept can be used to track articulated retinal instruments. A similar approach used in the *eyeSAW* system is presented by Richa and Balicki et al. [195, 200, 201]. It tracks retinal instruments in a variety of conditions, but does not require a complex appearance model. Like the OCT tracker, it operates on the ROI images and generates the tool pose with respect to the ROI of the retina in both left and right images. The algorithm is a template-based tracking method based on a simple appearance model of the tool and uses the sum of conditional variance (SCV) as a robust similarity measure for coping with illumination variations in the scene.

The tracking of a model template is written as the problem of finding the transformation parameters $\mathbf{p}$ that minimize the SCV:

$$\min_{\mathbf{p}} \sum_{\mathbf{x}} \left(I(w(\mathbf{x}, \mathbf{p})) - \hat{T}(\mathbf{x}) \right)^2 \tag{7.2}$$

where $w(\mathbf{x}, \mathbf{p})$ is a transformation function of parameters $\mathbf{p}$ that maps pixel positions $\mathbf{x}$ from the reference image $T(\mathbf{x})$ to the current image I from the microscope: $\mathbf{x} \mapsto w(\mathbf{x}, \mathbf{p})$. The term $\hat{T}$ is computed from the joint intensity distribution between the image I and the template T. During tracking, the expected image $\hat{T}$ is computed only once for every incoming frame using $I(w(\mathbf{x}, \mathbf{p}))$ from the previous tracking step. This step allows $\hat{T}$ to "adapt" to the current illumination conditions.

Compared to similarity measures used in retinal tool tracking such as Mutual Information [200], SCV has a significantly smaller computational complexity, which is desirable in real-time tracking applications.

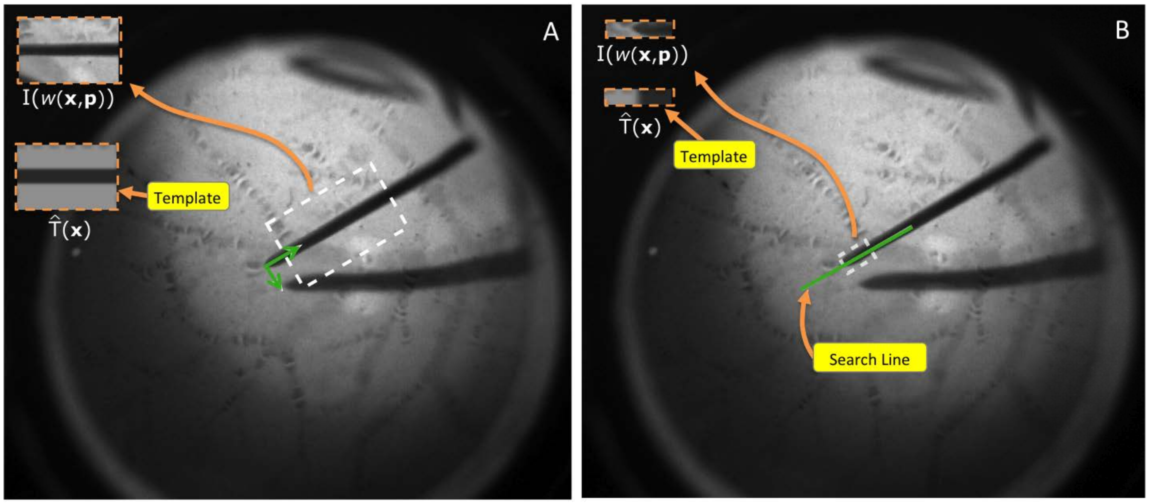


Figure 7.7: SCV-based tool tracking: A) first step to find the tool shaft. B) second step to find the tip location.

The tracking process involves estimating three transformation parameters ($\mathbf{p} = \{x, y, theta\}$) and is decoupled into two steps due to the linear geometry of the tracked tool. First, gradient-based tracking is used to estimate the tool rotation and the vertical translation of the tool shaft as shown in Figure 7.7A. In the second step, a different template is used to search along the tool shaft for the tool tip position as shown in Figure 7.7B. A brute force approach searches at discrete 0.25 px steps. The specific appearance models for the tool in each step above were chosen to represent a dark shaft on a bright background and were designed to avoid possible tracking "lock" onto specific background patterns.

The tracker is initialized in a semi-manual manner by positioning the tool in the center of the Region of Interest. The initialization step is rather cumbersome and should be eliminated by incorporating a fully autonomous tracker algorithm such as the ones proposed by Sznitman et al. [202, 203], which includes both detection and tracking.

7.6 OCT Tracking

Visual tracking of the single fiber intraocular OCT probe (Section 6.2) can be accomplished with visual tool tracking methods described in the previous section. This would require estimating the intersection point between the OCT tool and the retina surface, which is prone to optical distortion and tool axis detection errors. A novel alternative to the geometric approach is to track the visible spot created by the OCT light projecting on the surface[8], which provides the precise position of the OCT beam location on the retina, effectively correlating OCT data with the anatomy.

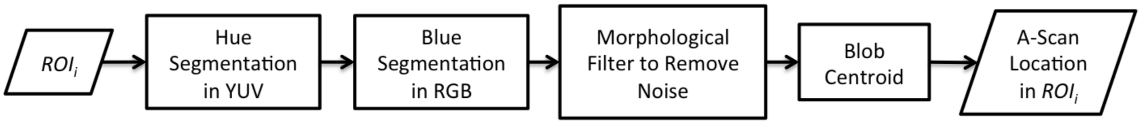


Figure 7.8: Flowchart of OCT projection visual tracking algorithm.

To facilitate robust tracking, a camera sensor is chosen that captures OCTs near-IR light predominantly on its blue RGB channel (see Figure 7.9), which facilitates segmentation as blue hues are uncommon in the retina. The tracking algorithm[9](see Figure 7.8) begins by thresholding the ROI in YUV color space to detect the blue patch. The area around this patch is then further segmented using adaptive histogram thresholding (AHT) on the blue RGB channel. Morphological operations are then used to remove noise from the binary image. This two-step process eliminates

[8] OCT tracker was developed in collaboration with Rogerio Richa.

[9] The OCT tracking algorithm is an alternative to the segmentation method developed by Yang et al. [204] and used by Brian Becker [205] to track the location of the aiming beam in laser ablation.

false detection of the bright light pipe and specular reflections on the metal shafts, and also accounts for common illumination variability from the handheld light pipe and auto-adjustable white balance on the cameras. The location of the A-Scan is assumed to be at the centroid of this segmented blob. Initial detection is executed on the whole ROI while subsequent inter-frame tracking is performed within a small search window centered on the previous result. Left and right image tracker results are constrained to lie on the same image scan line. The algorithm is very efficient, processing two ROIs in less than a millisecond, while running in parallel with other visualization components on a multi-core CPU.

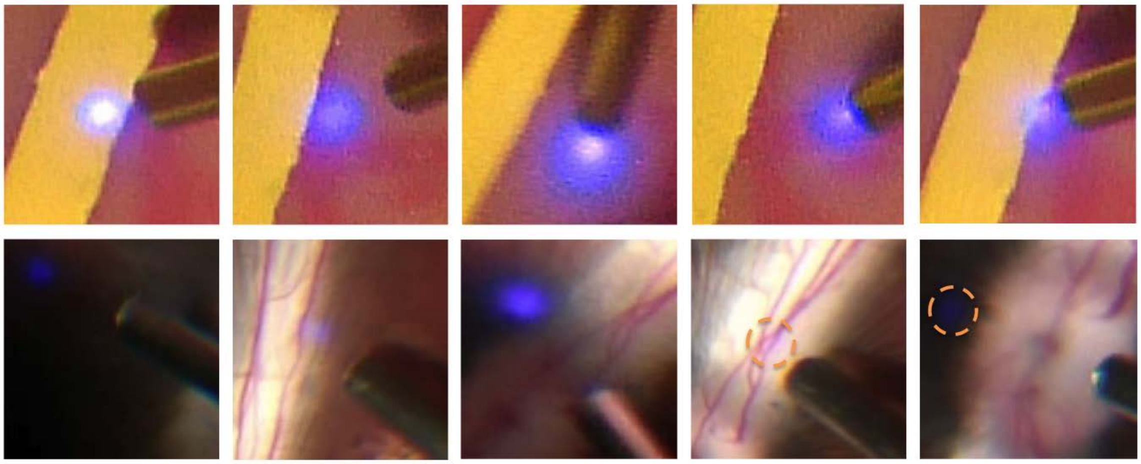


Figure 7.9: Examples of OCT light projections showing high variability in appearance from illumination, tool obstruction, and background type. Top row – eye phantom retina; bottom row – rabbit retina.

To independently validate the OCT tracker, 100 image frames from the M-Scan in the Eye Phantom experiment videos (Section 7.10) were randomly chosen, see Figure 7.9. The position of the OCT projection was manually segmented in each frame and compared to the OCT tracking algorithm results. The tracking error is 2.2 ± 1.74 px which translates to roughly 40 μm. For comparison, the pronounced bright area is between 10 and 15 px in diameter. Sources of this error can be attributed to manual segmentation variability, sensor saturation from limited dynamic range of the camera, as well as OCT projection occlusions by the tool tip when the tool was closer than $\sim 500\mu$m to the retina. With the inevitable development in digital camera technologies that provide higher resolution and higher dynamic range, these errors will decrease. Furthermore, by using focusing optics on the probe, such as the ball lens presented by Zhao et al. [161], it is possible to extend the imaging range of the OCT, which would avoid most occlusions from the probe's proximity to the surface. The focusing lens could also improve tracking precision by decreasing the OCT projection spot size, generating a brighter and sharper visual target, which is easier to segment.

The algorithm has been demonstrated in rabbit eye experiments (see Figure 7.9). It works quite well in peripheral regions that are visually similar to the human retina but tends to generate some false positives when the imaging over a rabbit's highly reflective (white) myelinated streak. In those cases, the blue-RGB channel processing step needs to be adjusted to consider only the pixels that are not completely saturated, i.e. the blue pixels that have corresponding red and green channels that are unsaturated.

7.7 Overlays

The lack of computer-integrated visualization systems for microsurgery has been the major hindrance to augmenting microscopy for eye surgery. Computer integration enables the rendering and fusing of graphic primitives (overlays) in the surgical display. One of the earliest attempts is by Berger et al. [187, 188], who built a prototype of a slitlamp-based video injection system that aligns previously-stored fundus images with the microscope view, and fuses them directly into the microscope eyepieces. Although prohibitively slow, it provides near real-time comparison of the biomicroscopic fundus view in human subjects. In a bench-top experiment, Becker et al. [198, 205] used rectangle overlays to depict targets for laser photocoagulation in a simulated surgical environment. The same system was used in a circle-tracing experiment that compared freehand tracing to Micron assistance. These trials required the operator to be aware of the limiting workspace extents of the device, which was communicated with multiple circle overlays which were displayed at the tip of the surgical instrument, and depicted the arc that the instrument could be actively moved by the Micron3D [120]. The Micron3D experiments were performed on a setup that relied on computational stereo reconstruction, which cannot be simply applied in vitreoretinal microscopy due to dynamic lens distortions.

The *eyeSAW* visualization subsystem provides many basic graphical primitives such as image overlays, lines, text, intensity bars, etc. that are static or dynamic graphical "widgets" which can be drawn anywhere on the raw image, and often have common attributes like transformation, color, opacity, width, etc. These can be combined with information from the visual tracking subsystems and other sensors in the *eyeSAW* environment to create Behaviors. In some cases, sensor information may be very visual in nature (e.g., OCT images) and also may have a spatial aspect (e.g., location of the OCT data on the retina), where visual presentation or visual sensory substitution may be more appropriate than the audio sensory substitution described in Section 5.3. Visual sensory substitution can also be complemented with audio feedback for a more effective communication.

When designing a Behavior that incorporates a visual overlay, it is imperative to present information minimally and clearly, and to specifically tailor it for a given task to minimize the surgeon's cognitive load. In some scenarios, the information should be located as close as possible to the surgical work area to minimize the eye gaze travel while also avoiding any possibility of obstructing the surgeon's direct view of the tools and the tissue being operated on. Some surgeons may prefer to forgo any potentially distracting graphics near the surgical area, and in exchange, position them on the periphery of the display, so that the information can be referenced when needed by a quick translation of the gaze. Non-critical information, such as system status (time, recording status, vitrector settings, etc.) or patient status (temperature, heart rate, etc) should be displayed minimally in the corners of the display, or, as a last resort, dynamically adjusted based on available screen real-estate.

In the current implementation, an overlay object is rendered on both left and right images of the stereo display. By setting the same overlay position on both images, the overlay object appears at the depth of the video-microscope's focal plane and as a result minimizes the surgeon's gaze adjustment. Some overlays are "attached" to a location on the retina or the tool tip in which case their left-right position should account for the stereo disparity to match the visual depth of the object. The left/right position of a feature like the tool tip is typically provided by the tool tracker and stored in the Transformations Manager.

7.8 Force Overlay Behaviors

In current practice, surgeons indirectly assess the relative stress applied to tissue via visual interpretation of changing light reflections from deforming tissue. This type of natural "visual sensory substitution" requires significant experience and concentration, and is common to only expert surgeons. In earlier chapters this skill was augmented via audio sensory substitution (see Section 5.3) incorporating real-time sensing information from smart tools and communicating it aurally. Graphical force sensory substitution is has also been shown to be an effective mode of communicating tool-to-tissue force information in minimally invasive robotic surgery. Kitagawa et al. [134] presented a visual overlay in the master console of the da Vinci system that used resizing color bars, where height and shade changed according to the measured tension at the patient-side robot manipulator, which was outfitted with strain gauges. Using the same system, Reiley et al. [206] showed that a color-changing dot that is located virtually on the surgical instrument can effectively communicate the forces measured at the instrument tip. The color indicated "safe", "OK", and "dangerous" states. Gwilliam et al. [207] combined both of these ideas and created a color bar graph that is overlaid on the visually tracked tool-tip of the daVinci instrument with force sensing capability. These concepts have been shown to restore some ability for the surgeons to "feel" forces in minimally invasive surgery. However, due to the lack of force-sensing instruments and sufficient visualization technologies, these paradigms haven't been investigated for eye surgery applications.

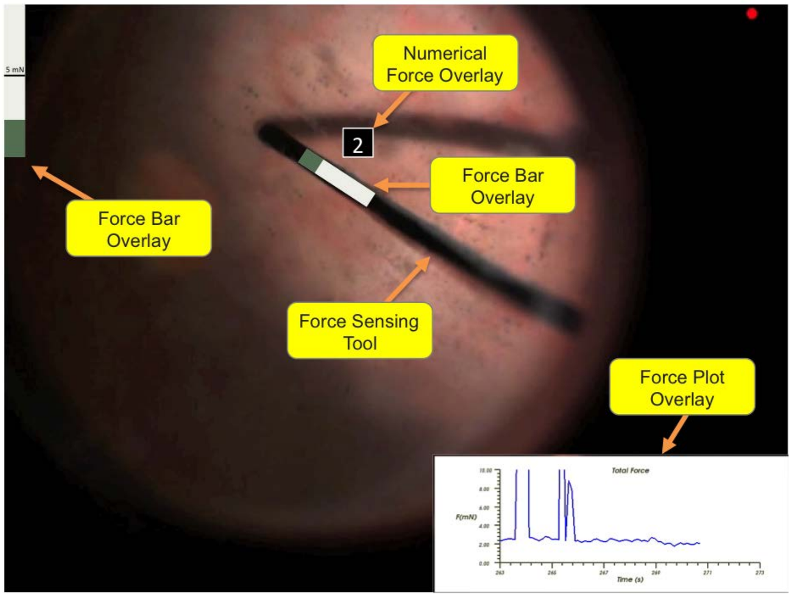


Figure 7.10: Various Force Overlay Behaviors.

The Force Overlay Behavior provides a set of visual sensory substitution methods to communicate the forces measured by the micro-force sensing instruments described in Section 5.1. The force bar overlay concept presented by Kitagawa et al. [134] has been implemented and used in real-time feedback. It is typically positioned away from the surgical area and represents the magnitude of the force at the tool tip. It is rendered on both left/right images in the display coordinate frame with zero disparity. As an alternative, the force bar can be overlaid on top of the force sensing surgical instruments. This is similar to the concept presented by Gwilliam et al. [207]. The tool orientation and tip position generated by the tool tracker are used as an input to render the overlay on corresponding left and right images. This ensures visually natural 3D depth of the widget and presents the information directly on the tool, near the surgical site. Additionally, the surgeon can enable a numerical force overlay that can be statically positioned anywhere on the display or can travel with the tool-tip similarly to the force bar. This is a simple box with a number representing the current force. Since the forces tend to vary rapidly, the numerical display should be updated periodically (e.g., 1 Hz) and display the largest force measured during that period.

One of the more effective visual force feedback methods is the scrolling force X-Y plot, where the x-axis is time and y-axis represents the magnitude of the force measured at the tool tip, as shown in Figure 7.10. The easily readable 10-second force history provided by the force plot has been indispensable during in vivo experiments to assess the forces required to tear the rabbit retina. The X-Y force plot image is also rendered in another process (force sensing module) located on a remote machine where computing resources are abundant. It is sent over the network at 20 Hz, and displayed with the Image Overlay widget. The Force Overlay Behavior controls the value, location, colors and sizing of the overlays, as well as the data ranges to be displayed.

Currently, the surgeon naturally integrates the force magnitude, presented visually, with the observed motion of the tool, which is assumed to be directly related to the force vector. It is possible to design more sophisticated multi-DOF overlays to represent the direction of the force, such as a 3D arrow. However, this would require the registration of tool orientation to the surgeon's point of view, which can be done either automatically (e.g., computer vision) or would need to rely on the surgeon to determine the relationship between the force-tool coordinate system (e.g., tool orientation) and the display's coordinate system.

Anecdotally, the visual force feedback provides exact quantitative information in a passive way. I.e. surgeons could access the visual information when they deemed it necessary by looking over at one of the overlays. In comparison, the audio sensory substitution was effective in communicating that some force is being applied by the instrument, generating a general "feel" of its magnitude. Some surgeons may benefit from the choice of available feedback methods when both of these paradigms are enabled.

7.9 H-Scan Overlay Behavior

One of the simplest but most effective overlays, especially for the OCT subsystem development, has been the A-Scan history display or the H-Scan. The Behavior simply overlays a constant feed of images from the OCT H-Scan generator. These are created by laterally stacking 1 to 10 seconds of A-Scans, creating an image, then zooming, cropping, resizing, (e.g., 240 × 512 px), adjusting brightness and contrast, adding depth gradations, and applying lossless compression for fast network transmission. The H-Scan Generator runs on the OCT machine to maximize the computational resources for the Visualization process, and also to minimize the network bandwidth that would be required to transfer thousands of A-Scans per second. The H-Scan image is regenerated at a preset rate (e.g., 20 Hz) and sent as an event to the H-Scan Behavior, where it is rendered as an image overlay on both left and right channels of the view. This has an effect of a scrolling OCT signal with the horizontal axis representing time, and the vertical axis representing the A-Scan depth as shown in Figure 7.11.

