MACHINE LEARNING FOR BEAMFORMING IN AUDIO, ULTRASOUND, AND RADAR

by

Arun Asokan Nair

A dissertation submitted to Johns Hopkins University

in conformity with the requirements for the degree of

Doctor of Philosophy

Baltimore, Maryland

July, 2021

Abstract

Multi-sensor signal processing plays a crucial role in the working of several everyday technologies, from correctly understanding speech on smart home devices to ensuring aircraft fly safely. A specific type of multi-sensor signal processing called beamforming forms a central part of this thesis. Beamforming works by combining the information from several spatially distributed sensors to directionally filter information, boosting the signal from a certain direction but suppressing others. The idea of beamforming is key to the domains of audio, ultrasound, and radar.

Machine learning is the other central part of this thesis. Machine learning, and especially its sub-field of deep learning, has enabled breakneck progress in tackling several problems that were previously thought intractable. Today, machine learning powers many of the cutting edge systems we see on the internet for image classification, speech recognition, language translation, and more.

In this dissertation, we look at beamforming pipelines in audio, ultrasound, and radar from a machine learning lens and endeavor to improve different parts of the pipelines using ideas from machine learning. We start off in the audio domain and derive a machine learning inspired beamformer to

tackle the problem of ensuring the audio captured by a camera matches its visual content, a problem we term audiovisual zooming. Staying in the audio domain, we then demonstrate how deep learning can be used to improve the perceptual qualities of speech by denoising speech clipping, codec distortions, and gaps in speech.

Transitioning to the ultrasound domain, we improve the performance of short-lag spatial coherence ultrasound imaging by exploiting the differences in tissue texture at each short lag value by applying robust principal component analysis. Next, we use deep learning as an alternative to beamforming in ultrasound and improve the information extraction pipeline by simultaneously generating both a segmentation map and B-mode image of high quality directly from raw received ultrasound data.

Finally, we move to the radar domain and study how deep learning can be used to improve signal quality in ultra-wideband synthetic aperture radar by suppressing radio frequency interference, random spectral gaps, and contiguous block spectral gaps. By training and applying the networks on raw single-aperture data prior to beamforming, it can work with myriad sensor geometries and different beamforming equations, a crucial requirement in synthetic aperture radar.

Thesis Committee

Readers

Trac D. Tran (Primary Reader, Advisor)
Professor
Department of Electrical and Computer Engineering
Johns Hopkins Whiting School of Engineering

Muyinatu A. Lediju Bell (Secondary Reader, Advisor)
John C. Malone Assistant Professor
Department of Electrical and Computer Engineering
Johns Hopkins Whiting School of Engineering

Dissertation Defense Committee Member

Vishal Patel
Assistant Professor
Department of Electrical and Computer Engineering
Johns Hopkins Whiting School of Engineering

Acknowledgments

I would first like to thank my advisors Dr. Trac D. Tran and Dr. Muyinatu A. Lediju Bell for their advice over the course of my Ph.D. journey. Dr. Tran has always been supportive of my choices to explore what I found interesting, offering stellar guidance on interesting ideas and avenues I should consider. His openness, kindness, wisdom, and knowledge have been wonderful constants over the years of my Ph.D., something I will be forever grateful for and qualities I will always aspire to. I would also like to thank Dr. Bell for all the guidance she has given me. She has always been available to talk, taught me a lot about how to better structure my work, introduced me to many different and productive research questions, inspired me to be more driven and detail-oriented, earnestly tried to assist me however she could, and wowed me with her ability to manage many different projects and tasks. This thesis would not be possible without her help.

I would like to thank Dr. Vishal Patel for serving on my dissertation defense and thesis proposal committees. His suggestions and guidance have helped strengthen this dissertation. In addition, I am thankful to Dr. Carey Priebe and Dr. Daniel Robinson for serving on my graduate board oral examination committee and imbibing in me a love for statistics and optimization,

respectively.

During my Ph.D., I was fortunate to intern with some excellent industry research groups. During my internship with the computational imaging group at Snapchat Research NYC, my mentor, Dr. Austin Reiter, was an awesome collaborator who supported me however he could, day in and day out. I am also grateful to Dr. Shree Nayar, my manager, for teaching me so many things that have all made me a better, more well-rounded researcher. I would also like to thank Dr. Changxi Zheng, Marian Pho, and Karl Bayer. I would be remiss if I didn't also mention my co-interns, Dr. SRV Vishwanath, Dr. Amit Kumar, and Chang Xiao, who helped me so much both during and outside of work. The myriad experiences we shared and the daily lunch trips we took exploring NYC are very fond memories.

I was also lucky to have the opportunity to intern with the Microsoft Applied Sciences group. I enjoyed the discussions I had with my mentor, Dr. Kazuhito Koishida, who taught me so much about deep learning for speech enhancement and practical aspects of deploying deep learning models to devices. He was always patient, understanding, and overall a true pleasure to work with. Dr. Dung Tran, formerly in the DSP lab with me as well, helped me get settled by always being approachable, setting aside some time every week to speak with me at length. I would also like to thank Dr. Oscar Chang, my co-intern, Dr. Saeed Amizadeh, Asing Chang, and Uros Batricevic, the other members of the group, for their support and the lovely weekly tea-time discussions we had on myriad topics.

I am very thankful to the other members of the labs I am a part of for

making my time at JHU so much better. From the DSP lab, I would like to thank Dr. Xiaoxia Sun, Dr. Dung Tran, Dr. Tao Xiong, Dr. Xiang Xiang, Dr. Luoluo Liu, Dr. Akshay Rangamani, Dr. Sonia Joy, Yang Jiao, and Minh Bui. From the PULSE lab, I would like to thank Derek Allman, Joshua Shubert, Michelle Graham, Alycen Wiacek, Eduardo Gonzalez, Mardava Gubbi, Theron Palmer, Kendra Washington, Kelley Kempski, Justina Huang, and Dr. Manish Bhatt. I have learned so much from each and every one of you.