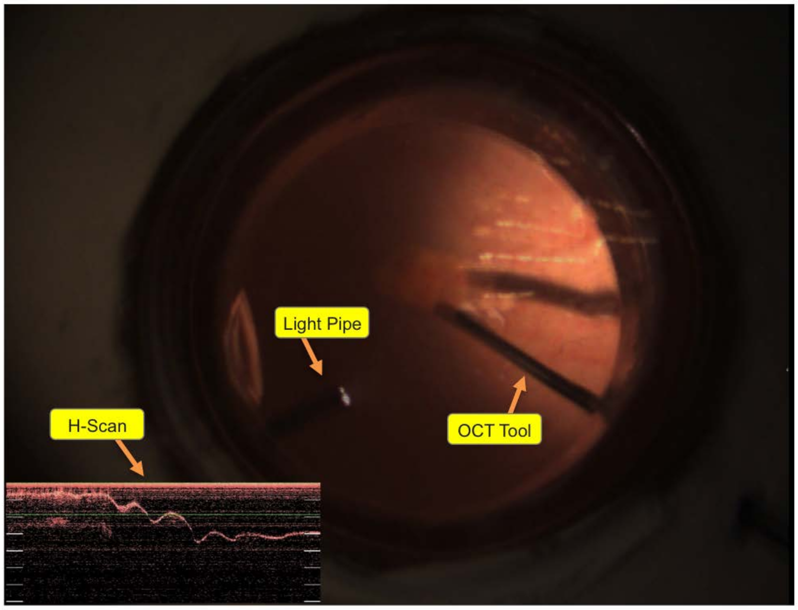


Figure 7.11: Screen capture showing the H-Scan Behavior in an Eye Phantom setup with OCT tool interrogating the retina insert.

This overlay Behavior controls various visual aspects of the H-Scan. For example, in the case of diagnostics, the surgeon might want to zoom in on the closest 1 mm of the data (top of the H-Scan) to see finer detail of the retina layers.

7.10 M-Scan Behavior

The surgical removal of an Epiretinal Membrane (ERM) (see Section 1.5.1) involves identifying, or, if necessary, creating an "edge" that is then grasped and peeled. In a typical procedure, the surgeon uses a stereo-microscope, a vitrectomy system, and an intraocular light guide to completely remove the vitreous from the eye to access the retina. Then, to locate the transparent ERM and identify a potential target edge, the surgeon relies on a combination of pre-operative fundus and OCT images, direct visualization (often enhanced by coloring dyes), and mechanical perturbation in a trial-and-error technique [29]. However, many dyes are considered toxic, and trial-and-error methods can cause further damage to the delicate retina. Once an edge is located, various tools can be employed, such as forceps or a pick, to engage and delaminate the membrane from the retina while avoiding damage to the retina itself. It is imperative that all of the ERM is removed, which can be millimeters in diameter, often requiring a number of peels in a single procedure.

Localizing candidate peeling edges is difficult. Surgeons must rely on inconsistent and inadequate preoperative imaging due to a number of factors, including developing pathology, visual occlusion, tissue swelling, and other direct or indirect effects of the surgical intervention. Furthermore, precision membrane peeling is performed under very high magnification, allowing the surgeon to visualize only a small area of the retina ($\sim 5-15\%$) at any one time. This requires the surgeon to mentally register sparse visual anatomical landmarks with information from preoperative images, and also consider any changes in retinal architecture due to the operation itself.

There are a few prior works that strive to provide visual navigation capability for identifying anatomical landmarks on the retina during posterior segment eye surgery. They rely on micron-resolution imaging provided by OCT. This process can be used to image the cross-section of the retina to visualize ERMs, which appear as thin, highly reflective bands anterior to the retina. A system built by Ehlers et al. [157] involves the use of a surgical microscope with integrated volumetric OCT imaging capability. Their system, though, is prohibitively slow; requires ideal optical quality of the cornea and lens; and most importantly, lacks a unified display, requiring the surgeon to look away from the surgical field to examine the OCT image. This increases the risk of inadvertent collision between tools and delicate inner eye structures. Fleming et al. proposed registering preoperative OCT annotated fundus images with intraoperative microscope images to aid in identifying ERM edges [153]. However, they did not present a method to easily inspect the OCT information during a surgical task. It is also unclear whether preoperative images would prove useful if the interval between their acquisition and the surgery permits advancement of the ERM. Other relevant work uses OCT scanning with probes capable of real-time volumetric images [156], but these are still too large and impractical for clinical applications, and do not address how to present the data in real time. Conceptually close works in other medical domains [208–210] have not been applied to retinal surgery, and all, except for [210], rely on computational stereo that is very difficult to achieve in vitreoretinal surgery due to the complicated optical path, narrow depth of field, extreme image distortions, and complex illumination conditions.

To address this gap in the field, the M-Scan Behavior has been developed to provide intraoperative imaging of retinal anatomy for diagnostics and surgical navigation, with ERM peeling as one of its driving applications. The concept was first presented by Balicki et al. [189] and is summarized in this section. The intraocular guidance capability leverages functions provided by the *eyeSAW* environment, while combining the intraocular OCT capability presented in Chapter 6 with a graphic visualization system (tracking, overlays, etc.). The Behavior allows a vitreoretinal surgeon to directly image cross-sections of the retina using a single-fiber OCT probe (see 6.2.1), and then to inspect these tomographic scans interactively, at any time, using a surgical tool as a pointer[10]. The locations of these "M-Scans" are registered and superimposed on a 3D view of the retina.

In an edge finding task, the M-Scan is used in the following manner. The surgeon inserts the OCT probe into the eye through a trocar, so that the tip of the instrument is positioned close to the retina and provides sufficient tissue imaging depth. The surgeon presses a foot pedal

[10] Any tool can be used to review the scans, including the OCT probe itself.

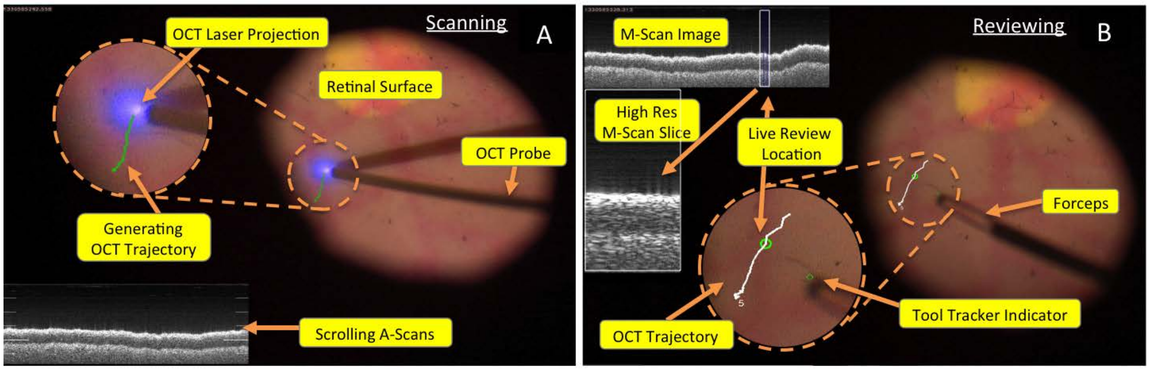


Figure 7.12: A) Creating M-Scan with OCT probe. B) Review mode with forceps as input. Each M-Scan has a unique ID which is located near the start of the trajectory.

while translating the probe across a region of interest. Concurrently, the visualization subsystem is tracking the trajectory of the OCT relative to the retina in the video, using the using OCT projection tracker (see Section 7.6) and the retina tracker (see Section 7.4). At the same time, the OCT M-Scan component running on the OCT machine is recording A-Scans and generating a 2D cross-sectional image. Once the scan is finished, the image is compressed (locally) and sent along with a thumbnail image and A-Scan acquisition timestamps to the visual M-Scan Behavior running in the visualization process. The Behavior combines the tracking results stored in a central Transformations Manager and provides the annotation logic to display the M-Scan and current tool locations. The resulting visual M-Scan is illustrated in Figure 7.12A. The location of these M-Scans is internally annotated on a global retina map, and then projected in stereo onto the current view of the retina. The surgeon reviews the scan by pointing a tool at a spot on the M-Scan trajectory where the corresponding high-resolution section of the OCT image is displayed, see Figure 7.12B. This is achieved by calculating the closest point on the path to the tool position that is accessed through the Transformations Manager. The timestamp associated with that point (interpolated if between frame timestamps) is used to indicate the corresponding location on the M-Scan thumbnail overlay. The review mode is active until a new task is selected, a command to clear display is given or a new M-Scan is started.

The M-Scan Behavior has been demonstrated in a simulated ERM imaging and navigation task. The system setup is shown in Figure 7.1 with the details of the video microscope found in Section 3.2. For the simulation, a realistic eye phantom described in Section 3.3.2 was used. It has a thin, multi-layer latex insert with hand painted vascular patterns, approximating the retina. These vascular details are coarser than those found in the human retina; although not as "good" as the human retina's finer textures, they are still sufficient for visual tracker development. For an independently verifiable ground truth ERM model, a $\sim$1 mm sliver of yellow, 60 μm thick, polyester insulation tape was adhered to the surface of the retina. It is clearly visible in the video images and its OCT image shows high reflectivity in comparison with the less intense latex background layers.

To assess the overall accuracy of the system, 15 M-Scans were performed in the following manner: the area near the tape was first explored by translating the eye to build an internal map of the retina. Then, the OCT probe was inserted into the eye and an M-Scan was performed at constant speed with a trajectory shown in Figure 7.13A. To facilitate analysis, a special review mode was implemented where a computer mouse input was used to select the location of the edges of the tape in the OCT thumbnail. While hovering over a particular place on the OCT thumbnail, the corresponding A-Scan location was automatically highlighted on the scan trajectory, i.e. a circle would appear on the trajectory path. The captured video of the display was then manually post-processed to extract the pixel location of the tape edge for comparison with the location inferred from the M-Scan review.

The average overall localization error, which included retina and OCT tracking, was 5.16 ± 5.14 px for the 30 scanned edges analyzed. Considering an average magnification level[11] in this

[11] The magnification levels in standard operations are higher than what was used in the experiment. This is

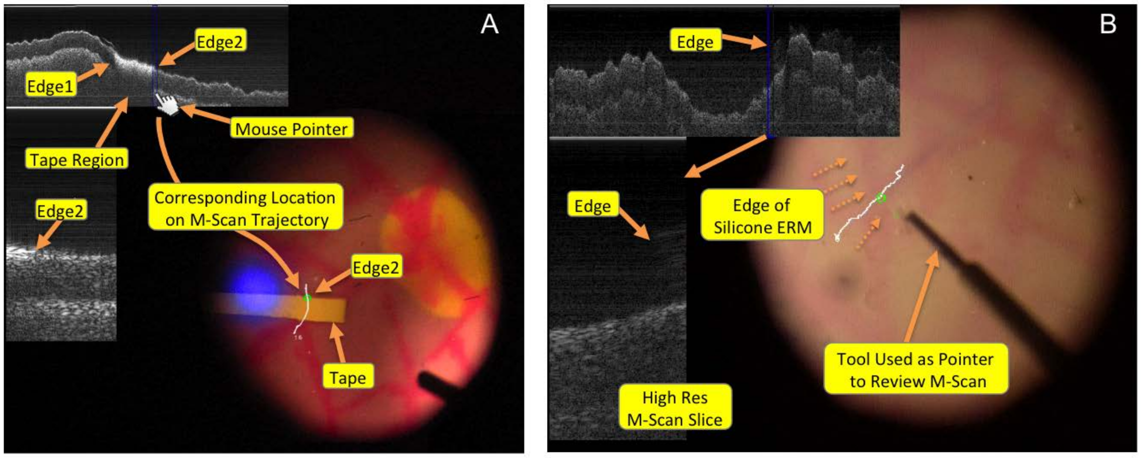


Figure 7.13: M-Scan in Eye Phantom: A) with tape simulating ERM used for validation of overall tracking B) with silicone layer (invisible ERM) demonstrating more realistic surgical scenario. The surgeon uses the forceps as a pointer to review the M-Scan. The green circle is the projection of the pointer on the M-Scan path and corresponds to the location of the blue line on the OCT image and the zoomed-in high-resolution OCT "slice" image on the left.

experiment, this error is equivalent to ∼100 ±100μm, using the tape width (∼55 px = 1 mm) as a reference. These accuracy values are within the functional range for a peeling application where the lateral size of target structures, such as ERM cavities, can be hundreds of microns wide, and the surgeons are approaching their physiological limits of precise freehand micro-manipulation [130].

Part of the error can be attributed to the OCT tracking (2.2 ± 1.74 px, see Section 7.6), which is sensitive to OCT projection occlusions by the tool itself[12] and can also be impaired by mediocre dynamic range and resolution of the camera. By developing a new OCT probe to image at greater distances from the retina (e.g., 3 mm) the OCT projection occlusion problem will be minimized, and will result in improved OCT tracking precision. It will also decrease the risk of accidental collisions with the retina.

The other major components of the overall tracking errors were due to the retina tracker

expected to improve the accuracy. E.g., doubling the magnification should reduce the error by roughly half.

[12] For a typical tool orientation and location, OCT projections are partially occluded when the tool was closer than ∼500 μm to the retina.

using the planar retina model, and to optical distortions at the periphery of the contact lens. Since the retinal model does not account for retinal curvature, the background tracker is only reliable when the translations of the retina are smaller than $^1/_3$ of the ROI size. I.e. the largest errors were observed when the scan path was reviewed from a significantly different point of view than the original scanning view. One simple solution to this problem would be increasing microscope magnification level or using a higher power contact lens. Going forward, a higher order retina model will be required to further reduce these errors.

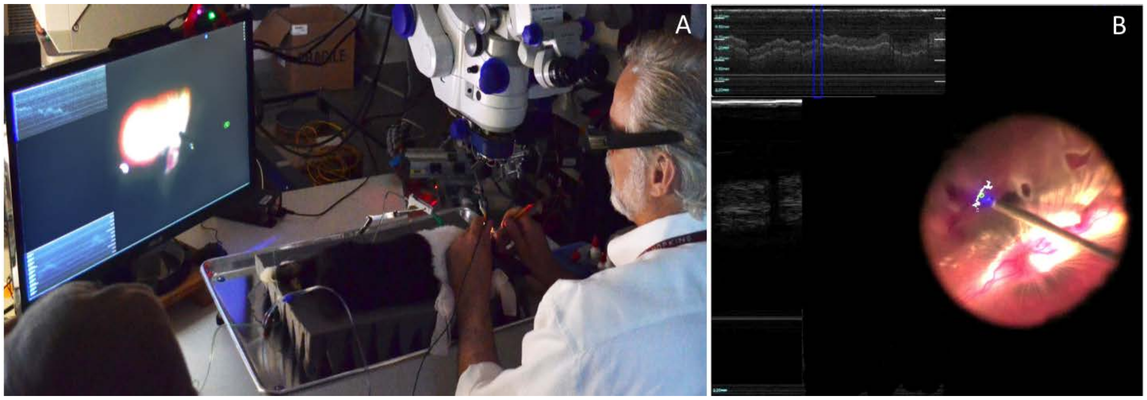


Figure 7.14: In vivo OCT M-Scan experiments. A) Operating room setup with a rabbit. B) Screenshot of an in vivo M-Scan in a rabbit eye.

Additionally, for the purpose of a more realistic demonstration, a thin layer of 100% silicone adhesive (Household Adhesive Sealant; DAP, Inc.) was placed on the surface of the retina to simulate a scenario where an ERM is difficult to visualize directly. Figure 7.13B shows the enface image of the invisible membrane and the corresponding M-Scan disclosing its cross-sectional structure. The location of the edges is easily determined by tracing the Forceps at the M-Scan trajectory in the M-Scan review mode. Furthermore, preliminary in vivo experiments on rabbits (see Figure 7.14) are very encouraging, with similar behavior of the trackers to the eye phantom. These demonstrations show how a surgeon can use the M-Scan capability to determine the extents of the ERM. Besides locating target peeling edges, the M-Scan capability allows continual monitoring of the state of the retina throughout the procedure, e.g., detecting complications, or for assessing completeness of the peeling process.

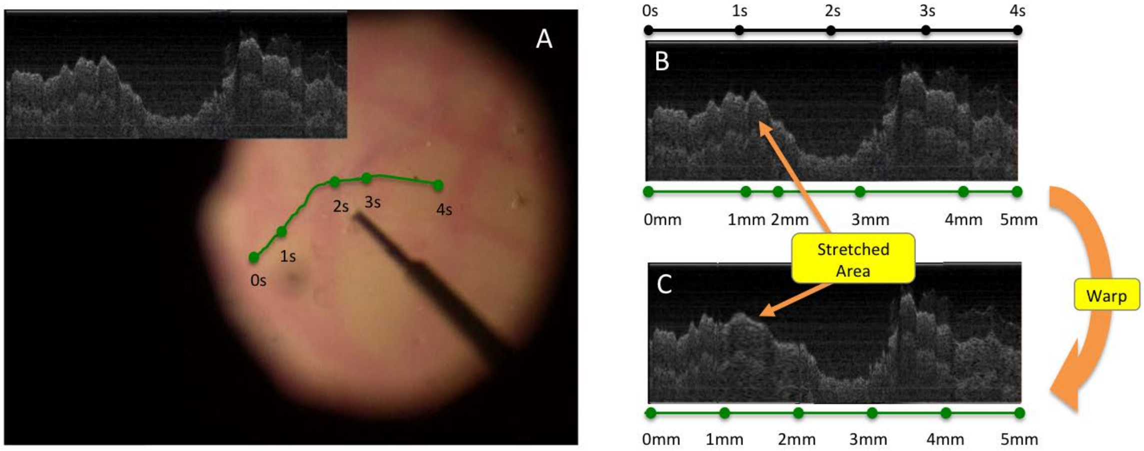


Figure 7.15: A) Screenshot of a simulated M-Scan with the timestamps along the OCT spot trajectory. Notice the variable speed. B) The M-Scan thumbnail with timestamps (in seconds) and corresponding distance along the path (in mm). C) Warped M-Scan thumbnail that accounts for speed variability creating a normalized x-axis based on distance traveled rather than time.

The current prototype does not include retina map registration between tracking sessions, e.g., when the light is turned on and off, the M-Scans associated with the internal retina map are not associated with the current retina map. This can be addressed with a feature-based registration of the two retinal images (e.g., retina map-to-video frame, or map-to-map) using methods proposed by Richa et al. [195] or Shen and Stewart et al. [193]. Another improvement is OCT image enhancement that aligns A-Scans vertically, removing the visible hand tremor effects. The signal processing method demonstrated by Huang and Liu et al. [160] will generate smoother OCT cross-section images which are more intuitive to interpret. This can also be addressed with a robotic assistant, an aid that naturally reduces hand tremor, or through active control of the distance between the OCT probe and the retinal surface (see Section 6.5.2). The robotic assistant can also be incorporated to perform semi-automatic B-Scans (see Section 6.5.4) that can be naturally combined with the visual M-Scan Behavior.

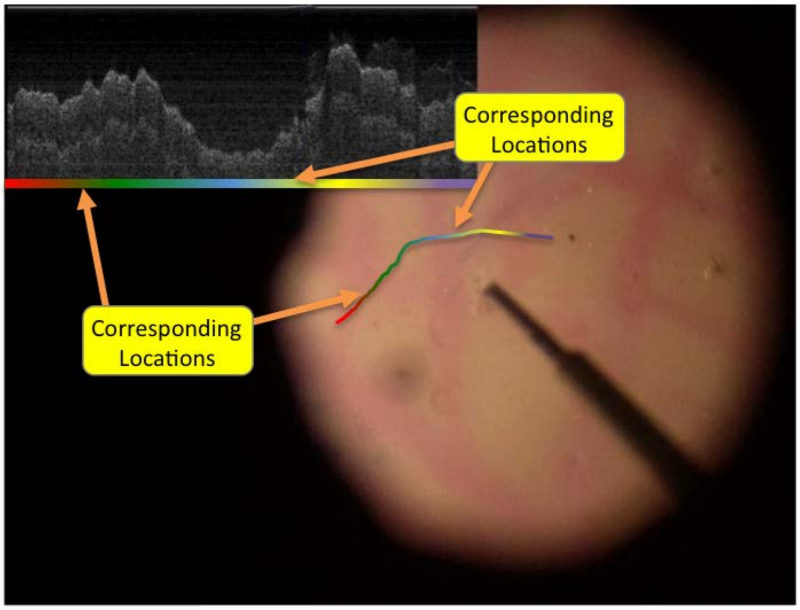


Figure 7.16: Alternative review method for the M-Scan Behavior where the scan trajectory is color coded to facilitate the M-Scan interpretation. Each color on the trajectory is mapped to a particular A-Scan with the same color in the bar below the thumbnail.

Another improvement involves converting the M-Scan x-axis units from time to mm to correct for scan speed variability[13]. This can be done by interpolating the OCT image laterally based on the M-Scan trajectory segment lengths. Figure 7.15 shows how the trajectory generated with variable tool speed is used to warp the thumbnail image so that the horizontal pixels are the same physical width and correspond to distance along the trajectory[14].

To improve M-Scan interpretation without the need for pointing devices, an intuitive correlation between the A-scan location along the scan trajectory and the corresponding positions

[13] In practice extracting trajectory length in mm may be difficult. Using retinal map pixels is sufficient to correct the M-Scan OCT image.

[14] The segments associated with a particular section of the M-Scan thumbnail are synchronized through system-wide timestamping.

on the 2D M-Scan thumbnail image is required. Figure 7.16 shows how that could be achieved by color coding the positions on the trajectory[15]. A color bar under the M-Scan thumbnail would be used to match the A-Scan to the location where it was captured.

7.11 Proximity Detection

Another common surgical skill that requires significant time to master is depth perception derived from shadow cues cast by the tool on the retina. This skill requires a particular use of the light pipe, as well as correct interpretation of the light pipe's resulting shadows, which correlate the distance of the tool to the tissue. In operations where the surgical instrument is not outfitted with an OCT range-sensing function (see Section 6.3), computer vision-based tracking of the tool tip position relative to the retina surface could be used to alert the surgeon of a possible collision. By leveraging the integrated functionality of the *eyeSAW* modules, such capability is possible and has been presented by Richa and Balicki et al. [200, 201][16].

Standard computational stereo algorithms are difficult to apply in eye surgery, where lens distortion, extreme magnification, and constant zoom alteration are common. An effective alternative uses the difference between tool and retina stereo disparities to estimate the proximity between the surgical tools and the delicate intraocular structures. Such a novel approach avoids the complex problem of calibrating the microscope's cameras and identifying constantly changing lens model parameters.

The proximity detection outlined here comprises two modules: tool tracking and a retina disparity map. The tool tracking module described in Section 7.5 provides the tool tip position

[15] Thanks to Dr. Handa for suggesting to color code the trajectory path.

[16] The general concept was developed with Rogerio Richa and Balazs Vagvolgyi. Richa devised the SCV based tracking methods. Experiments were designed and performed jointly with Rogerio Richa.

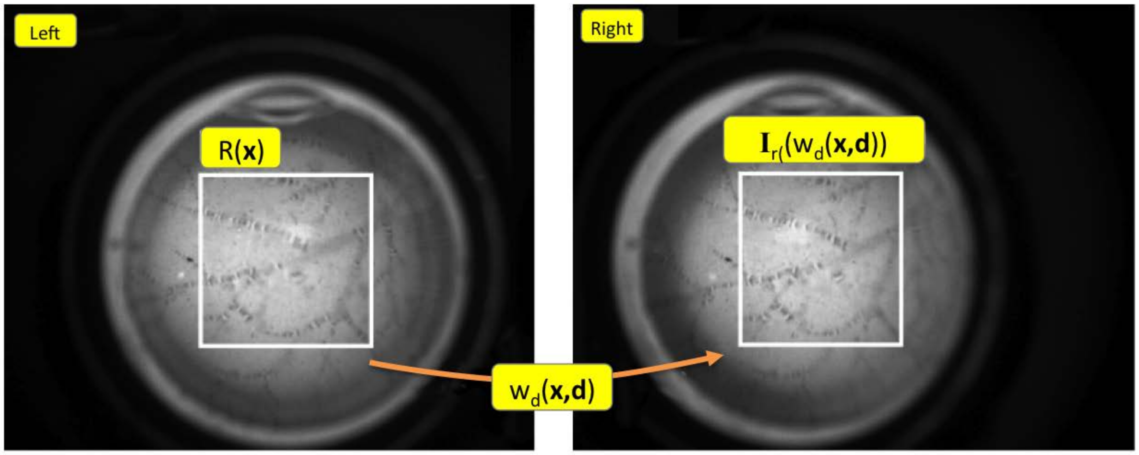

Left
Right
R(x)
$\mathbf{I}_{r(}(w_d(\mathbf{x},\mathbf{d}))$
$w_d(\mathbf{x},\mathbf{d})$

Figure 7.17: Disparity mapping by tracking a rectangular patch in the left and right images.

in the left and right images. The disparity value for the tool is simply the difference in position of the tool tip along the horizontal axis. The retina disparity map is another template matching task that estimates 1D horizontal displacement of a small rectangular patch between the left and right images of the retina. The Sum of Squared Differences (SSD) is used as a similarity measure between images and the minimization problem below[17] is solved using the Efficient Second-Order Minimization (ESM). The objective is to estimate the horizontal disparity that best aligns the selected template R to the right camera image I_r.

$$\min_{\mathbf{p}} \sum_{\mathbf{x}} (R(\mathbf{x}) - I_r(w_d(\mathbf{x}, \mathbf{d}))^2 \tag{7.3}$$

Only the central region of the ROI is considered where the estimation of the retinal surface disparity is not severely affected by lens distortions. Richa et al. [201] tested disparity maps with more degrees of freedom and found them to be prone to errors with increased DOFs. In practice, the 1 DOF model is sufficient due to the shallow depth of focus of the microscope and other disturbances

[17] $w_d(\mathbf{x}, \mathbf{d})$ is a horizontal warping function. See Richa et al. [201]for more information.

that cause a loss of visual information. Furthermore, surgeons often place a contact lens on top of the patient's eye during surgery for a magnified view of the retina, which significantly narrows the field of view.

Since the surgical tool has a different stereo disparity than the retinal surface, the corresponding tool pixels on both stereo images must be detected and removed from the retina disparity map calculations. Such an "occlusion map" is created using the results from the tool tracking. Furthermore, a way to improve the precision and the speed of the retina map disparity is to only consider the small patch around the surgical instrument, instead of the central region of the view. This would provide the most locally precise retina disparity measurement and avoid large searches.

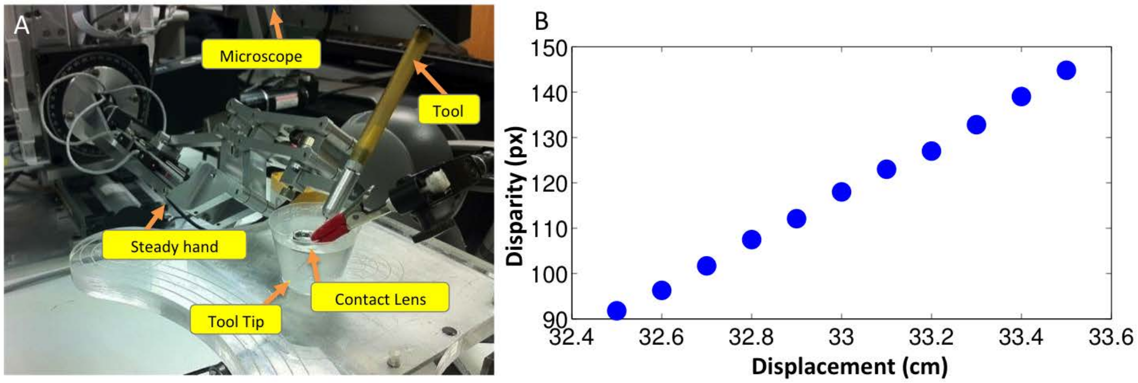


Figure 7.18: Calibration for visual proximity detection. A) Setup using the EyeRobot2. The tool tip is in water below the contact lens. B) Proximity calibration results showing a linear relationship between disparity and tool depth.

Proximity between the surgical tools and the retinal surface can be detected when the difference between tool and retina disparities is small (below a certain threshold). One way to determine that threshold is to manually calibrate the disparities in vivo by bringing the tool tip within a minimum safe distance from the retina and noting the disparity. In general this is not a practical approach. Another, more effective approach is to approximate the conversion between pixel disparity and physical depth using an external tracker or a robotic assistant [18]. To demonstrate a simple calibration procedure, EyeRobot2 was commanded to move along its Z axis in 1000 μm increments in a simulated environment shown in Figure 7.18A. The stereo disparity for the tool tip was tracked and plotted against the corresponding tool depth in Figure 7.18B. This relationship is linear and yields about 188 μm / pixel[19]. A more practical method would collect robot position data and compare it with the disparity measurements continuously during regular operation. This would refine the calibration and automatically account for any magnification or distortions.

In a practical demonstration with the Eye Phantom, the proximity detector was set to warn the operator when the difference between the tool tip and retina disparity fell below 10 pixels (approximately 2 mm). A force-sensing instrument (see Section 5.1) was used to detect any contact with the retina. The corresponding disparity and force plots are shown in Figure 7.19. The non-zero force events corresponding to deliberate contact with the retina are within the warning zone provided by the proximity detection.

There are a number of Behaviors that can communicate a possibility of collision with the retina based on the real-time visual proximity sensing. The simplest Behavior presents a continuous monotonic audio beep that sounds when the tool tip is too close to the retina (e.g., 10 pixel disparity). A more sophisticated approach, similar to the OCT range sensing feedback presented in Section 6.4, is to communicate two or three proximity distances with easily distinguished audio tracks. In theory, it is possible to reuse the OCT range-sensory substitution Behavior by swapping the range sensing inputs. However, this may require additional functionality such as an online

[18] The diameter of the surgical tool shaft can be used as reference for an approximated conversion between pixels and millimeters, this reference for conversion only holds for each image individually since the baseline for the stereo cameras is unknown.

[19] The calibration method assumes that the tool tip has negligible thickness, otherwise the thickness of the instrument tip needs to be incorporated into the proximity measurements.

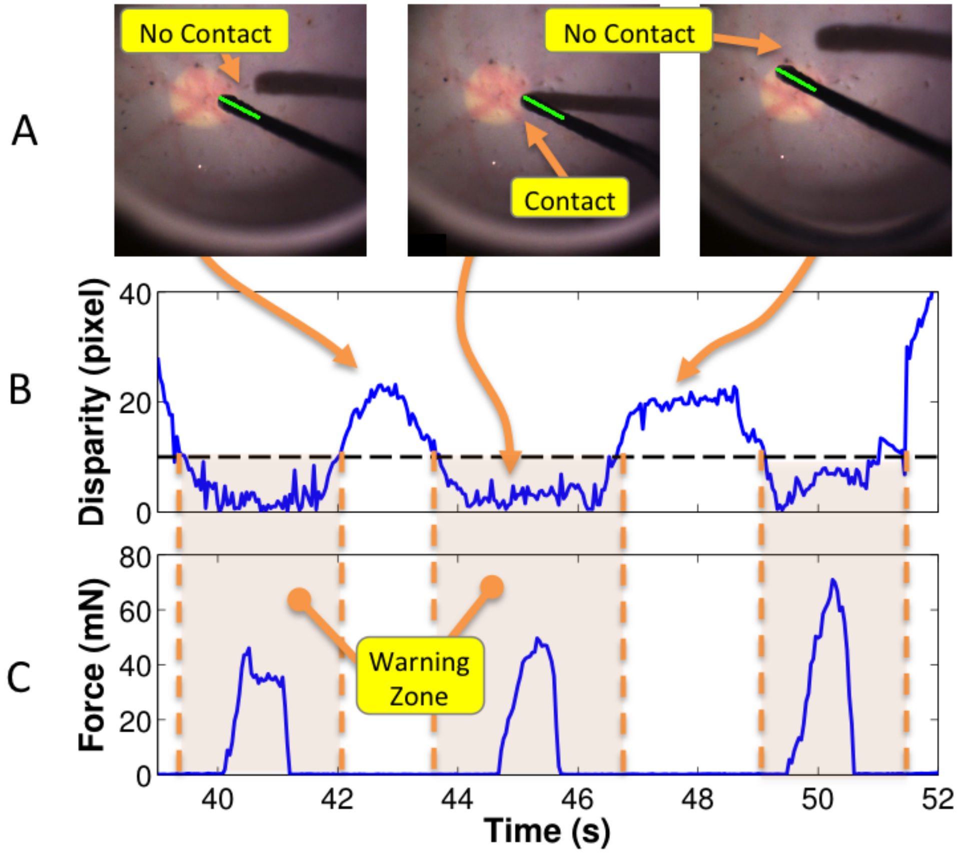


Figure 7.19: A) Screenshots of tool tip approaching surface in the Eye Phantom. B) The difference between the retina and tool tip disparities. Proximity warning is below 10 pixels (∼ 2 mm). C) Corresponding force sensor measurements showing contact with surface.

calibration routine, conversion from pixels to μm, or incorporation of the tracking resolution and tracking confidence in the presentation of range data. It is up to the system designer to determine whether a new audio range feedback Behavior is necessary or if an existing one (OCT) can be altered.

An alternative method uses a visual sensory substitution. In the simplest case, the sensory substitution Behavior is a square box overlay that appears near the tool tip whenever the proximity threshold is triggered. Another approach is a bar overlay similar to the force overlay bar in Section 7.8, where a rectangular gage is overlaid over the tool shaft[20]. The gage (percent filled) corresponds to the distance of the tip to the surface. Other methods include a basic numeric display near the tool tip or on the side of the screen. Haptic approaches can also be implemented. These include virtual fixtures with the EyeRobots and vibratory alarms attached to various parts of the surgeon's body. It is important to note that the system designer should consider the visual competition for a surgeon's focus, and not use too similar widgets for giving a variety of feedback. Avoiding any possibility of confusion is crucial in providing safe surgical assistance.

7.12 Tele-Visualization

The tele-robotic setup described in Section 4.9 may include a master console (e.g., da Vinci master) that is located in another room or even another continent. To ensure a near transparent experience, the system requires a stereo visualization with low latency, high frame rate, high resolution, and high dynamic range, in addition to intuitive and low-latency robot control. For long range telecasts, using existing communication infrastructure, such as Ethernet as a data transfer technology for its simplicity and ubiquity is desirable. The remote visualization aspect of

[20] Sufficiently away from the tip not to interfere with the visual access.

tele-microsurgery to enable tele-robotic eye surgery has been investigated and was presented by Balicki et al. in [125]. It is summarized here.

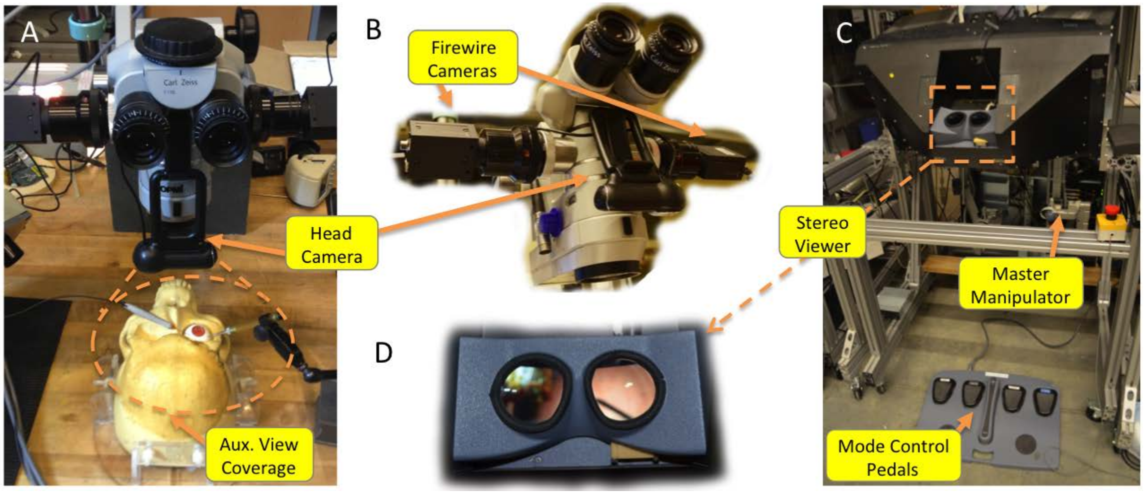


Figure 7.20: Tele-visualization setup: A) Head viewing camera mounted on the video-microscope. Ring shows the view region covered by the head view camera. B) Close-up of the head view camera. C) Da Vinci master console. D) Close-up of the da Vinci stereo viewer ports.

The system uses the standard da Vinci master console (see Figure 4.19), which has two CRT display monitors, at 1024 × 768 pixels per video channel. The display console itself is very ergonomic, allowing the optimal alignment of visual and motor axes, and offering excellent stereopsis. The visualization hardware on the slave side (see Section 3.1) comprises a standard ophthalmological surgery stereo-microscope (Zeiss OPMI-MD) outfitted with two IEEE-1394B capture cameras (Grasshopper by Point Grey Research), each capturing 1024 × 768 pixels resolution images at 45 FPS. The cameras are configured specifically to capture in lower resolution to match the master side displays. The left and right video streams are rendered locally on a 3D Display. During the development process, it became clear that an additional viewpoint is necessary to cover the visualization of any activity outside of the eye, e.g., monitoring the insertion of the instruments into the trocars. To address this, a small USB camera (640 × 480 px, 15 FPS) was mounted on the microscope facing down, and aligned so that the viewer had a wide view of the patient's head, as shown in Figure 7.20. A Picture-in-Picture (PnP) Behavior was designed to provide an auxiliary view (see Figure 7.21) so the remotely-located surgeon is able to simultaneously view the stereo feeds from the microscope and from the head-viewing camera. The PnP feed is rendered as overlays on both left and right video frames. This allows for additional stereoscopic view when stereo cameras are used. The Behavior allows for custom placement and sizing of the auxiliary view, and provides the ability to swap between the video sources for the primary and auxiliary views.

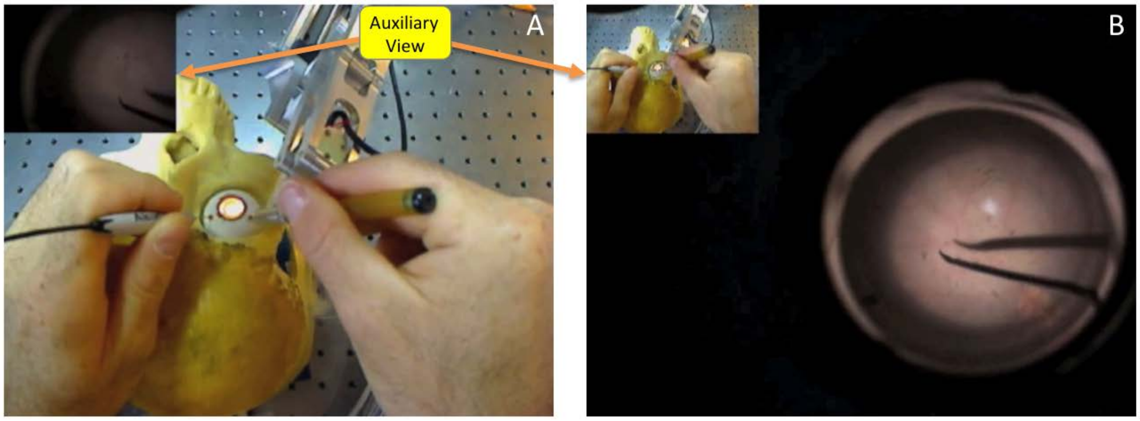


Figure 7.21: A) Main view displays the video feed of the patient's head from the head camera mounted on the microscope, while the auxiliary view displays the intraocular stereo feed from the microscope. B) Video feeds are swapped.