I am also grateful to my ECE family – Niharika Shimona D'Souza, for the many conversations we have had on matters of both fact and fiction, Jordi Abante, for making the many academic and non-academic experiences we've shared better, Raghavendra Reddy Pappagari, for being a source of support by knowing just how to make me feel better, and Ranjani Srinivasan, for the many discussions we have had from which I have always learned something.

I have had the good fortune to count on some really good friends during my time in JHU. Karuna Agarwal, Mardava Gubbi, and Ratan Sadanand have supported me through thick and thin and I felt better knowing they always had my back. I would also like to thank Anjali Nelliat, Bhagyashree Gubbi, Chin-Fu Liu, Chris Weng, Radhika Rajaram, Rocky Wang, Rupini Shukla, and Stephanie Hao for each making my journey better.

I am also grateful to my close friends from IIT Madras, Aahlad Manas and Arjun Bhagoji, for their continued support from far away. Mardava Gubbi and I always look forward to our online D&D sessions which seem to help me roll natural 20s and not natural 1s in life.

Finally, I would not be where I am today without the continual encouragement of my family. My parents, Asokan Nair and Anjana Nair, have always provided me with unwavering support and a readiness to always lend an ear through both the ups and the downs of my Ph.D. journey. My sisters, Anupama Nair and Anuradha Nair, on the cusp of embarking on their own journeys through college, allowed me to feel close to home from the other side of the planet by excitedly telling me about their lives and listening to me talk about mine. My grandparents – Lalitha Nayar, Leelamani Pillai, M Gopalakrishnan Nayar, and P Bhaskaran Nair – always encouraged me to be curious about the world around me and entertained my never-ending stream of *kinnaram* questions.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Life today has become inextricably linked with the many sensors working in concert in our environment. When we wake up and check our phone for the day's weather, our phone communicates with a cell tower miles away to get us the information (Roh et al., 2014). This is made possible by large arrays of wireless transmitters and receivers on the cell tower that work together to beam information across large distances. Or perhaps one possesses a smart home device like an Amazon Echo or Google Home that one queries for this information. Multiple microphones working in concert on the device help boost the signal quality before handing it off to automatic speech recognition and natural language understanding algorithms in the cloud (Chhetri et al., 2018).

The challenge in the above scenarios is how to efficiently process the data from the multiple sources. The discipline of signal processing that studies processing sensory data from multiple spatially distributed sources jointly for directional signal transmission and reception is termed beamforming (Van Trees, 2004) and forms a crucial component of working systems in fields as

diverse as audio, ultrasound, radar, and wireless communication.

In a beamformer, the multiple sensors are not just giving us independent observations of the quantity of interest, but as the sensors are spatially distributed they collect spatial samples of the propagating wave fields (Van Veen and Buckley, 1988). One can exploit this geometric knowledge of the system under study to further enhance performance and solve tasks impossible to solve with a single omni-directional sensor. Such tasks include performing spatial filtering by boosting signals arriving from a single target point (or direction) and attenuating all other interfering signals from other points (or directions). This allows for the successful separation of multiple sources transmitting in the same spectrum as long as they are spatially separated, a scenario of operation common to audio, ultrasound, radar, and wireless communication.

Recently, the field of machine learning, and its sub-field of deep learning in particular, has become very popular due to the breakthroughs it has enabled in tackling several difficult problems such as large scale image classification (He et al., 2016), automatic speech recognition (Chan et al., 2016), natural language translation (Vaswani et al., 2017), and others. Machine learning is succinctly defined by Tom Mitchell as "the study of computer algorithms that improve automatically through experience" (Mitchell et al., 1997) and thus it should be no surprise that machine learning methods are the driving force behind scaling algorithms to the big data (Qiu et al., 2016) regime where datasets can contain well over a billion samples and models can contain well over a trillion learnable parameters (Fedus, Zoph, and Shazeer, 2021).

Motivated by the success of machine learning, there have been many attempts to combine machine learning ideas with beamforming. Given that there are multiple points in the processing pipeline of beamforming systems where machine learning can be used to improve overall performance, we divide these attempts into the following categories:

1. **Machine learning prior to beamforming:** Machine learning models can be applied to the raw signal received in sensor elements prior to the beamforming step. In the audio domain, it is popular to use machine learning on raw multichannel audio to set the beamformer weights – either directly as in Xiao et al., 2016 and Li et al., 2016 or by learning intermediate representations which plug in to standard beamformers (Erdogan et al., 2016; Heymann, Drude, and Haeb-Umbach, 2016; Ceolini and Liu, 2019). Audio dereverberation using deep networks (Kinoshita et al., 2017) prior to beamforming is also popular. In ultrasound, machine learning has been used on raw pre-beamformed ultrasound data to compress it for wireless transmission before decompressing it for beamforming (Perdios et al., 2017) and to interpolate sub-sampled raw data prior to beamforming (Yoon et al., 2018). In the radar field, Elbir, Mishra, and Eldar, 2019 employed a convolutional neural network to perform cognitive radar antenna selection and Nguyen, Tran, and Tran, 2019 used a generative adversarial network to denoise raw radar data, both prior to beamforming.