The visualization software architecture built with the *cisstStereoVision* library (SVL) (see Section 7.1) is extended to provide remote viewing capability[21]. The major addition is the head-view camera stream that is incorporated as an overlay allowing simultaneous display of both input streams on the display. Figure 7.22 shows the processing steps that effectively enable display mirroring.

Once all the video streams are color-corrected, magnified, resized, split-off for tracking,

[21] The SVL library has been developed by Balazs Vagvolgyi. The Tele-visualization software architecture was designed in a collaboration with Balazs Vagvolgyi.

etc., they are combined into a single stereo pair with the desired overlays. These images are then split into smaller sub-images and each one is encoded with either no compression, lossy (jpeg), or a lossless (DEFLATE [211] via zlib library) compression algorithm. The encoding is the most resource-intensive step, so significant gain in speed is achieved through parallelization, where each compression task (sub-image compression) is assigned to a different processor on a multi-core system. The intra-frame encoding was chosen to minimize latency. The uncompressed version is a good option when ample network resources are available, such as a dedicated 1-Gbit Ethernet connection. The encoded sub-images are serially sent over the Ethernet network to the master console using TCP/IP, and in the cases where the system is on a dedicated network, the UDP/IP protocol used as packet loss is insignificant and packet order is sufficiently consistent to minimize image frame losses.

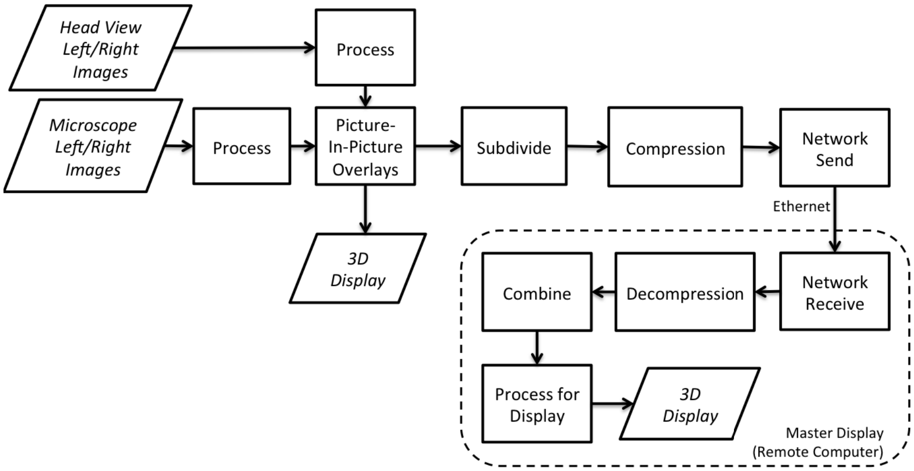


Figure 7.22: Tele-visualization process flowchart.

The remote (master) display system buffers incoming encoded sub-images until all are received and the process proceeds in reverse, i.e. the images are decoded (decompressed) and combined into stereo pairs, then formatted and rendered for the 3D display. Any incomplete frames are discarded if a newer frame is received. A pedal interface is used as input into the Picture-in-Picture Behavior to switch between the different auxiliary display modes.

The prototype system[22] used a switched 1-Gb Ethernet network to transfer the dual XGA progressive video stream with a fairly low, approximately 6% (16x) average (jpeg - quality value 85) compression ratio in order to keep the quality loss introduced by the compression close to the source image SNR. This yielded a 44 FPS remote display with network bandwidth utilization of 10–15 MB/s[23]. The local stereo display ran at 45 FPS (1024 × 768 px). The same setup ran at 20 FPS when lossless or no compression were used, with 35 MB/s and 65 MB/s network bandwidth usage, respectively. The display latency for this setup is estimated to be 4 or 5 frames, which is around 100 ms. The system was able to simultaneously archive in lossless proprietary CVI format at 35 FPS.

Despite the latency, the tele-visualization system has sufficient frame rate for the purpose of tele-robotic surgery where robot motions are relatively slow and stable. However, the latency will still need to be improved to provide < 50 ms response, which is especially challenging when used with higher-resolution cameras. Future consumer-grade hardware will have sufficient processing resources and bus speeds that will allow higher resolution video with lower latencies in transmitting the stereo video feed. When the master and slave are separated by great distances, the performance will be limited by the large network bandwidth and latency requirements for the JPEG-based intra-frame video encoding. Simultaneously, the network routing behavior should also allow for low-latency robot control data transfer. In any collaborative environment, audio communication is a must, and will need to be added to a practical implementation of this system.

[22] Prototype system did not include any visual trackers.

[23] MB/s is MBytes/s.

7.13 Telestration

For safety and ergonomic reasons, current intraocular surgery involves a single surgeon controlling hand-held instruments while his or her gaze is completely focused on the view inside the microscope eyepieces. This poses a challenge in educational scenarios where the trainee is controlling the instruments, leaving the expert surgeon to rely solely on verbal communication to convey complex instructions, e.g., describing a proper tool motion. Ideally, this exchange of information would be done at least in part visually, without switching surgical positions[24] or turning on the operating room lights.

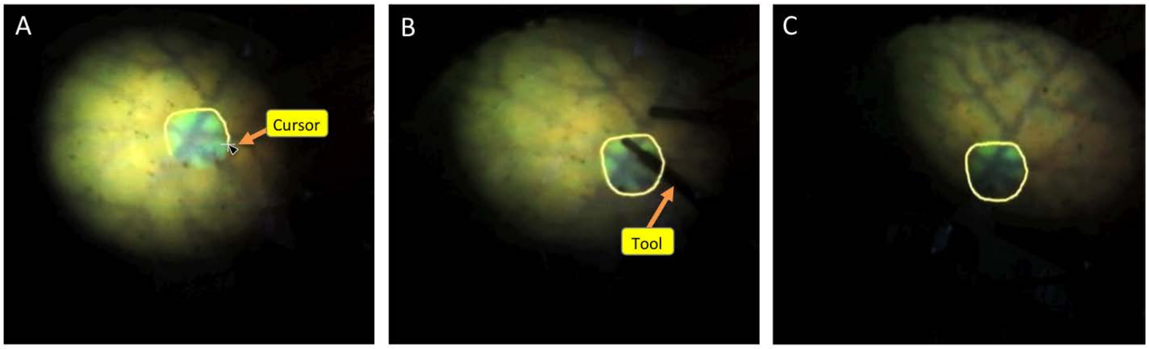


Figure 7.23: Telestration example: A)"freesketch" around the optic nerve. B) The annotation persists even when tools are obstructing the retina. C) The annotation travels with the retina when the viewport moves.

One of the *eyeSAW* capabilities that can aid in communication between an expert surgeon and his/her trainees is the Telestration Behavior, which provides a means for graphically annotating a surgeon's view through the microscope with drawings (e.g., line sketches). This is an increasingly common feature of tele-medicine systems and is also included in the da Vinci system. The novel version presented here uses a variety of user inputs to make annotations on the retina that appear

[24] Removing the tools from the eye and inserting them back in is time consuming. Also in many cases, the light pipe should be inside the eye to illuminate the retina for visibility.

in real time on the microscope display. The annotations made with this system persist virtually in their precise positions on the retina, and move as such with the retina on the display until they are turned off, or erased. Figure 7.23 shows a line annotation around the optic nerve of the Eye Phantom. It persists even when the tool is moved over the annotated area, and travels with the optic nerve when the view (eye) is translated. It is also visible even when partially obstructed by the iris.

The Telestration Behavior is based on drawing graphical primitives such as "freesketch" (set of connected line segments), straight lines, squares, and circles, using one of the many possible inputs that are transformed to the retina coordinate system. In the initial prototype, the Telestration Behavior is located on the same process as the main visualization engine, i.e. all the widgets and interaction feedback are processed in the main visualization Stream and directly rendered onto the microscope 3D display. This allows the Telestration Behavior to use the overlay primitives and the transformations between the retina and display coordinate frame directly (with direct memory access). This minimizes visual synchronization delays which have to be near real-time in augmented reality applications to reduce operator fatigue.

The user input components are required to provide the X,Y position of the cursor on the display, and are controlled with two input "buttons". A simple example is a standard computer mouse. Once the Telestration Behavior is enabled, the first button is used for drawing, i.e. when it is engaged (pressed and held down) a multi-segment line is created, corresponding to the list of positions generated by the mouse motion. These positions are then transformed into the retina coordinate system and rendered on the display over the retina. The problem of transformation from 2D input (screen coordinates) to the 3D visualization coordinates is solved by setting the position of the annotation in the left and right images to be the same at the time the annotation is created[25] This is based on the assumption that the microscope is focused on the retina and therefore the disparity between left and right retina patch is zero at the time the annotation is created. In practice, this works quite well since the microscope has a very shallow depth of field and needs to be focused on the area of interest to provide a clear view. When the cursor approaches an annotation, the annotation is highlighted (selected) and the second button can be used to erase it. Each annotation has a number (text overlay) next to it for referencing to further facilitate efficient communication.

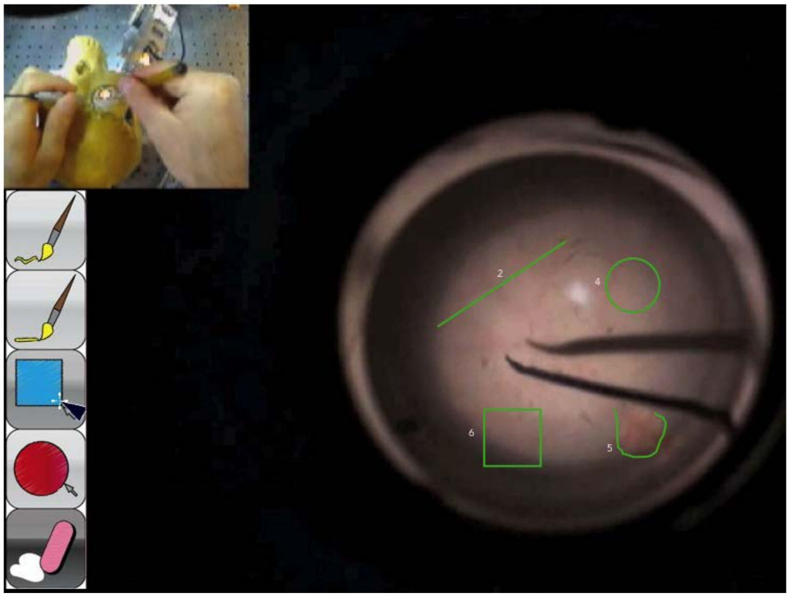


Figure 7.24: Telestration toolbar on the left is used to select the drawing widgets. Example widgets are line, freesketch, square, circle, and erase.

A more advanced interface has a set of drawing widgets[26], shown in Figure 7.24. The toolbar is hidden on the side of the screen and appears when the cursor approaches that side of the display. The operator can then move the cursor over the toolbar and click to choose the

[25] The location of annotations in the left and right image will change over time due to the zooming, rotations, translations, and disparity from focal plane changes.

[26] Thank you to CIS2 (The Johns Hopkins University, 600.466) course students, especially Robert Eisinger for programming the drawing widgets.

desired widget, e.g., a line, or a circle, that will be drawn on the retina. The initial implementation has widgets for straight straight lines, “freesketch”, rectangles, circles, and erasing. To erase, the operator selects the erase widget, and then moves the mouse pointer towards an annotation to be removed. The annotation is selected when it becomes highlighted, at which point it can be erased with a click. Since this multi-widget interface requires more user interaction, it is more applicable to remote telestration where a dedicated workstation can be used, such as the daVinci master in Figure 7.20C.

Besides the computer mouse, there are many other pointing devices that can be used as inputs to this Behavior. In the cases where the main surgeon needs to create an annotation on the fly, e.g., make a mark for future reference, the tool tracking module (see Section 7.5) is used to provide the position of the tool tip relative to the retina. The tool position can easily be accessed through the Transformations Manager (see Section 7.3). In such input scenarios, the tool is typically far away from the retina, so by using the average of the tool positions from the left and right tool tracking results, the annotation positions are projected onto the retina. Any of the readily available foot pedals are “wired-in” to provide the button events to control the drawing state. While not implemented, computer-vision-based detection of closing and opening of the forceps tool can be also be used as an input to control the drawing states.

The OCT tool tracking has also been used as an input into the Telestration Behavior. Since the tracker provides the exact stereo position of the pointer on the retina from the projection of the OCT beam onto the retina surface, the annotations are located exactly where the surgeon is pointing the tool. The OCT Safety Barrier Behavior can be used to minimized risk of collisions with the retina.

Although the use of surgical instruments as pointing devices inside the eye is an interesting approach, it is debatable if the practice can be safe enough for proper use by novice surgeons. One alternative that has been implemented incorporates the EyeRobot's Joystick Behavior (see Section 4.8), which provides cursor position and button events. This is considered safer since the robot can keep the tool effectively static while a portion of the instrument is still in the eye. In case of an emergency the tool can be rapidly retracted, as in normal robot operation.

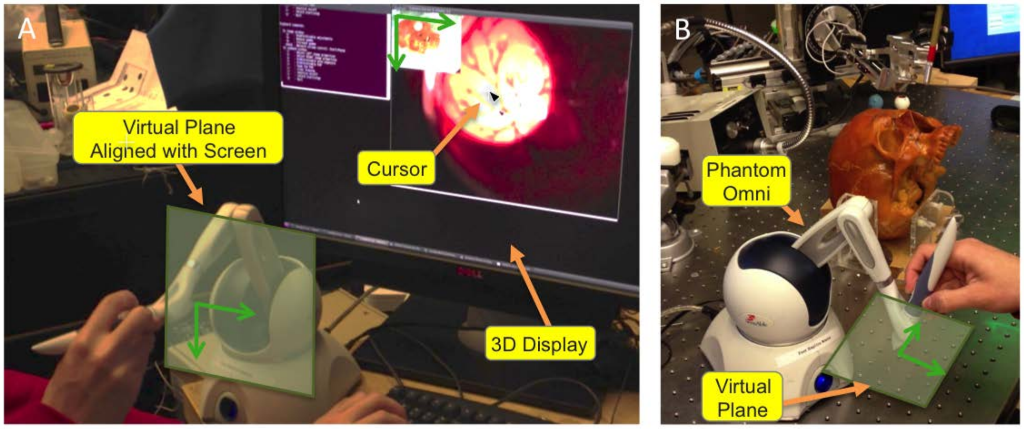


Figure 7.25: Remote telestration interface with a haptic device with plane virtual fixture aligned A) with the 3D display or B) with the patient.

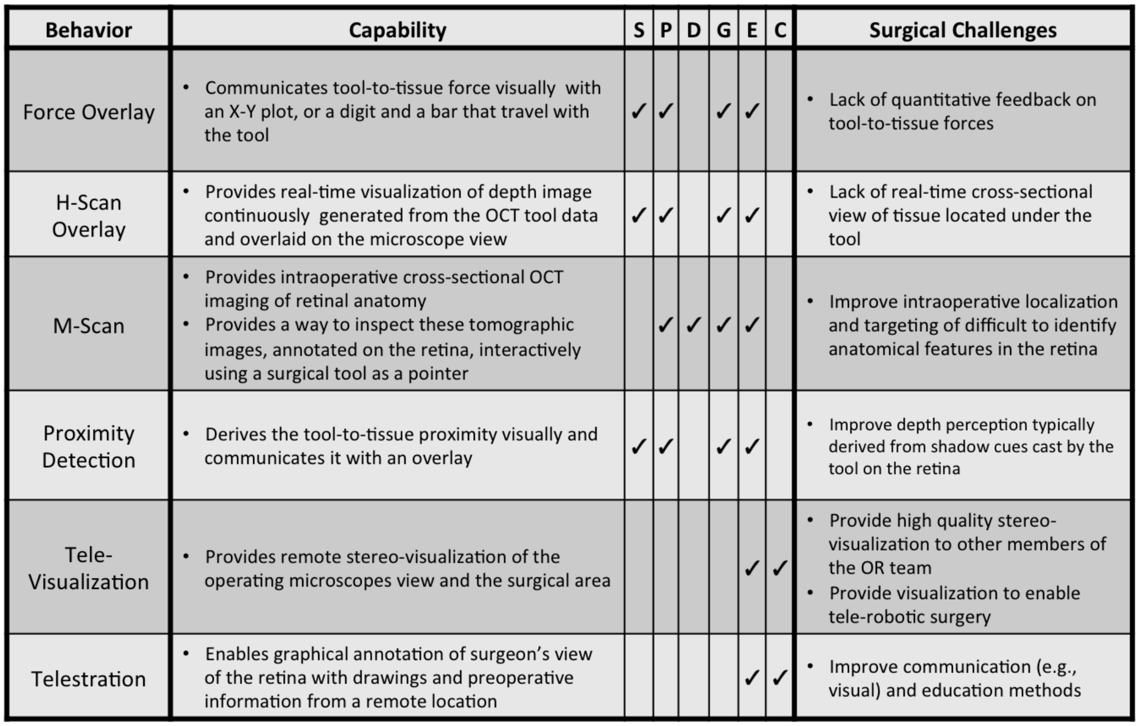
Another alternative is using an external haptic device such as the Phantom Omni (SensAble Technologies, Inc.) or the da Vinci master console. These devices are designed for 3D input in virtual simulations and in control of robotic manipulators. To facilitate X,Y motion, the end-effector of the devices is constrained to only move in a plane using a virtual fixture. The alignment of this plane should be intuitive to the operator. The most common arrangements that have been prototyped include a plane that is parallel to the display screen, or one that is parallel with the operating table. This is shown in Figure 7.25.

The rapid introduction of tablet computers (e.g., iPad, Apple Inc.) and smart phones into the operating room lends them naturally as interfaces to the Telestration Behavior. This would allow for a very intuitive and portable way to create graphical annotations. For this purpose, an alternate architecture could be implemented that prepares the telestration primitives remotely on the tablet (hidden from the surgeon), and then enables (visualizes) them on the surgical display when the annotation is complete. Furthermore, such non-critical auxiliary interfaces can include a mode for annotating a video frame, i.e. the remote video feed is paused, the annotations are made to the still frame, and then the image with the annotations is sent to the visualization sub-system to be incorporated into the view of the retina. This same technique can be used to annotate the fundus image, e.g., to prepare an operative plan, and then registered with the retina map during the operation to transfer the annotations on the live view of the retina.

The Telestration Behavior has an immediate potential to improve surgical practice by facilitating communication between collaborating surgeons. It can also quickly become a standard tool in the education of new surgeons. Furthermore, the geometric information (e.g., points on the retina) generated through the Telestration Behavior can be incorporated into vision-based virtual fixtures. For example, the ER2 Joystick can be used to create a circle annotation over a retinal vein. This location would then be used in the ER2-assisted cannulation Behavior to perform vision-based servoing to guide the pipette into the vessel.