2. **Machine learning to replace beamforming:** Machine learning models can also be trained to replace the beamforming model in several

pipelines. For example, in the audio case, Tolooshams et al., 2020 introduced a channel-attention mechanism inside the deep network while Tzirakis, Kumar, and Donley, 2021 viewed each audio channel as a node in a graph neural network to capture spatial correlations between different channels and replace the beamforming step for multichannel speech enhancement. In the ultrasound field, Simson et al., 2018 proposed the DeepFormer network to directly reconstruct high quality ultrasound images from raw sub-sampled data bypassing beamforming, while Vedula et al., 2018 instead expressed the ultrasound beamforming operation as a grid resampling operation using a spatial transformer network (Jaderberg et al., 2015). Deep learning has also been used to form high quality images from delayed input data directly, replacing beamforming (Hyun et al., 2019; Luijten et al., 2019). In the closely related field of photoacoustic imaging, Allman, Reiter, and Bell, 2018 used a deep network to directly perform source detection and artifact removal on raw received data, bypassing the beamforming step. In the radar field, to the best of our knowledge, there has surprisingly not been work on this topic. This is likely due to the wide variance in radar sensor geometries and scenes as a result of the popularity of synthetic aperture radar. Any model replacing beamforming will likely be restricted to only a certain geometry and scene, a much more crippling impediment in radar than audio and ultrasound.

3. **Machine learning post-beamforming:** Machine learning can be applied on the post-beamformed data as well. In the audio domain, there is a

large body of research into single channel speech enhancement (Abdulbaqi, Gu, and Marsic, 2019; Hu et al., 2020; Bulut and Koishida, 2020), which can be viewed either as the scenario of having only one microphone available or having access to only post-beamformed data. In ultrasound, machine learning models have applied on beamformed ultrasound data to improve image quality (Gasse et al., 2017; Perdios et al., 2019), perform breast mass segmentation (Kumar et al., 2018), and implement ultrasound-based robotic visual servoing (Mebarki, Krupa, and Chaumette, 2010). In radar, applications where machine learning has been used post-beamforming include classification (Geng et al., 2015), target recognition (Ding et al., 2016), and denoising (Wang, Zhang, and Patel, 2017), to name just a few.

Note however that these categories need not be mutually exclusive – there are several works that extend across the boundaries of the above categorization e.g. Cauchi et al., 2015; Cheng and Bao, 2020.

1.1 Thesis Outline and Contributions

In this thesis, we examine several scenarios in the audio, ultrasound, and radar domains where machine learning can be leveraged to improve signal processing in systems involving beamforming. Our contributions address problems in all three of the categories enumerated prior and are summarized as follows:

1. We start off in the audio domain and tackle the problem of ensuring

the audio captured by a camera matches its visual content, a problem we term audiovisual zooming. While traditional beamforming formulations are designed to steer in a single direction (or few directions), we demonstrate how, inspired by the simple linear discriminant analysis formulation from machine learning, we are able to elegantly derive and analyze a beamformer for audiovisual zooming that enhances incoming signal from the entire field of view of the camera while suppressing audio originating from outside it. We present this work in Chapter 2.

2. Continuing in the audio domain, we then showcase how deep learning can be used to eliminate speech clipping, codec distortions, and gaps in speech to improve the perceptual quality of single channel speech. Through this study, we reveal the importance of recovering the phase of the speech which is traditionally ignored in single channel speech enhancement but forms the foundation of beamforming. This work is presented in Chapter 3.

3. Moving to the ultrasound domain, we showcase how algorithmic advances in machine learning can be applied to improve ultrasound image quality. We improve the performance of short-lag spatial coherence ultrasound imaging by noting the step of directly summing across the lags can be improved. We instead consider the content of images formed with different lags and exploit the differences in tissue texture at each short-lag value by weighting the addition of lag values and by applying robust principal component analysis. We present this work in Chapter 4.

4. Staying in the ultrasound domain, we demonstrate how deep learning

can function as an alternative to beamforming in ultrasound. We design a fully convolutional neural network that improves the information extraction pipeline in ultrasound by simultaneously generating both a segmentation map and a B-mode image of high quality directly from raw received ultrasound data. This work is presented in Chapter 5.

5. Finally, we move to the radar domain and study the problem radar signal enhancement. Specifically, we investigate how deep learning can be used to improve signal quality in ultra-wideband synthetic aperture radar suffering from radio frequency interference, random spectral gaps, and a contiguous block spectral gap. We design our networks to operate on raw single-aperture data prior to beamforming and by doing so, we show that the same network can work with various sensor geometries, a crucial requirement for successful deployment to synthetic aperture radar scenarios. We present this work in Chapter 6.

This thesis is built from the contents of several peer-reviewed publications and thus many sections include text from the original manuscripts in unaltered form. Consequently, there might at first glance appear to be overlap between material in different chapters but this overlap is required to maintain the self-consistency and flow of each chapter, and to cater to the nuances of each problem being addressed. For example, signal-to-noise ratio (SNR) is redefined in almost every single chapter but the definition of SNR varies depending on the task and domain at hand.

References

Roh, Wonil, Ji-Yun Seol, Jeongho Park, Byunghwan Lee, Jaekon Lee, Yungsoo Kim, Jaeweon Cho, Kyungwhoon Cheun, and Farshid Aryanfar (2014). "Millimeter-wave beamforming as an enabling technology for 5G cellular communications: Theoretical feasibility and prototype results". In: *IEEE communications magazine* 52.2, pp. 106–113.

Chhetri, Amit, Philip Hilmes, Trausti Kristjansson, Wai Chu, Mohamed Mansour, Xiaoxue Li, and Xianxian Zhang (2018). "Multichannel audio front-end for far-field automatic speech recognition". In: *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, pp. 1527–1531.

Van Trees, Harry L (2004). *Optimum array processing: Part IV of detection, estimation, and modulation theory*. John Wiley & Sons.