7.14 Conclusions and Future Work

This chapter presented the visualization aspects of the *eyeSAW* system and related Behaviors that have been developed for vitreoretinal surgery applications. These Behaviors are summarized in Figure 7.26. Visualization subsystem provides basic stereoscopic microscope display, which is considered to be the most important aspect of the vitreoretinal surgery system because of its primary feedback function. Although the current version of the 3D video microscopy only



Behavior	Capability	S	P	D	G	E	C	Surgical Challenges
Force Overlay	• Communicates tool-to-tissue force visually with an X-Y plot, or a digit and a bar that travel with the tool	✓	✓		✓	✓		• Lack of quantitative feedback on tool-to-tissue forces
H-Scan Overlay	• Provides real-time visualization of depth image continuously generated from the OCT tool data and overlaid on the microscope view	✓	✓		✓	✓		• Lack of real-time cross-sectional view of tissue located under the tool
M-Scan	• Provides intraoperative cross-sectional OCT imaging of retinal anatomy • Provides a way to inspect these tomographic images, annotated on the retina, interactively using a surgical tool as a pointer		✓	✓	✓	✓		• Improve intraoperative localization and targeting of difficult to identify anatomical features in the retina
Proximity Detection	• Derives the tool-to-tissue proximity visually and communicates it with an overlay	✓	✓		✓	✓		• Improve depth perception typically derived from shadow cues cast by the tool on the retina
Tele-Visualization	• Provides remote stereo-visualization of the operating microscopes view and the surgical area					✓	✓	• Provide high quality stereo-visualization to other members of the OR team • Provide visualization to enable tele-robotic surgery
Telestration	• Enables graphical annotation of surgeon's view of the retina with drawings and preoperative information from a remote location					✓	✓	• Improve communication (e.g., visual) and education methods

Figure 7.26: List of visualization-based Behaviors with specific capabilities they provide, and the corresponding surgical challenges that they address. Behaviors are also characterized by capability types: S-Safety, P-Precision, D-Diagnostics, G-Guidance, E-Education, C-Communication.

approximates the performance of a conventional microscope (e.g., low latency, high resolution, high dynamic range, 3D perception), the rapid development of consumer electronics will greatly improve camera and display technology, and potentially enable video microscopy to surpass the quality provided the standard operating microscopes. There are many benefits of using video microscopy, e.g, it improves surgical ergonomics through the use a large mobile LCD display, and enables computer-vision based surgical assistance. This chapter presented many assistive Behaviors that augment the visual display to communicate intra-operative sensor data through overlays [189], provide educational capability with remote stereo-viewing and telestration [125], and facilitate intraocular image guidance and surgical planning [189, 195]. The following points describe the lessons learned during the visualization subsystem and Behavior development, and also the directions of future research:

- Any augmented reality feedback during a high-risk surgical maneuver has to be minimal, intuitive, and precise. Surgeons rely heavily on clear visual access so any additional visual information has to be unobtrusive and provide great benefit that outweighs the possible distraction or additional cognitive load imposed on the surgeon.

- Although slight video latency can be manageable (< 100 ms) by the surgeon, any temporal or spatial discrepancy between the overlays and the corresponding anatomy is extremely distracting and possibly disorienting. This issue needs to be considered in future hardware and software implementations.

- The software architecture of the visualization subsystem is agnostic to the type of cameras and displays, requiring minimal development to integrate new devices. This is a necessary design consideration to keep up with inevitable improvements in hardware, especially the electronics that will increase image resolution and bandwidth. Future software iterations should focus on improving tool and retina tracking, especially investigate using higher-order retina representation models. The Transformation Manager has been very useful in rapid prototyping by simplifying the access to complex system of spatial relationships. It should be redesigned with automatic transformation graph generator, include interpolation and extrapolation filters, and methods to notify components when a particular transformation path is updated by all the required or desired transformation dependencies.

- The 3D LCD displays are readily available and can be easily positioned to suit surgeon's preferences; however, future research should investigate direct image injection into the microscope viewer as presented by Berger at al. [188]. This could provide additional redundancy and also consider the surgeons who prefer the ergonomics of the standard surgical microscope. Furthermore, many of the visual Behaviors presented in this chapter are compatible with this alternative display method.

- The promising results of the M-Scan experiments demonstrated the system's novel function and its potential, and are the basis for a future multi-phantom, multi-subject study. This study will also provide qualitative data for the redesign of the user interface. The M-Scan is a prototype Behavior for intraocular localization and assessment of retinal anatomy by combining visual tracking and OCT imaging. The surgeon may use this functionality to locate peeling targets, as well as monitor the peeling process for detecting complications and assessing completeness, potentially reducing the risk of permanent retinal damage associated with membrane peeling. The Behavior can be easily extended to include other intraocular sensing instruments (e.g., force, ultrasound), can be used in the monitoring of procedures (e.g., laser ablation), and can incorporate preoperative imaging and planning.

- Visual Behaviors could greatly benefit from an intuitive interface such as a tablet, which would allow a surgeon educator to draw directly on the live view of the retina, where selected annotations are mirrored on the surgical display. Initial experience with the tablet interface has also been very practical in terms of controlling Behaviors and is likely to be the ideal Scenario Manager interface for future iterations of the project.

- The functionality provided by the visualization system can enable many new Behaviors. For example, a method similar to the one presented by Yamamoto et al. [208], shows a "heat map" of the cumulative forces that have been applied to particular areas of the retina. Such historical force data could indicate where most of the peeling was performed, and used in planning follow-up treatments. Another Behavior that is currently under the development is mapping the locations and time of intense surgical light exposure on the retina. Excessive light exposure is toxic, especially on the macula, and is a common complication for novice surgeons. This information could be used to automatically warn the surgeon of a possible overexposure.

- Another research direction could follow the early work by Becker at al. to incorporate vision-based feedback to virtual fixtures for robot assistance in targeted laser treatments [212], membrane peeling [120], and cannulation [129]. Another Behavior that could add a third-hand in a form of a robotic light pipe holder that works in unison with the surgeon who is controlling two surgical instruments, one in each hand. This would enable bimanual tissue manipulation while the light pipe is positioned automatically to illuminate the area near the surgical instruments.

- Finally, computer vision analysis of standard procedures could be used to evaluate surgical skill, provide surgical training feedback, or document essential aspects of a procedure, as demonstrated by Lin et al. [213] in robotic minimally invasive surgery. A vision-based approach could potentially provide the same benefits for vitreoretinal surgery and should be investigated in the future.

7.15 Recapitulation of Contributions

Visualization System (Section 7.1) This high performance stereo-visualization system includes low-latency 3D display pipeline, visual trackers, and augmented reality overlays. The novel software architecture is built using the SVL library, and focuses on efficient use of system resources for variable rate asynchronous processing of video "streams". It provides functionality to build visual Behaviors that augment the surgical display by presenting real-time information to the surgeon. The extendable and distributed architecture facilitates rapid development of augmented reality applications that integrate external sensors and computer vision algorithms. The visualization system is central to *eyeSAW* development and can be used to develop microsurgical applications that rely on stereo-microscope visualization. **Credit**: The *SVL* framework with visual trackers, filters, and overlay functionality was designed by Balazs Vagvolgyi. The *eyeSAW* visualization architecture was designed by Balazs Vagvolgyi and Marcin Balicki.

Video Latency Testing Device (Section 7.2) This custom, stand-alone device was developed to precisely measure overall latency of the visualization system as the surgeon would experience it. It generates a visual event in the field of view of the video-microscope, which is then detected directly on the 3D display. The time between the two events is the latency measured with sub-millisecond precision. This device has been used to test the latency of the video-microscope prototypes presented in this dissertation. Such a device can be used to test the latency of any camera and display system. **Credit**: Marcin Balicki.

Image Transformations Manager (Section 7.3) A centralized repository of all coordinate transformations in the visualization system was developed. The temporal granularity of the transformations is based on the frequency of the main display Stream. This facilitates more efficient and thread-safe read and writes. The design pattern can be applied to other transformation systems that have asynchronous updates but are based on data that is derived or associated with a high frequency source. **Credit**: Collaboration with Balazs Vagvolgyi.

OCT Tracking (Section 7.6) This alternative algorithm visually tracks the OCT beam projection on the retina instead of tracking the instrument shaft itself. This provides a very precise position of the OCT beam location on the retina, effectively correlating OCT data with the anatomy. The algorithm is fast and robust and can be used in other applications that require localization of OCT in video images. **Credit**: Marcin Balicki in collaboration with Rogerio Richa.

Augmented Reality with Visual Overlays (Section 7.7, 7.8, 7.9) This is the first demonstration of augmented reality for vitreoretinal surgery that uses stereo overlays with video-microscopy to display real-time intraocular sensor information. The overlays are dynamically created locally or remotely, and can be “attached” to objects (tools, retina) in the field of view. One example is communicating force at the tool tip which can be done with an X-Y plot, or a digit and a bar that travel with the tool. Overlay Behaviors can easily be extended to incorporate other sources of information, and to use the available widgets and rendering methods to present the information to the surgeon. **Credit**: General overlay system designed by Balazs Vagvolgyi. Specific overlay Behaviors are developed by Marcin Balicki.

M-Scan Behavior (Section 7.10) A new method for localizing difficult to identify anatomical features in the retina was developed using video stereo-microscopy and intraocular OCT instruments. To use this Behavior, the surgeon sweeps the OCT tool across a region of the retina while the system tracks the OCT beam and the retina to produce a cross-sectional image of the anatomy corresponding to that trajectory on the retina. The surgeon is then able to interrogate the OCT image by pointing a surgical instrument at the M-Scan trajectory superimposed on the retina and displayed in 3D. Visual M-Scan is an alternative intraoperative imaging method designed to increase surgical precision, and to minimize the surgeon's cognitive load by providing immediate and specific diagnostic information. **Credit**: Marcin Balicki.

Visual Proximity Detection (Section 7.11) A novel computer vision method was developed for detecting contact between surgical instruments and the retina without specialized proximity sensors. The method estimates the distance between the tool and the retina based on their relative stereo-pair disparities. When the difference in the disparities is below a certain threshold, collision is detected. This sensor-less approach can be used for collision warning in any stereo video-microscope applications that are not compatible with classic computational stereo methods. **Credit**: The general concept was developed by Marcin Balicki in collaboration with Rogerio Richa and Balazs Vagvolgyi. Rogerio Richa devised the SCV based tracking methods. Experiments were conducted jointly by Marcin Balicki and Rogerio Richa.

Tele-Visualization (Section 7.12) A remote visualization system was developed for vitreoretinal surgery, where a wide angle view of the surgical field of the patient's head is provided alongside the highly magnified stereo-microscope visualization. A picture-in-picture method for switching between the two views is also presented. The auxiliary patient view is essential for a practical tele-robotic system, since most of the procedures involve tool manipulation inside and outside of the eye. This concept can be applied to any micro-surgical tele-robotic system designed to operate on multiple visual scales. **Credit**: Concept developed by Marcin Balicki with implementation help from Balazs Vagvolgyi.

Telestration (Section 7.13) Demonstration of a novel concept for vitreoretinal surgery that provides the means to visually annotate the retina structures during an operation. Any user inputs into the system (tool tracker, haptic joystick, robot joystick) can be used to control basic drawing primitives to mark up the visible anatomy. These annotations are then attached virtually to the retina. This tool has great potential to improve communication in vitreoretinal surgery, especially for education and surgical staff collaboration. **Credit**: Developed in a collaboration with Balazs Vagvolgyi. User input methods developed by Marcin Balicki. Drawing GUI implemented by Robert Eisinger.

Chapter 8

Summary and Conclusions

This dissertation explored the partnership between a human, a computer, and other technologies to enhance human abilities to perform very demanding tasks, with vitreoretinal surgery as a driving application. The approach involved designing a novel computer-integrated augmentation environment with specific distributed architecture and system design methods. The resulting design guidelines with hardware and software infrastructure provided the necessary foundation for the development of innovative technical capabilities called Behaviors. A Behavior leverages the functionality of the integrated data-driven system by incorporating robotics, sensors, multimedia components, and information to provide a specific aid to the surgeon to address fundamental physiological and technical limitations in microsurgery. The system has been demonstrated in many in vivo operating scenarios and many of the Behaviors have been evaluated and shown to address the most challenging facets of vitreoretinal surgery.

8.1 Chapter Summaries

Chapter 1 introduced the anatomy of the eye and the most common conditions of the retina that affect vision, some of which can only be addressed with surgical treatments. A typical vitreoretinal surgery operation was presented, followed by two prototypical procedures that exhibit many technical and physiological challenges faced by surgeons in the following categories: safety, precision, diagnostics, guidance, education/training, and intra OR-communication. A set of Behaviors was proposed that complement the surgeon's skills in many of the tasks involved in the execution of the two procedures.

Chapter 2 presented the system and software engineering challenges associated with designing a distributed, fully integrated, multi-device, and multi-function system. A surgical system design methodology was introduced, which divides the surgical procedures into manageable tasks. The tasks are further analyzed to produce required or desirable surgical assistance capabilities. Each capability is realized through a novel system architecture design pattern called the *Behavior* design pattern. The Behavior incorporatesComponent Based Software Engineering principles to encapsulate particular system functionality and provide standard interfaces for centralized management via the Scenario Manager application. It was shown that by following the proposed design pattern, it is possible to systematically organize many Behaviors in an inherently complex system. The system can be dynamically reconfigured for each step in the procedure to present Behaviors with different operational parameters, or completely new Behaviors. The proposed software architecture can be used for any multi-component and distributed system that requires centralized configuration and control.

Chapter 3 presented several novel experimental testbeds used for validation of various Behaviors.

These included a liquid filled artificial eye model with interchangable retinas, a number of different membrane peeling and OCT imaging phantoms, etc. The testbeds have been indispensable in every step of the system and individual technology development, from surgical ergonomics to specific device functionality validation. Without these testbeds, it would be virtually impossible to accurately measure the performance of the system. This is especially true in the case of vitreoretinal surgery where the micron-level scale is inherently difficult to work with, and access to consistent biological samples is limited and time consuming, often requiring expert skill.

Chapter 3 also presented a novel software application to facilitate the analysis of voluminous multi-media data collected during subject and rabbit experiments. It is analogous to a standard movie editing tool, but with the ability to support any type of data, not just video and audio. The framework is easily extensible to include other proprietary data formats and types collected during experiments, e.g., force. This framework is generic and can be used in other research projects that have multiple timestamp-synchronized data streams.

Chapter 4 presented the development of the EyeRobot, a cooperatively controlled surgical robot, and associated Behaviors that focus on augmenting surgeons' ability to precisely and safely maneuver the instruments inside the eye, i.e., to minimize the effects of hand tremor, provide micron-scale tool positioning, and stabilize the retinal targets by limiting the effects of tool interaction with the sclera, etc.

A number of improvements to the robot design and control were presented, including quick release tool adapter, variable gain control, etc. Additionally, many novel robot control methods (Behaviors) and setups that are specific to vitreoretinal surgery were introduced. These include cooperative teleoperation of the EyeRobot, two-robot (bimanual) virtual fixtures, PseudoRCM constraint, virtual joystick, etc. A human subjects experiment showed that the EyeRobot provides significant assistance in a very challenging vein cannulation task by providing a stable and precise maneuverability of the pipette.

Chapter 5 presented a novel device developed by the *eyeBRP* team to measure micro-forces applied to eye tissues and Behaviors that use this real-time information to comply to the environment and to augment surgical performance by providing intuitive feedback aurally or haptically. These Behaviors address surgical challenges with application of milliNewton-level forces which is affected by physiological hand tremor, fatigue, deficient visual and kinesthetic feedback, patient movement, and poor accessibility due to the anatomy of the eye.

The force-sensing surgical instruments were first used to measure interaction forces during the manipulation of a rabbit eye retina and chicken egg membranes and also in vein cannulation in chicken embryos. These experiments showed that human-imperceptible manipulation tool-to-tissue forces can be measured in real-time with eye surgery-compatible instruments. The results were also used to design task-specific Behaviors to provide force information to the surgeons. These include audio sensory substitution methods and active robot-assisted guidance to reduce or minimize the force application in delicate tasks such as membrane peeling. Various novel robot and force feedback Behaviors were compared which showed that audio force feedback is a very effective and precise communication method, outperforming the robot-based Behaviors in most tests. Both types of Behaviors could operate simultaneously to provide optimum aural and manual assistance in membrane peeling.

A follow-up multi-subject study showed that regardless of surgical experience, the audio force feedback helped to reduce mean and large forces during a simulated peeling task. Additionally, the feedback from the surgeons was overwhelming in favor of introducing such technology into the OR, and could be especially beneficial in the training of vitreoretinal surgeons.

Chapter 6 presented Behaviors that rely on intraocular Optical Coherence Tomography (OCT) imaging to provide diagnostic information of the retina, as well as surgical safety and intervention assistance capabilities that leverage the integrated nature of the *eyeSAW* ecosystem. The OCT-based Behaviors address surgical challenges of deficient visual depth perception, lack of detailed information of the cross-sectional anatomy of the retina, the effects of natural hand tremor and inherent imprecise tool maneuvering.

A novel OCT integrated ophthalmic micro-pick was developed that enables simultaneous A-mode imaging and tissue manipulation. Since the instrument tip is visible in the OCT image, it is inherently self-registered. A complementary algorithm was presented that tracks the position of the tissue surface, ahead of the surgical part of the instrument.

The resulting real-time range information is used in an Audio OCT Feedback Behavior to warn the surgeon of a possible collision with intraocular tissue. In a Servo-to-Target Behavior, the OCT range functionality is used in robot-control to precisely and safely position the tool relative to the retina surface, e.g., for an incision. Another robot-based Behavior actively prevents unintended collisions with the retina by enforcing a virtual safety barrier.

Diagnostic scanning Behaviors were also presented, where the position of the single fiber OCT probe is manipulated by the robot in order to create planar and volumetric OCT images. Additionally, a robot-assisted B-Scan Behavior demonstrated how simultaneous range-sensing and scanning is combined to create automatic planar scans while keeping a constant tool-to-tissue distance, which improves image quality.

Many of these Behaviors are not limited to a particular type of a robotic assistant and were also demonstrated using a hand-held robotic manipulator, CMU Micron, that provided a very dynamic response (e.g., high speed OCT scans).

Chapter 6 also presents a novel method for automatic calibration of a spectrometer, the vital component of an FD-OCT system. The method leverages the precision and controllability of a robot to perform an on-the-fly calibration in a typical scanning environment. It improves over existing methods by not requiring any additional equipment beyond what is already available in the *eyeSAW* system.

Chapter 7 explores the capabilities provided by a custom video stereo microscopy system to augment visual feedback. Vitreoretinal surgeons rely primarily on vision from the ophthalmic microscope for surgical feedback during the operation. The vision-based Behaviors presented in this chapter are designed to address many surgical challenges which include inadequate depth perception, fusion of pre-operative images with the view of the retina, no structural or advanced diagnostic functionality that may be beneficial during the operations, very narrow field of view at high magnifications which leads to poor localization of anatomical targets, and the uncomfortable ergonomics of the ophthalmic microscope setup.

A high-performance microscope visualization subsystem was developed that provides capture, display, and video processing functionality of HD-quality stereo video imagery in real-time using off-the-shelf equipment. The visualization system includes modules for background (retina) tracking as well as trackers for conventional tool and OCT projection trackers which provide essential functionality for visualization-based Behaviors. It also includes a unique transformation manager module that serves as an efficient centralized repository for all coordinate transformations in the visualization system.

A number of implemented Behaviors provide visual sensory substitution that fuses intra-operative sensing information, virtually, in the field of view. Examples include tool-to-tissue force and the tool's proximity to tissue that are represented using text or colored intensity bars superimposed on the surgeon's view of the retina, either near the tip of the sensing tool or placed in a non-obstructive section of the view. Others simply display real-time X-Y plots from OCT or force sensing Components.

The Visual M-Scan Behavior demonstrates a novel application of a single fiber OCT system and the retina background and OCT projection trackers for localizing difficult to identify anatomical features in the retina. It is designed to increase surgical precision and to minimize the surgeon's cognitive load. The M-Scan offers an immediate and specific diagnostic information, i.e., cross-sectional images of the retina, for intraocular image guidance to aid in localizing retinal targets, e.g., epiretinal membranes.

The Televisualization Behavior provided a remote stereo-viewing capability, while the Telestration Behavior presented methods to visually annotate anatomical features from a remote location. Both of these have a great potential to improve the education of new surgeons, and become an integral part of a tele-robotic system.

The introduction of large stereo displays improves surgical ergonomics and provides high quality 3D visual access to the surgical site for the whole surgical team rather than just for the surgeon and an assistant, which is the case in conventional microscope setups.

The visualization latency of the system was quantified with a novel testing device and showed that the current implementation will still need performance improvements to be clinically viable. Additionally, the testing device can be applied to test the latency of any camera and display system.

8.2 Future Work

The surgical system design methodology and the Behavior software architecture design pattern presented in Chapter 2 generalize well for the development of human-centric, computer-integrated systems where one or more individuals are cooperating with the system on a multi-step task. By providing a combination of complementary capabilities that are dynamically marshaled for the particular needs of the operator and the task, the skill of the operator is effectively improved.

These concepts apply not only to ophthalmologic surgery but also to other microsurgical disciplines such as otology (e.g., cochlear implant surgery [68, 69]), neurosurgery, otolaryngology, and vascular surgery. Although medicine was the initial inspiration for the system, the concepts are also relevant beyond medicine, e.g., satellite servicing [70, 71], that comprise well defined, mission-critical tasks with inherent uncertainty that require human input, and rely on a computer-integrated system for assistance.

So far, the Behavior design pattern has only been tested in a simulated set of Behaviors and will need to be fully integrated into the existing code base to undergo formal testing as a complete system in a realistic scenario. The *eyeSAW* system implementation is a work in progress requiring additional engineering effort towards a fully clinical grade performance. Especially important, safety and robustness aspects of the system design were not fully investigated and should be revisited in the future iterations of the design.

A number of preliminary studies were conducted to assess the feasibility and performance of many Behaviors. Others were simply demonstrated and will require quantified assessment with multi-user studies as well as in vivo experiments to assess their safety and efficacy. The Behaviors presented in this dissertation are just a small subset of the many potential aids that could transform the practice of vitreoretinal surgery and other specialized surgical disciplines. The following are examples of Behaviors that should be investigated in the future:

Light Map - visualization-based Behavior that maps the cumulative intraoperative light exposure on the retina, creating a type of a "Heat Map". This information can be used by the surgeon to limit harmful overexposure by adjusting their surgical plan on-the-fly.

Force Application Map - similarly to the Light Map Behavior, a map is created on-the-fly that integrates the location and magnitude of force application on the retina. The surgeon can use this to assess which areas have been heavily manipulated and could be compared to future surgical outcomes.

Visual Virtual Fixtures - a set of Behaviors that are defined by virtually annotating the retina. These geometric definitions are then used by the robot controller to generate a variety of virtual fixtures; for example, to restrict the motion of the tool to a region of the retina. Alternatively, an audio or visual alarm could be incorporated to warn if the tool is near a particular region.

Eye Safety Zones - a virtual fixture Behavior that attempts to minimize the occurrence of inadvertent collisions between the instrument and the tissues by enforcing different virtual fixtures in different zones of the eye. I.e., if the instruments held by the robotic-assistant are close to the retina, the maximum tool velocity, or control gains are significantly reduced, compared to when the tool tip is in the center of eye. Alternatively, the robot could enforce a "no-fly zone" where the tool is not allowed to travel to minimize tool collisions with the lens, a common concern for novice surgeons.

Extra Hand Light - a robot and vision-based Behavior that controls the position of the light pipe that is held by the robotic-assistant. The robot can automatically move the light pipe to follow the other surgical instruments in the eye, enabling the surgeon to manipulate tissues bi-manually, e.g., with two forceps instruments.

Force Guided Cannulation - a force-based robot control Behavior that enables a safe and semi-automatic puncture and insertion of a pipette into a retinal blood vessel for a direct delivery of medication. Additionally, the force-sensor feedback can be used by the robot to assist in maintaining the cannulation for the duration of the injection, which tends to be the most difficult part of the task.

Micro-Elastography - provides functional imaging of the retinal tissues by incorporating real-time intraocular OCT and a sinusoidal compression of the retina (e.g., using a water jet). This provides the surgeon with structural and functional information that he or she can use for diagnostic purposes.

The work presented here was part of a very large interdisciplinary project and could not progress very far without the consultation and contributions from the many expert members of the *eyeBRP* team. Future developments in other domains should consider assembling a group of such domain experts to facilitate rapid progress and generation of innovative solutions that span many disciplines.

The *eyeSAW* system has great potential to revolutionize microsurgical training by providing specific manual and information-driven guidance in a simulated training environment with phantoms, and also in a real surgical scenario using safety Behaviors and real-time sensory substitution Behaviors. Future development should investigate physical and software simulators, and expand *cisstDataPlayer* (see Section 3.4) functionality to simulate various components in the system.

A basic tenet of the work presented herein is that the system developers consider all aspects of surgical practice including the greater relevance of their work. In the case of the *eyeSAW* system development, the ultimate inspiration is the potential benefits to the millions of ophthalmological surgery patients that undergo high-risk but necessary procedures.

Bibliography

[1] G. Guthart and J. Salisbury, "The Intuitive telesurgery system: overview and application," in *International Conference on Robotics and Automation.*, vol. 1, no. April. IEEE, 2000, pp. 618–621.

[2] X. Zhang, K. M. Bullard, M. F. Cotch, M. R. Wilson, B. W. Rovner, G. McGwin, C. Owsley, L. Barker, J. E. Crews, and J. B. Saaddine, "Association between depression and functional vision loss in persons 20 years of age or older in the United States, NHANES 2005-2008." *JAMA ophthalmology*, vol. 131, no. 5, pp. 573–81, May 2013.

[3] G. Rees, H. W. Tee, M. Marella, E. Fenwick, M. Dirani, and E. L. Lamoureux, "Vision-specific distress and depressive symptoms in people with vision impairment." *Investigative ophthalmology & visual science*, vol. 51, no. 6, pp. 2891–6, Jun. 2010.

[4] N. A. Frost, J. M. Sparrow, J. S. Durant, J. L. Donovan, T. J. Peters, and S. T. Brookes, "Development of a questionnaire for measurement of vision-related quality of life," *Ophthalmic Epidemiology*, vol. 5, no. 4, 1998.

[5] R. Misajon, G. Hawthorne, J. Richardson, J. Barton, S. Peacock, A. Iezzi, and J. Keeffe, "Vision and quality of life: the development of a utility measure." *Investigative ophthalmology & visual science*, vol. 46, no. 11, pp. 4007–15, Nov. 2005.

[6] E. L. Hamblion, A. T. Moore, and J. S. Rahi, "The health-related quality of life of children with hereditary retinal disorders and the psychosocial impact on their families." *Investigative ophthalmology & visual science*, vol. 52, no. 11, pp. 7981–6, Jan. 2011.

[7] World Health Organization, "Visual impairment and blindness - Fact Sheet N°282," WHO Media Centre, Tech. Rep. June, 2012.

[8] J. W. Y. Yau, S. L. Rogers, R. Kawasaki, E. L. Lamoureux, J. W. Kowalski, T. Bek, S.-J. Chen, J. M. Dekker, A. Fletcher, J. Grauslund, S. Haffner, R. F. Hamman, M. K. Ikram, T. Kayama, B. E. K. Klein, R. Klein, S. Krishnaiah, K. Mayurasakorn, J. P. O'Hare, T. J. Orchard, M. Porta, M. Rema, M. S. Roy, T. Sharma, J. Shaw, H. Taylor, J. M. Tielsch, R. Varma, J. J. Wang, N. Wang, S. West, L. Xu, M. Yasuda, X. Zhang, P. Mitchell, and T. Y. Wong, "Global prevalence and major risk factors of diabetic retinopathy." *Diabetes care*, vol. 35, no. 3, pp. 556–64, Mar. 2012.

[9] H. Nazimul, K. Rohit, and H. Anjli, “Trend of retinal diseases in developing countries,” *Expert Review of Ophthalmology*, vol. 3, no. 1, pp. 43–50, Feb. 2008.

[10] D. Landau, E. M. Schneidman, T. Jacobovitz, and Y. Rozenman, “Quantitative In Vivo Retinal Thickness Measurements in Healthy Subjects,” *Ophthalmology*, vol. 104, no. 4, pp. 639–642, Apr. 1997.

[11] S. E. Burk, A. P. Da Mata, M. E. Snyder, R. H. Rosa, and R. E. Foster, “Indocyanine green–assisted peeling of the retinal internal limiting membrane,” *Ophthalmology*, vol. 107, no. 11, pp. 2010–2014, Nov. 2000.

[12] C. H. G. Wright, S. F. Barrett, and A. J. Welch, “Design and development of a computer-assisted retinal laser surgery system.” *Journal of biomedical optics*, vol. 11, no. 4, p. 041127, 2006.

[13] M. P. S. Group, “Argon laser photocoagulation for neovascular maculopathy. Five-year results from randomized clinical trials.” *Archives of ophthalmology*, vol. 109, no. 8, pp. 1109–14, Aug. 1991.

[14] R. L. Avery, J. Pearlman, D. J. Pieramici, M. D. Rabena, A. a. Castellarin, M. a. Nasir, M. J. Giust, R. Wendel, and A. Patel, “Intravitreal bevacizumab (Avastin) in the treatment of proliferative diabetic retinopathy.” *Ophthalmology*, vol. 113, no. 10, pp. 1695.e1–15, Oct. 2006.

[15] N. M. Bressler, “Age-Related Macular Degeneration Is the Leading Cause of Blindness . . .” *JAMA*, vol. 291, no. 15, p. 1900, Apr. 2004.

[16] R. D. Jager, W. F. Mieler, and J. W. Miller, “Age-related macular degeneration.” *The New England journal of medicine*, vol. 358, no. 24, pp. 2606–17, Jun. 2008.

[17] N. Berker and C. Batman, “Surgical treatment of central retinal vein occlusion.” *Acta ophthalmologica*, vol. 86, no. 3, pp. 245–52, May 2008.

[18] T. H. Williamson, “Central retinal vein occlusion: what’s the story?” *British Journal of Ophthalmology*, vol. 81, no. 8, pp. 698–704, Aug. 1997.

[19] D. J. Browning, *Retinal Vein Occlusions: Evidence-Based Management.* Springer, 2012.

[20] E. M. Opremcak, A. J. Rehmar, C. D. Ridenour, and D. E. Kurz, “Radial Optic Neurotomy for CRVO,” *Retina (Philadelphia, Pa.)*, vol. 26, no. 3, pp. 297–305, Mar. 2006.

[21] N. Crama, V. Gualino, M. Restori, and D. G. Charteris, “Central retinal vessel blood flow after surgical treatment for central retinal vein occlusion.” *Retina (Philadelphia, Pa.)*, vol. 30, no. 10, pp. 1692–7, 2010.

[22] J. N. Weiss and L. A. Bynoe, “Injection of tissue plasminogen activator into a branch retinal vein in eyes with central retinal vein occlusion,” *Ophthalmology*, vol. 108, no. 12, pp. 2249–2257, Dec. 2001.

[23] K. Kadonosono, S. Yamane, A. Arakawa, M. Inoue, T. Yamakawa, E. Uchio, Y. Yanagi, and S. Amano, "Endovascular cannulation with a microneedle for central retinal vein occlusion." *JAMA ophthalmology*, vol. 131, no. 6, pp. 783–6, Jun. 2013.

[24] W. Wei, C. Popplewell, S. Chang, H. F. Fine, and N. Simaan, "Enabling Technology for Microvascular Stenting in Ophthalmic Surgery," *Journal of Medical Devices*, vol. 4, no. 1, p. 014503, 2010.

[25] R. N. Sjaarda and R. Michels, "Macular Pucker," *Retina*, vol. 3, pp. 2301–2312, 1994.

[26] J. E. Galbraith, "Ophthalmic microsurgery." *The Australian and New Zealand journal of surgery*, vol. 47, no. 3, pp. 310–4, Jun. 1977.

[27] S. Charles, "Techniques and tools for dissection of epiretinal membranes." *Graefe's archive for clinical and experimental ophthalmology*, vol. 241, no. 5, pp. 347–52, May 2003.

[28] J. R. Wilkins, C. A. Puliafito, M. R. Hee, J. S. Duker, E. Reichel, J. G. Coker, J. S. Schuman, E. A. Swanson, and J. G. Fujimoto, "Characterization of epiretinal membranes using optical coherence tomography." *Ophthalmology*, vol. 103, no. 12, pp. 2142–51, Dec. 1996.

[29] Y. Hirano, T. Yasukawa, and Y. Ogura, "Optical coherence tomography guided peeling of macular epiretinal membrane," *Clinical ophthalmology*, vol. 2011:5, pp. 27–29, Jan. 2010.

[30] A. P. Da Mata, S. E. Burk, C. D. Riemann, R. H. Rosa, M. E. Snyder, M. R. Petersen, and R. E. Foster, "Indocyanine green-assisted peeling of the retinal internal limiting membrane during vitrectomy surgery for macular hole repair," *Ophthalmology*, vol. 108, no. 7, pp. 1187–1192, Jul. 2001.

[31] U. C. Christensen, "Value of internal limiting membrane peeling in surgery for idiopathic macular hole and the correlation between function and retinal morphology." *Acta ophthalmologica*, vol. 87 Thesis, pp. 1–23, Dec. 2009.

[32] B. C. Joondeph, "When Should You Peel the ILM in Macular Hole Surgery ?" *Retinal Physician*, Nov. 2010.

[33] B. Allf and E. de Juan Jr, "In vivo cannulation of retinal vessels," *Graefe's Arch Clin Exp Ophthalmol*, pp. 221–225, 1987.

[34] L. A. Bynoe, R. K. Hutchins, H. S. Lazarus, and M. A. Friedberg, "Retinal endovascular surgery for central retinal vein occlusion: initial experience of four surgeons." *Retina (Philadelphia, Pa.)*, vol. 25, no. 5, pp. 625–32, 2005.

[35] S. Singhy and C. Riviere, "Physiological tremor amplitude during retinal microsurgery," in *Proceedings of the IEEE 28th Annual Northeast Bioengineering Conference*. IEEE, 2002, pp. 171–172.

[36] P. Gupta, P. Jensen, and E. de Juan, "Surgical Forces and Tactile Perception During Retinal Microsurgery," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI'99*, 1999.

[37] L. Roach, "The Silent Epidemic," *EyeNet (an electronic publication of the American Academy of Ophthalmology)*, 2002.

[38] R. L. Harman, "Microscope-induced cervical spine disease ends surgical career," *EyeWorld News Service*, no. June, Jun. 2006.

[39] I. Lieberman, J. Herndon, J. Hahn, J. J. Fins, and A. Rezai, "Surgical innovation and ethical dilemmas: a panel discussion." *Cleveland Clinic journal of medicine*, vol. 75 Suppl 6, pp. S13–21, Nov. 2008.

[40] R. Taylor, B. Mittelstadt, H. Paul, W. Hanson, P. Kazanzides, J. Zuhars, B. Williamson, B. Musits, E. Glassman, and W. Bargar, "An image-directed robotic system for precise orthopaedic surgery," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 3, pp. 261–275, Jun. 1994.

[41] P. U. Dugel, "Early Clinical Experience With the CONSTELLATION ® Vision System," *Retinal Physician*, 2009. [Online]. Available: http://www.retinalphysician.com/

[42] R. Taylor and D. Stoianovici, "Medical robotics in computer-integrated surgery," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 765–781, Oct. 2003.

[43] P. Kazanzides, G. Fichtinger, G. D. Hager, M. Allison, L. L. Whitcomb, R. H. Taylor, and A. M. Okamura, "Surgical and Interventional Robotics: Core Concepts, Technology, and Design," *IEEE robotics & automation magazine / IEEE Robotics & Automation Society*, vol. 15, no. 2, pp. 122–130, Dec. 2008.

[44] S. P. DiMaio and C. J. Hasser, "The da Vinci Research Interface," in *MICCAI Workshop on Systems and Arch. for Computer Assisted Interventions*, Midas Journal, Sep. 2008.

[45] B. Vagvolgyi, S. DiMaio, A. Deguet, P. Kazanzides, R. Kumar, C. Hasser, and R. Taylor, "The Surgical Assistant Workstation: a software framework for telesurgical robotics research," in *MICCAI Workshop on Systems and Arch. for Computer Assisted Interventions*, Midas Journal, Sep. 2008.

[46] P. Kazanzides, S. P. DiMaio, A. Deguet, B. Vagvolgyi, M. Balicki, C. Schneider, R. Kumar, A. Jog, B. Itkowitz, C. Hasser, and R. H. Taylor, "The Surgical Assistant Workstation (SAW) in Minimally-Invasive Surgery and Microsurgery," in *MICCAI Workshop on Systems and Arch. for Computer Assisted Interventions*, Midas Journal, Jun. 2010, pp. 1–9.

[47] P. Kazanzides, M. Y. Jung, A. Deguet, B. Vagvolgyi, M. Balicki, and R. H. Taylor, "Component-based software for dynamic configuration and control of computer assisted intervention systems," *Insight Journal*, pp. 1–9, 2011.

[48] (2013) The cisst/SAW libraries. [Online]. Available: https://trac.lcsr.jhu.edu/cisst

[49] C. Szyperski, *Component Software: Beyond Object-Oriented Programming.* Addison-Wesley, 1997.

[50] A. Brooks, T. Kaupp, A. Makarenko, S. Williams, and A. Oreback, "Towards component-based robotics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2005, pp. 163–168.

[51] O. Schorr, N. Hata, A. Bzostek, R. Kumar, C. Burghart, R. H. Taylor, and R. Kikinis, "Distributed Modular Computer−Integrated Surgical Robotic Systems: Architecture for Intelligent Object Distribution," *Int Conf Med Image Comput Comput Assist Interv.*, vol. 3, pp. 979–987, 2000.

[52] A. Bzostek, R. Kumar, N. Hata, O. Schorr, R. Kikinis, and R. H. Taylor, "Distributed Modular Computer-Integrated Surgical Robotic Systems: Implementation Using Modular Software and Networked Systems," in *Intl. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI).* London, UK: Springer-Verlag, 2000, pp. 969–978.

[53] S. Pieper, M. Halle, and R. Kikinis, "3D Slicer," in *2004 2nd IEEE International Symposium on Biomedical Imaging: Macro to Nano (IEEE Cat No. 04EX821)*, vol. 2. IEEE, 2004, pp. 632–635.

[54] A. Enquobahrie, P. Cheng, K. Gary, L. Ibanez, D. Gobbi, F. Lindseth, Z. Yaniv, S. Aylward, J. Jomier, and K. Cleary, "The image-guided surgery toolkit IGSTK: an open source C++ software toolkit." *Journal of digital imaging*, vol. 20 Suppl 1, no. August, pp. 21–33, Nov. 2007.

[55] M. Nolden, S. Zelzer, A. Seitel, D. Wald, M. Müller, A. M. Franz, D. Maleike, M. Fangerau, M. Baumhauer, L. Maier-Hein, K. H. Maier-Hein, H.-P. Meinzer, and I. Wolf, "The Medical Imaging Interaction Toolkit: challenges and advances : 10 years of open-source development." *International journal of computer assisted radiology and surgery*, vol. 8, no. 4, pp. 607–20, Jul. 2013.

[56] L. Ibanez, J. Jomier, D. Gobbi, R. Avila, M. Blake, H.-s. Kim, K. Gary, S. Aylward, and K. Cleary, "IGSTK: A State machine architecture for an open source software toolkit for image-guided surgery applications," *The Insight Journal - 2005 MICCAI Open-Source Workshop.*, pp. 1–9, 2005.

[57] A. Elkady and T. Sobh, "Robotics Middleware: A Comprehensive Literature Survey and Attribute-Based Bibliography," *Journal of Robotics*, vol. 2012, pp. 1–15, 2012.

[58] H. Bruyninckx, "Open robot control software: the OROCOS project," in *IEEE International Conference on Robotics and Automation*, vol. 3. Ieee, 2001, pp. 2523–2528.

[59] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "{ROS}: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[60] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[61] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, 1st ed. Addison-Wesley Professional, 1994.