Van Veen, Barry D and Kevin M Buckley (1988). "Beamforming: A versatile approach to spatial filtering". In: *IEEE assp magazine* 5.2, pp. 4–24.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Chan, William, Navdeep Jaitly, Quoc Le, and Oriol Vinyals (2016). "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 4960–4964.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *arXiv preprint arXiv:1706.03762*.

Mitchell, Tom M et al. (1997). "Machine learning". In:

Qiu, Junfei, Qihui Wu, Guoru Ding, Yuhua Xu, and Shuo Feng (2016). "A survey of machine learning for big data processing". In: *EURASIP Journal on Advances in Signal Processing* 2016.1, pp. 1–16.

Fedus, William, Barret Zoph, and Noam Shazeer (2021). "Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity". In: *arXiv preprint arXiv:2101.03961*.

Xiao, Xiong, Shinji Watanabe, Hakan Erdogan, Liang Lu, John Hershey, Michael L Seltzer, Guoguo Chen, Yu Zhang, Michael Mandel, and Dong Yu (2016). "Deep beamforming networks for multi-channel speech recognition". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 5745–5749.

Li, Bo, Tara N Sainath, Ron J Weiss, Kevin W Wilson, and Michiel Bacchiani (2016). "Neural network adaptive beamforming for robust multichannel speech recognition". In:

Erdogan, Hakan, John R Hershey, Shinji Watanabe, Michael I Mandel, and Jonathan Le Roux (2016). "Improved mvdr beamforming using single-channel mask prediction networks." In: *Interspeech*, pp. 1981–1985.

Heymann, Jahn, Lukas Drude, and Reinhold Haeb-Umbach (2016). "Neural network based spectral mask estimation for acoustic beamforming". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 196–200.

Ceolini, Enea and Shih-Chii Liu (2019). "Combining deep neural networks and beamforming for real-time multi-channel speech enhancement using a wireless acoustic sensor network". In: *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, pp. 1–6.

Kinoshita, Keisuke, Marc Delcroix, Haeyong Kwon, Takuma Mori, and Tomohiro Nakatani (2017). "Neural Network-Based Spectrum Estimation for Online WPE Dereverberation." In: *Interspeech*, pp. 384–388.

Perdios, Dimitris, Adrien Besson, Marcel Arditi, and Jean-Philippe Thiran (2017). "A deep learning approach to ultrasound image recovery". In: *2017 IEEE International Ultrasonics Symposium (IUS)*. Ieee, pp. 1–4.

Yoon, Yeo Hun, Shujaat Khan, Jaeyoung Huh, and Jong Chul Ye (2018). "Efficient b-mode ultrasound image reconstruction from sub-sampled rf data using deep learning". In: *IEEE transactions on medical imaging* 38.2, pp. 325–336.

Elbir, Ahmet M, Kumar Vijay Mishra, and Yonina C Eldar (2019). "Cognitive radar antenna selection via deep learning". In: *IET Radar, Sonar & Navigation* 13.6, pp. 871–880.

Nguyen, Lam, Dung N Tran, and Trac D Tran (2019). "Spectral Gaps Extrapolation for Stepped-Frequency SAR via Generative Adversarial Networks". In: *2019 IEEE Radar Conference (RadarConf)*. IEEE, pp. 1–6.

Toloosham, Bahareh, Ritwik Giri, Andrew H Song, Umut Isik, and Arvindh Krishnaswamy (2020). "Channel-attention dense u-net for multichannel speech enhancement". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 836–840.

Tzirakis, Panagiotis, Anurag Kumar, and Jacob Donley (2021). "Multi-Channel Speech Enhancement using Graph Neural Networks". In: *arXiv preprint arXiv:2102.06934*.

Simson, Walter, Magdalini Paschali, Nassir Navab, and Guillaume Zahnd (2018). "Deep learning beamforming for sub-sampled ultrasound data". In: *2018 IEEE International Ultrasonics Symposium (IUS)*. IEEE, pp. 1–4.

Vedula, Sanketh, Ortal Senouf, Grigoriy Zurakhov, Alex Bronstein, Oleg Michailovich, and Michael Zibulevsky (2018). "Learning beamforming in ultrasound imaging". In: *arXiv preprint arXiv:1812.08043*.

Jaderberg, Max, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu (2015). "Spatial transformer networks". In: *arXiv preprint arXiv:1506.02025*.

Hyun, Dongwoon, Leandra L Brickson, Kevin T Looby, and Jeremy J Dahl (2019). "Beamforming and speckle reduction using neural networks". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 66.5, pp. 898–910.

Luijten, Ben, Regev Cohen, Frederik J de Bruijn, Harold AW Schmeitz, Massimo Mischi, Yonina C Eldar, and Ruud JG van Sloun (2019). "Deep learning for fast adaptive beamforming". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 1333–1337.

Allman, Derek, Austin Reiter, and Muyinatu A Lediju Bell (2018). "Photoacoustic source detection and reflection artifact removal enabled by deep learning". In: *IEEE transactions on medical imaging* 37.6, pp. 1464–1477.

Abdulbaqi, Jalal, Yue Gu, and Ivan Marsic (2019). "RHR-Net: A residual hourglass recurrent neural network for speech enhancement". In: *arXiv preprint arXiv:1904.07294*.

Hu, Yanxin, Yun Liu, Shubo Lv, Mengtao Xing, Shimin Zhang, Yihui Fu, Jian Wu, Bihong Zhang, and Lei Xie (2020). "Dccrn: Deep complex convolution recurrent network for phase-aware speech enhancement". In: *arXiv preprint arXiv:2008.00264*.

Bulut, Ahmet E and Kazuhito Koishida (2020). "Low-latency single channel speech enhancement using u-net convolutional neural networks". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 6214–6218.