[62] F. Zheng, H. Hu, and J. Lü, "Applying a Component Behavior Model to MVC Pattern," in *2008 The 9th International Conference for Young Computer Scientists.* IEEE, Nov. 2008, pp. 1106–1111.

[63] D. Crockford, "The application/json media type for javascript object notation (json)," rfc4627, Tech. Rep., 2006.

[64] X. Liu, M. Balicki, R. H. Taylor, and J. U. Kang, "Towards automatic calibration of Fourier-Domain OCT for robot-assisted vitreoretinal surgery," *Optics Express*, vol. 18, no. 23, p. 24331, Nov. 2010.

[65] D. Mills, "Internet time synchronization: the network time protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.

[66] M. Y. Jung, A. Deguet, and P. Kazanzides, "A Component-based Architecture for Flexible Integration of Robotic Systems," in *Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[67] M. Henning, "A new approach to object-oriented middleware," *IEEE Internet Computing*, vol. 8, no. 1, pp. 66–75, 2004.

[68] J. Niparko, W. Chien, I. Iordachita, J. Kang, and R. Taylor, "Robot-assisted, sensor-guided cochlear implant electrode insertion," in *12th International Conference on Cochlear Implants and Other Implantable Auditory Technologies*, 2012.

[69] R. H. Taylor, W. W. Chien, I. I. Iordachita, and J. Niparko, "Method And Apparatus For Cochlear Implant Surgery," 2012.

[70] T. Xia, L. Simon, A. Deguet, L. Whitcomb, and P. Kazanzides, "Augmented Reality Environment with Virtual Fixtures for Robotic Telemanipulation in Space," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5059–5064.

[71] T. Xia, S. Leonard, I. Kandaswamy, A. Blank, L. L. Whitcomb, and P. Kazanzides, "Model-based telerobotic control with virtual fixtures for satellite servicing tasks," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, May 2013, pp. 1479–1484.

[72] M. Jung and P. Kazanzides, "Fault Detection and Diagnosis for Component-based Robotic Systems," in *IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*. IEEE, 2012, pp. 1–6.

[73] P. Kazanzides, Y. Kouskoulas, A. Deguet, and Z. Shao, "Proving the correctness of concurrent robot software," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 4718–4723.

[74] D. Kragic, P. Marayong, M. Li, A. Okamura, and G. Hager, "Human-Machine Collaborative Systems for Microsurgical Applications," *The International Journal of Robotics Research*, vol. 24, no. 9, pp. 731–741, Sep. 2005.

[75] M. N. Iyer and D. P. Han, "An eye model for practicing vitreoretinal membrane peeling." *Archives of ophthalmology*, vol. 124, no. 1, pp. 108–10, Jan. 2006.

[76] A. S. Jagtap and C. N. Riviere, "Applied force during vitreoretinal microsurgery with handheld instruments." *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, vol. 4, no. 1, pp. 2771–3, Jan. 2004.

[77] T. Leng, J. M. Miller, K. V. Bilbao, D. V. Palanker, P. Huie, and M. S. Blumenkranz, "The chick chorioallantoic membrane as a model tissue for surgical retinal research and simulation." *Retina*, vol. 24, no. 3, pp. 427–34, Jun. 2004.

[78] R. A. MacLachlan, B. C. Becker, J. Cuevas Tabares, G. W. Podnar, L. A. Lobes, and C. N. Riviere, "Micron: An Actively Stabilized Handheld Tool for Microsurgery," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 195–212, Feb. 2012.

[79] I. Tsui, A. Tsirbas, C. W. Mango, S. D. Schwartz, and J.-p. Hubschman, "Robotic Surgery in Ophthalmology," in *Robot Surgery*, 2010, no. January, ch. 10.

[80] J. D. Pitcher, J. T. Wilson, S. D. Schwartz, and J.-p. Hubschman, "Robotic Eye Surgery : Past , Present , and Future," *J Comput Sci Syst Biol*, pp. 1–4, 2012.

[81] A. Tsirbas, C. Mango, and E. Dutson, "Robotic ocular surgery." *The British journal of ophthalmology*, vol. 91, no. 1, pp. 18–21, Jan. 2007.

[82] J. P. Hubschman, A. Tsirbas, and S. D. Schwartz, "Robotic Surgery in Ophthalmology," *Retina Today*, no. June, pp. 81–84, 2008.

[83] A. P. Mulgaonkar, J.-P. Hubschman, J.-L. Bourges, B. L. Jordan, C. Cham, J. T. Wilson, T.-C. Tsao, and M. O. Culjat, "A prototype surgical manipulator for robotic intraocular micro surgery." *Studies in health technology and informatics*, vol. 142, pp. 215–7, Jan. 2009.

[84] J.-L. Bourges, J.-P. Hubschman, J. Wilson, S. Prince, T.-C. Tsao, and S. Schwartz, "Assessment of a hexapod surgical system for robotic micro-macro manipulations in ocular surgery." *Ophthalmic research*, vol. 46, no. 1, pp. 25–30, Jan. 2011.

[85] I. W. Hunter, T. Doukoglou, S. R. Lafontaine, P. G. Charette, L. A. Jones, M. A. Sagar, G. D. Mallinson, and P. J. Hunter, "A Teleoperated Microsurgical Robot and Associated Virtual Environment for Eye Surgery," *Presence*, vol. 2, no. 4, pp. 265–280, 1993.

[86] I. W. Hunter, L. A. Jones, M. A. Sagar, S. R. Lafontaine, and P. J. Hunter, "Ophthalmic microsurgical robot and associated virtual environment." *Computers in biology and medicine*, vol. 25, no. 2, pp. 173–82, Mar. 1995.

[87] P. S. Jensen, K. W. Grace, R. Attariwala, J. E. Colgate, and M. R. Glucksberg, "Toward robot-assisted vascular microsurgery in the retina." *Graefe's archive for clinical and experimental ophthalmology*, vol. 235, no. 11, pp. 696–701, Nov. 1997.

[88] K. W. Grace, P. Jensen, J. Colgate, and M. Glucksberg, "Teleoperation for ophthalmic surgery: From the Eye Robot to feature extracting force feedback," *Automedica*, 1998.

[89] T. Nakano, N. Sugita, T. Ueta, Y. Tamaki, and M. Mitsuishi, "A parallel robot to assist vitreoretinal surgery." *International journal of computer assisted radiology and surgery*, vol. 4, no. 6, pp. 517–26, Nov. 2009.

[90] W. Wei, R. E. Goldman, H. F. Fine, S. Chang, and N. Simaan, "Performance Evaluation for Multi-arm Manipulation of Hollow Suspended Organs," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 147–157, Feb. 2009.

[91] A. Guerrouad and P. Vidal, “SMOS: stereotaxical microtelemanipulator for ocular surgery,” in *Images of the Twenty-First Century. Proceedings of the Annual International Engineering in Medicine and Biology Society.* IEEE, 1989, pp. 879–880.

[92] A. Guerrouad and P. Vidal, “Advantage of computer aided teleoperation (CAT) in microsurgery,” in *Fifth International Conference on Advanced Robotics -Robots in Unstructured Environments.* IEEE, 1991, pp. 910–914 vol.1.

[93] D.-y. Yu, S. Cringle, and I. Constable, “Robotic ocular ultramicrosurgery,” *Australian and New Zealand Journal of Ophthalmology*, vol. 26, pp. S6–S8, May 1998.

[94] T. Ueta, Y. Yamaguchi, Y. Shirakawa, T. Nakano, R. Ideta, Y. Noda, A. Morita, R. Mochizuki, N. Sugita, M. Mitsuishi, and Y. Tamaki, “Robot-assisted vitreoretinal surgery: development of a prototype and feasibility studies in an animal model.” *Ophthalmology*, vol. 116, no. 8, pp. 1538–43, 1543.e1–2, Aug. 2009.

[95] Y. Ida, N. Sugita, T. Ueta, Y. Tamaki, K. Tanimoto, and M. Mitsuishi, “Microsurgical robotic system for vitreoretinal surgery.” *International journal of computer assisted radiology and surgery*, vol. 7, no. 1, pp. 27–34, Jan. 2012.

[96] P. S. Schenker, H. Das, and T. R. Ohm, “Development of a new high-dexterity manipulator for robot-assisted microsurgery,” in *Telemanipulator and Telepresence Technologies*, Dec. 1995, pp. 191–198.

[97] H. Das, H. Zak, J. Johnson, J. Crouch, and D. Frambach, “Evaluation of a telerobotic system to assist surgeons in microsurgery.” *Computer aided surgery : official journal of the International Society for Computer Aided Surgery*, vol. 4, no. 1, pp. 15–25, Jan. 1999.

[98] M. Siemionow, K. Ozer, W. Siemionow, and G. Lister, “Robotic Assistance in Microsurgery,” *Journal of Reconstructive Microsurgery*, vol. Volume 16, no. Number 8, pp. 0643–0650, 2000.

[99] T. Meenink, “Vitreo-retinal eye surgery robot: sustainable precision,” Ph.D. dissertation, 2011.

[100] S. Salcudean and J. Yan, “Towards a force-reflecting motion-scale system for microsurgery,” in *Robotics and Automation*, 1994, pp. 2296–2301.

[101] S. Ku and S. Salcudean, “Design and control of a teleoperated microgripper for microsurgery,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, no. April. IEEE, 1996, pp. 889–894.

[102] C. Bergeles, B. E. Kratochvil, and B. J. Nelson, “Visually Servoing Magnetic Intraocular Microdevices,” *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 798–809, Aug. 2012.

[103] R. Taylor, “A Steady-Hand Robotic System for Microsurgical Augmentation,” *The International Journal of Robotics Research*, vol. 18, no. 12, pp. 1201–1210, Dec. 1999.

[104] R. Kumar, P. Berkelman, P. Gupta, A. Barnes, P. Jensen, L. Whitcomb, and R. Taylor, “Preliminary experiments in cooperative human/robot force control for robot assisted microsurgical manipulation,” *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 610–617, 2000.

[105] B. Mitchell, J. Koo, I. Iordachita, P. Kazanzides, A. Kapoor, J. Handa, G. Hager, and R. Taylor, “Development and Application of a New Steady-Hand Manipulator for Retinal Surgery,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, no. April. IEEE, 2007, pp. 10–14.

[106] I. Fleming, M. Balicki, J. Koo, I. Iordachita, B. Mitchell, J. Handa, G. Hager, and R. Taylor, “Cooperative robot assistant for retinal microsurgery.” in *Medical Image Computing and Computer-Assisted Intervention*, vol. 11, no. Pt 2, Jan. 2008, pp. 543–50.

[107] A. Uneri, M. A. Balicki, J. Handa, P. Gehlbach, R. H. Taylor, and I. Iordachita, “New Steady-Hand Eye Robot with Micro-Force Sensing for Vitreoretinal Surgery.” *IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, vol. 2010, no. 26-29, pp. 814–819, Sep. 2010.

[108] X. He, D. Roppenecker, D. Gierlach, M. Balicki, K. Olds, P. Gehlbach, J. Handa, R. Taylor, and I. Iordachita, “Toward Clinically Applicable Steady-hand Eye Robot for Vitreoretinal Surgery,” in *ASME 2012 International Mechanical Engineering Congress & Exposition*, no. Ilm, 2012.

[109] B. C. Becker, C. N. Riviere, and G. D. Stetten, “Vision-Based Control of a Handheld Micromanipulator for Robot-Assisted Retinal Surgery,” Thesis, Carnegie Mellon University, 2012.

[110] S. Yang, R. A. MacLachlan, and C. N. Riviere, “Design and analysis of 6 DOF handheld micromanipulator,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, May 2012, pp. 1946–1951.

[111] R. A. Maclachlan and C. N. Riviere, “High-Speed Microscale Optical Tracking Using Digital Frequency-Domain Multiplexing.” *IEEE transactions on instrumentation and measurement*, vol. 58, no. 6, pp. 1991–2001, Jun. 2009.

[112] S. Yang, M. A. Balicki, R. A. Maclachlan, X. Liu, J. U. Kang, R. H. Taylor, and C. N. Riviere, “Optical coherence tomography scanning with a handheld vitreoretinal micromanipulator.” in *Conference of the IEEE Engineering in Medicine and Biology Society.*, vol. 2012, Aug. 2012, pp. 948–51.

[113] J.-P. Hubschman, J. Son, B. Allen, S. D. Schwartz, and J.-L. Bourges, “Evaluation of the motion of surgical instruments during intraocular surgery.” *Eye*, vol. 25, no. 7, pp. 947–53, Jul. 2011.

[114] G. Casiez, N. Roussel, and D. Vogel, “1-Euro filter: A simple speed-based low-pass filter for noisy input in interactive systems,” *Humn Factors in Computing Systems*, pp. 2527–2530, 2012.

[115] P. Kazanzides, J. Zuhars, B. Mittelstadt, and R. H. Taylor, “Force Sensing and Control for a Surgical Robot,” in *IEEE International Conference on Robotics and Automation*, 1992, pp. 612–617.

[116] L. B. Rosenberg, “Virtual fixtures: Perceptual tools for telerobotic manipulation,” in *Virtual Reality Annual International Symposium*, 1993, pp. 76–82.

[117] J. Abbott and A. Okamura, “Virtual fixture architectures for telemanipulation,” in *IEEE International Conference on Robotics and Automation.* Ieee, 2003, pp. 2798–2805.

[118] J. J. Abbott and a. M. Okamura, "Pseudo-admittance Bilateral Telemanipulation with Guidance Virtual Fixtures," *The International Journal of Robotics Research*, vol. 26, no. 8, pp. 865–884, Aug. 2007.

[119] A. Bettini, P. Marayong, S. Lang, A. M. Okamura, and G. D. Hager, "Vision-Assisted Control for Manipulation Using Virtual Fixtures," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 953–966, 2004.

[120] B. C. Becker, R. A. MacLachlan, L. A. Lobes, and C. N. Riviere, "Vision-based retinal membrane peeling with a handheld robot," in *IEEE International Conference on Robotics and Automation*. IEEE, May 2012, pp. 1075–1080.

[121] J. Funda, R. Taylor, B. N. Eldridge, S. H. Gomory, and K. G. Gruben, "Constrained Cartesian Motion Control for Teleoperated Surgical Robots," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 3, pp. 453–465, 1996.

[122] A. Kapoor, M. Li, and R. Taylor, "Constrained control for surgical assistant robots," *IEEE International Conference on Robotics and . . .* , no. May, pp. 231–236, 2006.

[123] A. Kapoor, "Motion Constrained Control of Robots for Dexterous Surgical Tasks," Ph.D. dissertation, The Johns Hopkins University, 2007.

[124] X. He, M. Balicki, P. Gehlbach, J. Handa, R. H. Taylor, and I. Iordachita, "A Novel Dual Force Sensing Instrument with Cooperative Robotic Assistant for Vitreoretinal Surgery," in *IEEE International Conference on Robotics and Automation*, 2013.

[125] M. Balicki, T. Xia, M. Y. Jung, A. Deguet, B. Vagvolgyi, P. Kazanzides, and R. Taylor, "Prototyping a Hybrid Cooperative and Tele-robotic Surgical System for Retinal Microsurgery," *Insight Journal*, pp. 1–10, 2011.

[126] S. Nudehi, R. Mukherjee, and M. Ghodoussi, "A shared-control approach to haptic interface design for minimally invasive telesurgical training," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 588–592, Jul. 2005.

[127] B. Khademian and K. Hashtrudi-Zaad, "A four-channel multilateral shared control architecture for dual-user teleoperation systems," in *IEEE RSJ International Conference on Intelligent Robots and Systems*. Ieee, Oct. 2007, pp. 2660–2666.

[128] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, Dec. 2006.

[129] B. C. Becker, S. Voros, L. A. Lobes, J. T. Handa, G. D. Hager, and C. N. Riviere, "Retinal vessel cannulation with an image-guided handheld robot." in *IEEE Engineering in Medicine and Biology Society. Conference*, vol. 2010, Jan. 2010, pp. 5420–3.

[130] C. Riviere and P. Jensen, "A study of instrument motion in retinal microsurgery," in *IEEE Engineering in Medicine and Biology Society Conference*, vol. 1. IEEE, 2000, pp. 59–60.

[131] P. Berkelman, L. Whitcomb, R. Taylor, and P. Jensen, "A miniature microsurgical instrument tip force sensor for enhanced force feedback during robot-assisted manipulation," in *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, Oct. 2003, pp. 917–922.

[132] P. J. Berkelman, D. L. R. M. D, J. Roy, S. Lang, L. L. Whitcomb, G. Hager, P. S. Jensen, E. D. Juan, R. H. Taylor, and J. K. N. M. D, "Performance Evaluation of a Cooperative Manipulation Microsurgical Assistant Robot Applied to Stapedotomy," in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, no. October 2001, 2001.

[133] J. Roy, D. Rothbaum, and L. Whitcomb, "Haptic feedback enhancement through adaptive force scaling: Theory and experiment," *Advances in Robot Control*, pp. 293–316, 2006.

[134] M. Kitagawa, D. Dokko, A. M. Okamura, and D. D. Yuh, "Effect of sensory substitution on suture-manipulation forces for robotic surgical systems." *The Journal of thoracic and cardiovascular surgery*, vol. 129, no. 1, pp. 151–8, Jan. 2005.

[135] M. Balicki, A. Uneri, I. Iordachita, J. Handa, P. Gehlbach, and R. Taylor, "Micro-force sensing in robot assisted membrane peeling for vitreoretinal surgery." *International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 13, no. Pt 3, pp. 303–10, Jan. 2010.

[136] I. Iordachita, Z. Sun, M. Balicki, J. U. Kang, S. J. Phee, J. Handa, P. Gehlbach, and R. Taylor, "A sub-millimetric, 0.25 mN resolution fully integrated fiber-optic force-sensing tool for retinal microsurgery." *International journal of computer assisted radiology and surgery*, vol. 4, no. 4, pp. 383–90, Jun. 2009.

[137] I. Kuru, B. Gonenc, M. Balicki, J. Handa, P. Gehlbach, R. H. Taylor, and I. Iordachita, "Force sensing micro-forceps for robot assisted retinal surgery," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* IEEE, Aug. 2012, pp. 1401–1404.

[138] X. He, M. A. Balicki, J. U. Kang, P. L. Gehlbach, J. T. Handa, R. H. Taylor, and I. I. Iordachita, "Force sensing micro-forceps with integrated fiber Bragg grating for vitreoretinal surgery," in *SPIE BiOS*, Feb. 2012, pp. 82 180W–82 180W.

[139] Z. Sun, M. Balicki, J. Kang, J. Handa, R. Taylor, and I. Iordachita, "Development and preliminary data of novel integrated optical micro-force sensing tools for retinal microsurgery," in *2009 IEEE International Conference on Robotics and Automation.* IEEE, May 2009, pp. 1897–1902.

[140] S. Sunshine, M. Balicki, X. He, K. Olds, J. U. Kang, P. Gehlbach, R. Taylor, I. Iordachita, and J. T. Handa, "A Force-sensing Microsurgical Instrument That Detects Forces Below Human Tactile Sensation," *Retina (Philadelphia, Pa.)*, Jul. 2012.

[141] N. Cutler, M. Balicki, M. Finkelstein, J. Wang, P. Gehlbach, J. McGready, I. Iordachita, R. Taylor, and J. T. Handa, "Auditory Force Feedback Substitution Improves Surgical Precision During Simulated Ophthalmic Surgery," *Investigative Ophthalmology & Visual Science*, Jan. 2013.

[142] P. Marayong, "Motion control methods for human-machine cooperative systems," Ph.D. dissertation, Johns Hopkins University, 2007.

[143] E. Gentleman, A. N. Lay, D. a. Dickerson, E. a. Nauman, G. a. Livesay, and K. C. Dee, "Mechanical characterization of collagen fibers and scaffolds for tissue engineering," *Biomaterials*, vol. 24, no. 21, pp. 3805–3813, Sep. 2003.