Gasse, Maxime, Fabien Millioz, Emmanuel Roux, Damien Garcia, Hervé Liebgott, and Denis Friboulet (2017). "High-quality plane wave compounding using convolutional neural networks". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 64.10, pp. 1637–1639.

Perdios, Dimitris, Adrien Besson, Florian Martinez, Manuel Vonlanthen, Marcel Arditi, and Jean-Philippe Thiran (2019). "On problem formulation, efficient modeling and deep neural networks for high-quality ultrasound imaging: Invited presentation". In: *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*. IEEE, pp. 1–4.

Kumar, Viksit, Jeremy M Webb, Adriana Gregory, Max Denis, Duane D Meixner, Mahdi Bayat, Dana H Whaley, Mostafa Fatemi, and Azra Alizad (2018). "Automated and real-time segmentation of suspicious breast masses using convolutional neural network". In: *PloS one* 13.5, e0195816.

Mebarki, Rafik, Alexandre Krupa, and François Chaumette (2010). "2-d ultrasound probe complete guidance by visual servoing using image moments". In: *IEEE Transactions on Robotics* 26.2, pp. 296–306.

Geng, Jie, Jianchao Fan, Hongyu Wang, Xiaorui Ma, Baoming Li, and Fuliang Chen (2015). "High-resolution SAR image classification via deep convolutional autoencoders". In: *IEEE Geoscience and Remote Sensing Letters* 12.11, pp. 2351–2355.

Ding, Jun, Bo Chen, Hongwei Liu, and Mengyuan Huang (2016). "Convolutional neural network with data augmentation for SAR target recognition". In: *IEEE Geoscience and remote sensing letters* 13.3, pp. 364–368.

Wang, Puyang, He Zhang, and Vishal M Patel (2017). "SAR image despeckling using a convolutional neural network". In: *IEEE Signal Processing Letters* 24.12, pp. 1763–1767.

Cauchi, Benjamin, Ina Kodrasi, Robert Rehr, Stephan Gerlach, Ante Jukić, Timo Gerkmann, Simon Doclo, and Stefan Goetze (2015). "Combination of MVDR beamforming and single-channel spectral processing for enhancing noisy and reverberant speech". In: *EURASIP Journal on Advances in Signal Processing* 2015.1, pp. 1–12.

Cheng, Rui and Changchun Bao (2020). "Speech Enhancement Based on Beamforming and Post-Filtering by Combining Phase Information". In: *Proc. Interspeech 2020*, pp. 4496–4500.

Chapter 2

Audiovisual Zooming: What You See is What You Hear

When capturing videos on a mobile platform, often the target of interest is contaminated by the surrounding environment. To alleviate the visual irrelevance, camera panning and zooming provide the means to isolate a desired field of view (FOV). However, the captured audio is still contaminated by signals outside the FOV. This effect is unnatural—for human perception, visual and auditory cues must go hand-in-hand.

We present the concept of *Audiovisual Zooming*, whereby an auditory FOV is formed to match the visual. Our framework is built around the classic idea of beamforming, a computational approach to enhancing sound from a single direction using a microphone array. Yet, beamforming on its own can not incorporate the auditory FOV, as the FOV may include an arbitrary number of directional sources. Inspired by the formulation of linear discriminant analysis (LDA) in machine learning, we formulate our audiovisual zooming as a generalized eigenvalue problem and propose an algorithm for efficient computation on mobile platforms. To inform the algorithmic and physical

implementation, we offer a theoretical analysis of our algorithmic components as well as numerical studies for understanding various design choices of microphone arrays. Finally, we demonstrate audiovisual zooming on two different mobile platforms: a mobile smartphone and a 360° spherical imaging system for video conference settings. The work presented in this chapter was published earlier in Nair et al., 2019.

2.1 Introduction

The camera can tilt, pan, pedestal, dolly, truck, and zoom—to control what the viewer sees. Historically, this rich vocabulary of camera control is only at the professional's disposal. Today, every mobile device is equipped with a compact and light camera, allowing anyone to decide what imagery in what way is to be captured. Whenever one captures a video, audio is also captured, but the vocabulary with which a user can exert control over the audio pales in comparison to user control over the video. No matter where the camera is pointed or how zoomed it is, the sound is always recorded regardless of its incoming direction, be it from behind the camera or somewhere in the view. As a result, the captured video might not match the audio, leading to an unnatural experience.

The problem is that the camera lacks an *auditory field of view*, one that is synchronized with and driven by the camera's optical field of view (FOV). In this work, we introduce the concept of focusing an auditory FOV (see Figure 2.1) to address the problem. We call our concept **Audiovisual Zooming**.

The closest field-of-study to this concept is *Beamforming* (Gannot et al.,

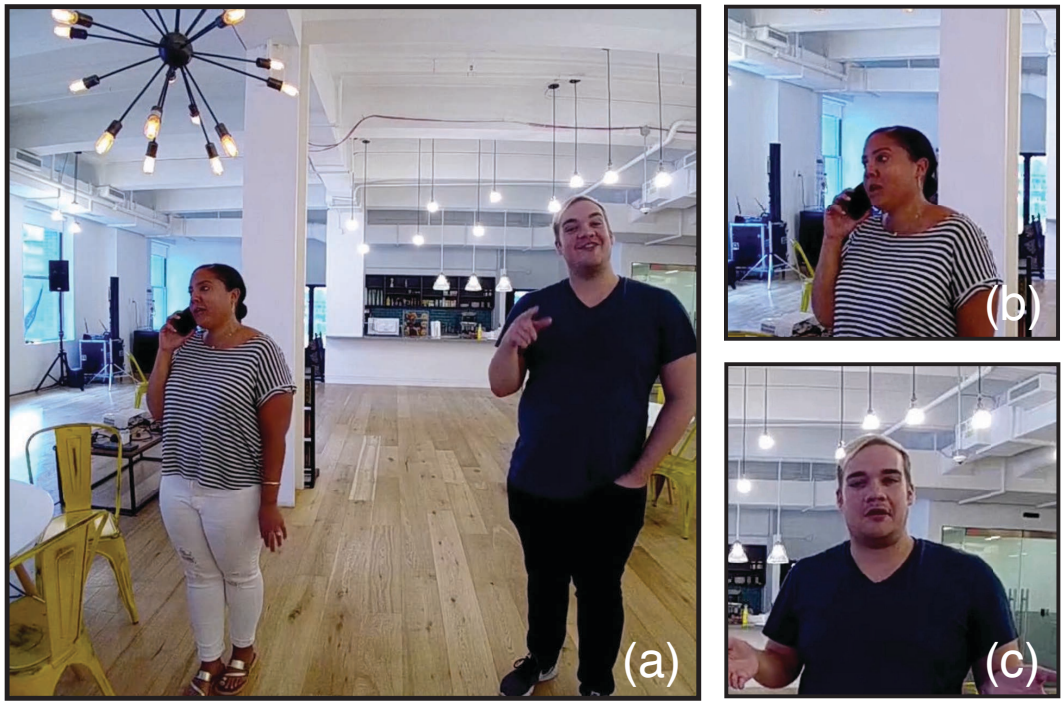


Figure 2.1: Audiovisual zooming. When the camera captures both people (a), we hear them both talk. (b) As the camera zooms in and focuses on the woman, her speech in the captured video is enhanced while the man's speech is suppressed. (c) Then, the camera pans and focuses on the man, in this process his speech becomes more pronounced while the woman's speech fades out. In our system, the camera's FOV synchronizes with its auditory focus—what you see is what you hear (see supplementary video).

2017), a computational technique that constructs a directional microphone by using an array of omnidirectional microphones. Leveraging the different time delays of signals that arrive from different directions, the idea is to linearly combine microphone signals into an output signal boosting the sound coming from a *target direction*, while suppressing everything else for directional sound filtering. In almost all beamforming techniques, the single target direction needs to be specified or estimated, and plays an important role in the mathematical formulation of beamforming. It is this essential notion of

target direction that sets apart our method from the traditional beamforming.

Our audiovisual zooming requires no target direction. In contrast, we introduce auditory FOV, which defines a directional region (i.e., a solid angle area) consistent with the camera's optical FOV. All sounds, no matter how many, coming from within this region are enhanced, while those outside of the region are suppressed. In this way, the captured audio is in synchronization with the captured imagery. In other words, *what you see is what you hear*.

One approach toward this goal is jointly analyzing the captured audio and visual content through deep learning (Ephrat et al., 2018; Zhao et al., 2018; Owens and Efros, 2018). The success of this approach lies in the strong correlation between the motion in captured imagery and the resulting audio, as well as the feasibility of constructing a large training dataset. But often the motion-audio correlation is weak or even undetectable—for example, when the sound source is far from the camera, or occluded by other objects (but still in the FOV). In addition, there may be arbitrary numbers/types of sound sources in the FOV. Constructing a training dataset that covers all these cases quickly becomes intractable, and the resulting deep neural networks are unlikely to run on a low-budget mobile device where the camera often resides.

Technical contributions: In this work, we augment the microphone array and beamforming approaches to enable audiovisual zooming, without learning from training data. Motivated by microphone array beamforming, we view the signals sampled by individual microphones as random variables of some underlying stochastic process. From this perspective, we estimate

two complex-valued matrices, called *spectral matrices*, in frequency domain: one describes the autocorrelation and cross-correlation of microphone signals that come from within the FOV, and the other describes signals coming from outside of the FOV. We show that with these two matrices, the problem of enhancing towards an FOV can be formulated as a generalized eigenvalue problem that can be easily solved on a mobile device. Our approach is not meant to improve beamforming, but rather to enable audiovisual zooming.

To analyze our approach, we derive a theoretical error bound for our spectral matrix estimation, and reveal a connection of the error residual to the performance of the classic minimum variance distortionless response (MVDR) beamformer. Empirically, we conduct simulations to understand how various design parameters affect a microphone array.

These inferences inform our implementation. Our final algorithm is simple and can be easily deployed on mobile devices. We realize the audiovisual zooming system by attaching a planar microphone array to two different mobile imaging platforms: a mobile smartphone and a 360° spherical imaging system for teleconference settings (see Figure 2.2). Finally, we demonstrate our system in a number of use cases.

2.2 Related Work

Our audiovisual zooming is built on classic beamforming. We therefore briefly review related work in this area. We also discuss the difference of our approach from the general idea of audiovisual machine learning approaches.