[144] C. Taylor, “The biomechanics of the normal and of the amputated upper extremities,” in *Human limbs and their substitutes*, P. Klopsteg and P. Wilson, Eds. New York, New York, USA: Hafner Publishing Co., 1968, ch. 7, pp. 169–221.

[145] B. Gonenc, M. A. Balicki, J. Handa, P. Gehlbach, C. N. Riviere, R. H. Taylor, and I. Iordachita, “Preliminary evaluation of a micro-force sensing handheld robot for vitreoretinal surgery,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2012, Dec. 2012, pp. 4125–4130.

[146] R. N. Stiles, “Mechanical and neural feedback factors in postural hand tremor of normal subjects.” *Journal of neurophysiology*, vol. 44, no. 1, pp. 40–59, Jul. 1980.

[147] O. Ergeneman, J. Pokki, V. Pocepcova, H. Hall, J. J. Abbott, and B. J. Nelson, “Characterization of Puncture Forces for Retinal Vein Cannulation,” *Journal of Medical Devices*, vol. 5, no. 4, p. 044504, 2011.

[148] X. He, J. Handa, P. Gehlbach, R. Taylor, and I. Iordachita, “A Sub-Millimetric 3-DOF Force Sensing Instrument with Integrated Fiber Bragg Grating for Retinal Microsurgery.” *IEEE transactions on bio-medical engineering*, vol. PP, no. 99, p. 1, Sep. 2013.

[149] M. K. Tameesh, R. R. Lakhanpal, G. Y. Fujii, M. Javaheri, T. H. Shelley, S. D’Anna, A. C. Barnes, E. Margalit, M. Farah, E. De Juan, and M. S. Humayun, “Retinal vein cannulation with prolonged infusion of tissue plasminogen activator (t-PA) for the treatment of experimental retinal vein occlusion in dogs.” *American journal of ophthalmology*, vol. 138, no. 5, pp. 829–39, Nov. 2004.

[150] J. G. Fujimoto, C. Pitris, S. a. Boppart, M. E. Brezinski, and R. Article, “Optical coherence tomography: an emerging technology for biomedical imaging and optical biopsy.” *Neoplasia (New York, N.Y.)*, vol. 2, no. 1-2, pp. 9–25, 2000.

[151] J. M. Herrmann, S. A. Boppart, B. E. Bouma, G. J. Tearney, C. Pitris, M. E. Brezinski, and J. G. Fujimoto, “Real Time Imaging of Laser Intervention with Optical Coherence Tomography,” in *Biomedical Optical Spectroscopy and Diagnostics / Therapeutic Laser Applications*, vol. 22, 1998.

[152] S. A. Boppart, F. T. Nguyen, A. M. Zysk, E. J. Chaney, J. G. Kotynek, U. J. Oliphant, F. J. Bellafiore, K. M. Rowland, and P. A. Johnson, “Coherent optical imaging and guided interventions in breast cancer: translating technology into clinical applications,” in *Proc. SPIE 6991, Biophotonics: Photonic Solutions for Better Health Care.*, J. Popp, W. Drexler, V. V. Tuchin, and D. L. Matthews, Eds., vol. 6991. Spie, Apr. 2008, pp. 699 102–699 102–12.

[153] I. N. Fleming, S. Voros, B. Vagvolgyi, Z. Pezzementi, J. Handa, and G. D. Hager, “Intraoperative Visualization of Anatomical Targets in Retinal Surgery,” in *IEEE Workshop on Applications of Computer Vision*, 2008, pp. 1–6.

[154] M. Balicki, J.-H. Han, I. Iordachita, P. Gehlbach, J. Handa, R. Taylor, and J. Kang, “Single Fiber Optical Coherence Tomography Microsurgical Instruments for Computer and Robot-Assisted Retinal Surgery,” in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, London, Sep. 2009, pp. 108–115.

[155] X. Liu, X. Li, D. D.-h. Kim, I. Ilev, and J. U. J. Kang, “Fiber-optic Fourier-domain common-path OCT,” *Chinese Optics Letters*, vol. 6, no. 12, pp. 899–901, 2008.

[156] S. Han, M. V. Sarunic, J. Wu, M. Humayun, and C. Yang, "Handheld forward-imaging needle endoscope for ophthalmic optical coherence tomography inspection," *Journal of Biomedical Optics*, vol. 13, no. April, pp. 1–3, 2008.

[157] J. P. Ehlers, Y. K. Tao, S. Farsiu, R. Maldonado, J. A. Izatt, and C. A. Toth, "Integration of a spectral domain optical coherence tomography system into a surgical microscope for intraoperative imaging." *Investigative ophthalmology & visual science*, vol. 52, no. 6, pp. 3153–9, May 2011.

[158] X. Liu, E. Meisne, J.-H. Han, K. Zhang, P. Gehlbach, R. Taylor, and J. U. Kang, "Internal limiting membrane layer visualization and vitreoretinal surgery guidance using a common-path OCT integrated microsurgical tool," in *Proc. SPIE 7550, Ophthalmic Technologies*, F. Manns, P. G. Söderberg, and A. Ho, Eds., vol. 7550, Feb. 2010, pp. 755 003–755 003–4.

[159] X. Liu and J. U. Kang, "Progress toward inexpensive endoscopic high-resolution common-path OCT," in *Proc. SPIE 7559, Optical Fibers and Sensors for Medical Diagnostics and Treatment Applications*, I. Gannot, Ed., vol. 7559, Feb. 2010.

[160] Y. Huang, X. Liu, C. Song, and J. U. Kang, "Motion-compensated hand-held common-path Fourier-domain optical coherence tomography probe for image-guided intervention." *Biomedical optics express*, vol. 3, no. 12, pp. 3105–18, Dec. 2012.

[161] M. Zhao, Y. Huang, and J. U. Kang, "Sapphire ball lens-based fiber probe for common-path optical coherence tomography and its applications in corneal and retinal imaging," *Optics Letters*, vol. 37, no. 23, p. 4835, Nov. 2012.

[162] U. Sharma and J. U. Kang, "Common-path optical coherence tomography with side-viewing bare fiber probe for endoscopic optical coherence tomography." *The Review of scientific instruments*, vol. 78, no. 11, p. 113102, Nov. 2007.

[163] W. Schroeder, K. Martin, and B. Lorensen, *Visualization Toolkit: An Object-Oriented Approach to 3D Graphic*, 4th ed. Kitware, Inc, 2006.

[164] G. Welch and G. Bishop, "An introduction to the Kalman filter," in *SIGGRAPH, Course*, vol. 8. ACM, Inc., 2001, pp. 27 599—-3175.

[165] D. Cabrera Fernandez and H. M. Salinas, "A tissue phantom for investigating volume quantification on retinal images obtained with the Stratus OCT system." *IEEE Engineering in Medicine and Biology Society. Conference*, vol. 2, pp. 1225–8, Jan. 2004.

[166] M. K. Garvin, M. D. Abramoff, R. Kardon, S. R. Russell, X. Wu, and M. Sonka, "Intraretinal layer segmentation of macular optical coherence tomography images using optimal 3-D graph search." *IEEE transactions on medical imaging*, vol. 27, no. 10, pp. 1495–505, Oct. 2008.

[167] M. Haeker, M. Abràmoff, R. Kardon, and M. Sonka, "Segmentation of the surfaces of the retinal layer from OCT images." *International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 9, no. Pt 1, pp. 800–7, Jan. 2006.

[168] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.

[169] X. Liu, Y. Huang, and J. U. Kang, "Distortion-free freehand-scanning OCT implemented with real-time scanning speed variance correction," *Optics Express*, vol. 20, no. 15, p. 16567, Jul. 2012.

[170] J. Ren, J. Wu, E. J. McDowell, and C. Yang, "Manual-scanning optical coherence tomography probe based on position tracking." *Optics letters*, vol. 34, no. 21, pp. 3400–2, Nov. 2009.

[171] A. Ahmad, S. G. Adie, E. J. Chaney, U. Sharma, and S. a. Boppart, "Cross-correlation-based image acquisition technique for manually-scanned optical coherence tomography." *Optics express*, vol. 17, no. 10, pp. 8125–36, May 2009.

[172] E. J. Seibel and Q. Y. J. Smithwick, "Unique features of optical scanning, single fiber endoscopy." *Lasers in surgery and medicine*, vol. 30, no. 3, pp. 177–83, Jan. 2002.

[173] E. J. Seibel, C. M. Brown, J. A. Dominitz, and M. B. Kimmey, "Scanning single fiber endoscopy: a new platform technology for integrated laser imaging, diagnosis, and future therapies." *Gastrointestinal endoscopy clinics of North America*, vol. 18, no. 3, pp. 467–78, viii, Jul. 2008.

[174] S. Moon, S.-W. Lee, M. Rubinstein, B. J. F. Wong, and Z. Chen, "Semi-resonant operation of a fiber-cantilever piezotube scanner for stable optical coherence tomography endoscope imaging." *Optics express*, vol. 18, no. 20, pp. 21 183–97, Sep. 2010.

[175] Y. Chen, A. D. Aguirre, P.-L. Hsiung, S. Desai, P. R. Herz, M. Pedrosa, Q. Huang, M. Figueiredo, S.-W. Huang, A. Koski, J. M. Schmitt, J. G. Fujimoto, and H. Mashimo, "Ultrahigh resolution optical coherence tomography of Barrett's esophagus: preliminary descriptive clinical study correlating images with histology." *Endoscopy*, vol. 39, no. 7, pp. 599–605, Jul. 2007.

[176] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proceedings of the 1968 23rd ACM national conference on -*. New York, New York, USA: ACM Press, 1968, pp. 517–524.

[177] M. Mujat, B. H. Park, B. Cense, T. C. Chen, and J. F. de Boer, "Autocalibration of spectral-domain optical coherence tomography spectrometers for in vivo quantitative retinal nerve fiber layer birefringence determination." *Journal of biomedical optics*, vol. 12, no. 4, p. 041205, 2007.

[178] Y.-R. Chen, B. Sun, T. Han, Y.-F. Kong, C.-H. Xu, P. Zhou, X.-F. Li, S.-Y. Wang, Y.-X. Zheng, and L.-Y. Chen, "Densely folded spectral images of the CCD spectrometer working in the full 200-1000nm wavelength range with high resolution," *Optics Express*, vol. 13, no. 25, p. 10049, 2005.

[179] C. Ding, P. Bu, X. Wang, and O. Sasaki, "A new spectral calibration method for Fourier domain optical coherence tomography," *Optik - International Journal for Light and Electron Optics*, vol. 121, no. 11, pp. 965–970, Jun. 2010.

[180] X. Liu, M. Balicki, R. H. Taylor, and J. U. Kang, "Automatic online spectral calibration of Fourier-domain OCT for robotic surgery." in *Proceedings of SPIE*, vol. 7890, Jan. 2011.

[181] J. U. Kang, J.-H. Han, X. Liu, K. Zhang, C. G. Song, and P. Gehlbach, “Endoscopic Functional Fourier Domain Common-Path Optical Coherence Tomography for Microsurgery,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 16, no. 4, pp. 781–792, Jul. 2010.

[182] K. Zhang, W. Wang, J. Han, and J. U. Kang, “A surface topology and motion compensation system for microsurgery guidance and intervention based on common-path optical coherence tomography,” *IEEE transactions on bio-medical engineering*, vol. 56, no. 9, pp. 2318–2321, 2009.

[183] K. Zhang and J. U. Kang, “Common-path low-coherence interferometry fiber-optic sensor guided microincision.” *Journal of biomedical optics*, vol. 16, no. 9, p. 095003, Sep. 2011.

[184] C. Song, P. L. Gehlbach, and J. U. Kang, “Active tremor cancellation by a “Smart” handheld vitreoretinal microsurgical tool using swept source optical coherence tomography,” *Optics Express*, vol. 20, no. 21, p. 23414, Sep. 2012.

[185] C. Song, D. Y. Park, P. L. Gehlbach, S. J. Park, and J. U. Kang, “Fiber-optic OCT sensor guided "SMART" micro-forceps for microsurgery.” *Biomedical optics express*, vol. 4, no. 7, pp. 1045–50, Jul. 2013.

[186] C. Song, P. L. Gehlbach, and J. U. Kang, “Ball lens fiber optic sensor based smart handheld microsurgical instrument,” vol. 8576, pp. 85 760I–85 760I–6, Mar. 2013.

[187] J. Berger, M. E. Leventon, N. Hata, W. Wells, and R. Kikinis, “Design Considerations for a Computer-Vision-Enabled Ophthalmic Augmented Reality Environment,” *In Lecture Notes in Computer Science*, vol. 1205, pp. 399—-408, 1997.

[188] J. W. Berger and B. Madjarov, “Augmented Reality Fundus Biomicroscopy,” *Arch Ophthalmol*, vol. 119, 2001.

[189] M. Balicki, R. Richa, B. Vagvolgyi, P. Kazanzides, P. Gehlbach, J. Handa, J. Kang, and R. Taylor, “Interactive OCT Annotation and Visualization for Vitreoretinal Surgery,” in *MICCAI Workshop on Augmented Environments for Computer-Assisted Interventions*, C. Linte, Ed. Heidelberg: Springer, 2012, pp. pp. 142—-152.

[190] T. Kim, P. M. Zimmerman, M. J. Wade, and C. a. Weiss, “The effect of delayed visual feedback on telerobotic surgery.” *Surgical endoscopy*, vol. 19, no. 5, pp. 683–6, May 2005.

[191] M. Lum, J. Rosen, T. Lendvay, M. Sinanan, and B. Hannaford, “Effect of time delay on telesurgical performance,” in *IEEE International Conference on Robotics and Automation.* Ieee, May 2009, pp. 4246–4252.

[192] D. E. Becker, A. Can, J. N. Turner, H. L. Tanenbaum, and B. Roysam, “Image processing algorithms for retinal montage synthesis, mapping, and real-time location determination.” *IEEE transactions on bio-medical engineering*, vol. 45, no. 1, pp. 105–18, Jan. 1998.

[193] H. Shen, C. V. Stewart, B. Roysam, G. Lin, and H. L. Tanenbaum, “Frame-Rate Spatial Referencing Based on Invariant Indexing and Alignment with Application to Online Retinal Image Registration æ,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 3, pp. 379–384, 2003.

[194] S. F. Barrett, M. R. Jerath, H. G. R. III, and A. J. Welch, "Digital tracking and control of retinal images," *Optical Engineering*, vol. 33, no. 1, p. 150, Jan. 1994.

[195] R. Richa, B. Vágvölgyi, M. Balicki, G. D. Hager, and R. H. Taylor, "Hybrid tracking and mosaicking for information augmentation in retinal surgery." in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 15, no. Pt 1, Jan. 2012, pp. 397–404.

[196] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*, 2006.

[197] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[198] B. C. Becker, R. a. MacLachlan, L. a. Lobes, and C. N. Riviere, "Semiautomated intraocular laser surgery using handheld instruments." *Lasers in surgery and medicine*, vol. 42, no. 3, pp. 264–73, Mar. 2010.

[199] Z. Pezzementi, "Articulated object tracking by rendering consistent appearance parts," in *IEEE International Conference on Robotics and Automation.* Ieee, May 2009, pp. 3940–3947.

[200] R. Rogério, B. Marcin, M. Eric, S. Raphael, T. Russell, and H. Gregory, "Visual tracking of surgical tools for proximity detection in retinal surgery," in *Information Processing In Computer-Assisted Interventions*, 2011, pp. 55–66.

[201] R. Richa, M. Balicki, R. Sznitman, E. Meisner, R. Taylor, and G. Hager, "Vision-based proximity detection in retinal surgery." *IEEE transactions on bio-medical engineering*, Jun. 2012.

[202] R. Sznitman, A. Basu, R. Richa, J. Handa, P. Gehlbach, R. H. Taylor, B. Jedynak, and G. D. Hager, "Unified detection and tracking in retinal microsurgery." in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 14, no. Pt 1, Jan. 2011, pp. 1–8.

[203] R. Sznitman, K. Ali, R. Richa, R. H. Taylor, G. D. Hager, and P. Fua, "Data-Driven Visual Tracking in Retinal Microsurgery," *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, vol. 15, no. Pt 2, pp. 568–75, Jan. 2012.

[204] M.-H. Yang and N. Ahuja, "Gaussian mixture model for human skin color and its applications in image and video databases," in *Its Application in Image and Video Databases." Proceedings of SPIE*, M. M. Yeung, B.-L. Yeo, and C. A. Bouman, Eds., Dec. 1998, pp. 458–466.

[205] B. C. Becker, C. R. Valdivieso, J. Biswas, L. A. Lobes, and C. N. Riviere, "Active guidance for laser retinal surgery with a handheld instrument." in *IEEEE Engineering in Medicine and Biology Society. Conference*, vol. 2009, Jan. 2009, pp. 5587–90.

[206] C. E. Reiley, T. Akinbiyi, D. Burschka, D. C. Chang, A. M. Okamura, and D. D. Yuh, "Effects of visual force feedback on robot-assisted surgical task performance." *The Journal of thoracic and cardiovascular surgery*, vol. 135, no. 1, pp. 196–202, Jan. 2008.

[207] J. C. Gwilliam, M. Mahvash, B. Vagvolgyi, A. Vacharat, D. D. Yuh, and A. M. Okamura, "Effects of haptic and graphical force feedback on teleoperated palpation," in *IEEE International Conference on Robotics and Automation.* Ieee, May 2009, pp. 677–682.

[208] T. Yamamoto, B. Vagvolgyi, K. Balaji, L. L. Whitcomb, and A. M. Okamura, "Tissue Property Estimation and Graphical Display for Teleoperated Robot-Assisted Surgery," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 4239–4245.

[209] P. Mountney, S. Giannarou, D. Elson, and G.-Z. Yang, "Optical biopsy mapping for minimally invasive cancer screening." *International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 12, no. Pt 1, pp. 483–90, Jan. 2009.

[210] S. Atasoy, D. Mateus, A. Meining, G.-z. Yang, and N. Navab, "Endoscopic Video Manifolds for Targeted Optical Biopsy," *IEEE transactions on medical imaging*, no. c, pp. 1–17, 2011.

[211] D. Salomon, *Data Compression.* London: Springer London, 2007.

[212] B. C. Becker, R. a. Maclachlan, G. D. Hager, and C. N. Riviere, "Handheld Micromanipulation with Vision-Based Virtual Fixtures." in *IEEE International Conference on Robotics and Automation*, vol. 2011, Aug. 2011, pp. 4127–4132.

[213] H. C. Lin, I. Shafran, T. E. Murphy, A. M. Okamura, D. D. Yuh, and G. D. Hager, "Automatic detection and segmentation of robot-assisted surgical motions." in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 8, no. Pt 1, Jan. 2005, pp. 802–10.

Vita

Marcin Balicki received a BSc in Interdisciplinary Engineering (2001) and Masters in Mechanical Engineering from Cooper Union (2004), where he was also an Adjunct Professor, teaching courses in engineering design and prototyping, engineering graphics, and product development. While teaching, he became interested in medical devices and joined the Minimally Invasive Surgery Lab, NYU School of Medicine. There he led software and hardware development of a novel navigation system for knee replacement surgery. For his PhD at The Johns Hopkins University, he has been working with a bioengineering research team to develop a breakthrough microsurgical system that incorporates new robotic manipulators, intra-ocular sensing devices, and visualization techniques to address the extreme challenges of vitreoretinal surgery. The goal of this system is to enable surgeons to perform currently impossible treatments, while also improving the safety and success rates of existing vitreoretinal procedures. Its ultimate benefit will be the millions of patients who suffer from blindness and difficult-to-treat eye conditions. His research interests include medical robotics, human-machine interfaces, software engineering and system integration.