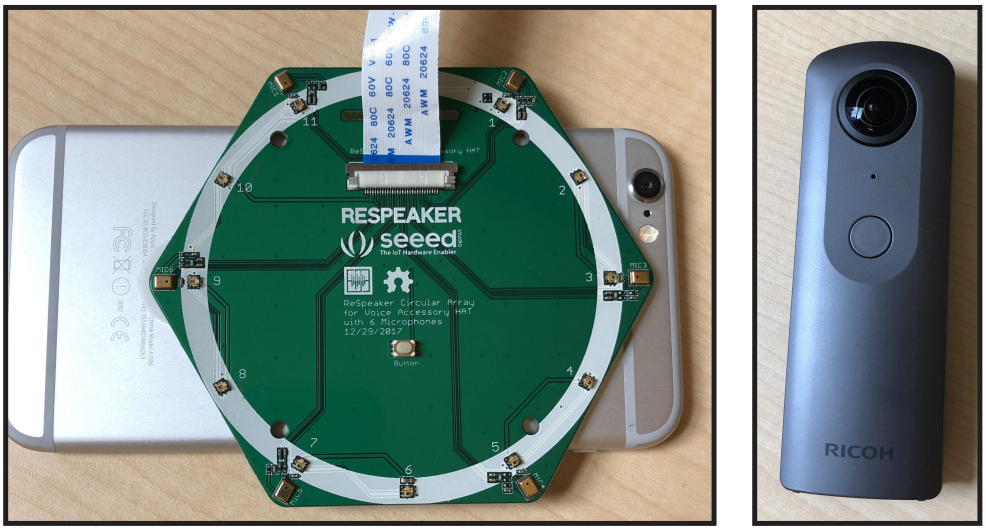


Figure 2.2: Audiovisual zooming is implemented on two mobile platforms: an off-the-shelf planar microphone array (with 6 microphones) is attached to a smartphone [left] and a 360° Ricoh camera [right] to show smartphone and teleconferencing utility.

2.2.1 Beamforming

A rich and mature research field, acoustic beamforming has a long history, dating back to 1970s when Billingsley and Kinns, 1976 invented the microphone antenna called the acoustic telescope. We refer the reader to Michel et al., 2006 for a review of the development of acoustic beamforming techniques and to Gannot et al., 2017 for an exhaustive survey of the state of the field.

In general, the various beamforming methods falls into one of two categories: fixed and adaptive. Fixed beamformers are best summarized by the well-known *Delay-and-Sum* method (Veen and Buckley, 1988; Teutsch, 2007), which delays the signal received by each microphone according to the relative propagation delays from a target direction, and then sums the signals together across the microphones. This serves to enhance the gain of the target direction,

but often does little to suppress anything else.

The seminal work of Capon, 1969 introduces an adaptive, or *data-dependent*, beamforming technique, later known as the Minimum Variance Distortionless Response (MVDR) beamformer (Stoica, Moses, et al., 2005; Trees, 2002). This approach optimizes a set of weights to linearly combine the signals in time-frequency space so as to minimize residual noise and constrain the sound from the desired direction to be undistorted. The robustness of MVDR beamformer is later improved by various extensions such as dynamic loading (Li, Stoica, and Wang, 2003). Since our method is built on the MVDR beamformer, we will briefly review its formulation in §2.3.2. There are also other variants, such as the Linearly Constrained Minimum Variance (LCMV) (Griffiths and Jim, 1982), Principal component (Hung and Turner, 1983; Yu and Yeh, 1995), and Generalized Eigenvalue (Warsitz and Haeb-Umbach, 2007) beamformers, all of which are special cases of a shared underlying optimization framework (Trees, 2002).

All these beamforming techniques have the same goal: enhancing the sound from *a single direction*, and they have no notion of *field of view* (FOV). In contrast, our goal is to enhance all sounds from within an *arbitrary FOV* and suppress everything outside. It is this very difference that requires a different beamforming formulation and thus necessitates the development of a new algorithm.

Recently, a few methods have been proposed to enhance sounds from multiple sources. Thiergart, Kowalczyk, and Habets, 2014 introduced acoustic zoom, wherein all detected sound sources are individually isolated via

direction-of-arrival (DOA) estimation and beamforming, and then combined through a weighting scheme defined by their zooming parameters. Ruochen, Yuhong, and Wei, 2014 used a spherical microphone array and the psychoacoustic theory to model sound perceptions and control audio boosting using camera metadata in so-called B-Format Encoding. A more recent method, Duong et al., 2017, uses MVDR beamforming in three orthogonal directions and chooses the sound from the microphone closest to the target region. Just by choosing a microphone signal, this method does not enhance received sound, and is inherently limited for small form-factor arrays. Another line of work using spherical arrays and cameras together (Mendat et al., 2017; *VisiSonics 5/64 Audio Visual Camera*) and beamforming based on spherical harmonics (Li and Duraiswami, 2007) can enhance multiple sound sources but again requires individual detection and isolation of each source using DOA estimation and beamforming before combining the individual beamformed tracks while also requiring large arrays (8.4cm – 20cm diameter) with large numbers of microphones (32-64) to function. Our method, in contrast, requires no estimation of DOAs and can be implemented using compact microphone arrays.

2.2.2 Audiovisual learning

Recently, a line of work has emerged that combines computer vision and audio via deep learning for speech recognition, separation, and enhancement (Feng et al., 2017; Mroueh, Marcheret, and Goel, 2015; Rivet et al., 2014; Hershey and Casey, 2002). Particularly related to our work, Ephrat et al., 2018 recently

introduced a deep learning model that detects and analyzes facial movements along with learning a mask on Fourier coefficients to mask out desired speech associated with particular facial motion. Zhao et al., 2018 addressed a similar problem of separating the sound of multiple on-screen objects by training a self-supervised model. Owens and Efros, 2018 used a deep neural network to predict whether audio and visual tracks are temporally aligned. Features learned through training are then used to perform an on/off screen speaker separation. Afouras, Chung, and Zisserman, 2018 trained a deep neural network that takes audio and visual cues to denoise speech spectrograms. While impressive, these work require that the visual component of the sound is both visible and has sufficient pixel resolution to capture the appearance and motion. Our work does not rely on any analysis of visual cues, and as such, can enhance sound coming from any FOV even when the motion that produces this sound is occluded or far away from the camera.

2.2.3 Summary

Our method differs from previous works in that 1) no knowledge of DOAs is required, 2) the user may specify any arbitrary FOV to match that of a camera's, and 3) our approach will enhance only the sound from within that FOV and attenuate everything else. *In this way, the camera drives the experience entirely, forcing the focused audio content to match what is being viewed*.

2.3 Theory of Audiovisual Zooming

A cornerstone of our audiovisual zooming system is microphone array beamforming. To understand our algorithm, we start with a brief review of this classic technique.

2.3.1 Microphone array model

We consider a microphone array that consists of M sensors receiving sound from all directions. The time-domain signals captured by microphone i ($i = 1...M$) is

$$y_i(t) = \sum_{s=1}^{S} h_{s \to i}(t) * x_s(t) + n_i(t), \tag{2.1}$$

where $*$ denotes the convolution operator, s indices individual sound sources, $x_s(t)$ is the signals emitted at sound source s, $n_i(t)$ is the noise at microphone i, and $h_{s \to i}(t)$ is the *Acoustic Transfer Function* for source s impinging on microphone i. This transfer function accounts for how the sound propagates from s to i, including both direct and indirect propagation (e.g., reflection and diffraction by the environment).

Because the sound propagation largely depends on its frequency components, the microphone array model is often expressed in time-frequency (T-F) domain (Gannot et al., 2017) through the Short-Time Fourier Transform (STFT). In T-F domain, the convolution operator becomes into a multiplication, and Eq. (2.1) is written as

$$Y_i(n, \omega) = \sum_{s=1}^{S} H_{s \to i}(n, \omega) X_s(n, \omega) + N_i(n, \omega), \tag{2.2}$$

where n and ω index the time frame and the discrete frequency bin, respectively. We then stack the STFT coefficients for all sensors in a vector,

$$\boldsymbol{Y}(n,\omega) = [Y_1(n,\omega),\ldots,Y_M(n,\omega)]^T. \tag{2.3}$$

With these notations, we now briefly review the classic beamforming algorithms, as follows.

2.3.2 Beamforming Briefing

The general idea of beamforming is simple. It linearly combines the input multi-channel signals into a mono-channel signal in T-F domain. Provided a set of frequency-dependent weights $\boldsymbol{w}(\omega) = [w_1(\omega),\ldots,w_M(\omega)]^T$, the linear combination outputs a signal as $\boldsymbol{w}^H(\omega)\boldsymbol{Y}(n,\omega)$, where the superscript H denotes conjugate transpose. By carefully choosing the weights $\boldsymbol{w}$, the resulting signal enhances the sound received from a given *single* direction $\boldsymbol{d}$.

Intuitively, this is possible because the sound signals recorded at different microphones differ in both amplitude and phase. One can choose the weights $\boldsymbol{w}$ to "adjust" the differences such that when the signals are superimposed, they interfere constructively for sound coming from the direction $\boldsymbol{d}$ but destructively for sound from other directions. Numerous algorithms have been devised to estimate the weights $\boldsymbol{w}$. Here we only review the ones that are most relevant to our method, while referring the reader to the textbooks Brandstein and Ward, 2013; Trees, 2002 for a comprehensive introduction.

2.3.2.1 Spectral matrix

A fundamental philosophy in microphone array processing is to model the received signal as a *stochastic process*. Each individual sample $y_i[t]$ of microphone i is assumed to be an outcome of some underlying random process.

An important notion from this vantage point is the *spectral matrix*, an $M \times M$ complex-valued Hermitian matrix, denoted as $\mathsf{R}(\omega)$, describing the frequency-domain signal statistics received by the microphone array. Its diagonal element $\mathsf{R}_{ii}(\omega)$ indicates the *autocorrelation* (in frequency domain) of the impinging signal received by microphone i, that is, the power spectrum of the signal at i. Its off-diagonal element $\mathsf{R}_{ij}(\omega)$ describes the *cross-correlation* of signals received by microphone i and j, reflecting the phase differences between the two microphone signals. In short, the spectral matrix encapsulates information needed for the estimation of $\boldsymbol{w}$—toward constructively enhancing the signal of a given direction.

In practice, $\mathsf{R}(\omega)$ is estimated using the frequency-domain snapshots $\boldsymbol{Y}(n, \omega)$ in (2.3). A simple yet common estimator is

$$\mathsf{R}(\omega) \approx \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{Y}(n, \omega) \boldsymbol{Y}^H(n, \omega), \tag{2.4}$$

from which many improvements have been developed (such as the Forward-Backward averaging (Stoica, Moses, et al., 2005)).

If a set of weights $\boldsymbol{w}(\omega)$ is used to combine the microphone signals in the frequency band ω, it can be shown that the output signal has the power spectrum expressed as $\boldsymbol{w}^H(\omega)\mathsf{R}(\omega)\boldsymbol{w}(\omega)$ (Brandstein and Ward, 2013)). From now on, when there is no confusion, we will ignore the frequency parameter

ω and simply write R and $\boldsymbol{w}$.

2.3.2.2 Minimum Variance Distortionless Response (MVDR) beamformer

In beamforming theory, the spectral matrix R is viewed as a composition of two parts, signal spectral matrix R_s and noise spectral matrix R_n. R_s accounts for the signal solely from the desired direction $\boldsymbol{d}$ (sometimes also called the direction of arrival), while R_n accounts for the *unwanted signals* including both the ambient noise (i.e., N_i in (2.2)) and those from the undesired directions. Note that R might not be a simple summation of R_s and R_n if the unwanted signals and the desired signal are (at least partially) correlated.

The classic MVDR beamformer finds the optimal $\boldsymbol{w}$ in the following sense: it minimizes the power of unwanted signals, while keeping the signal from the desired direction undistorted. This is formally expressed as a constrained optimization problem,

$$\boldsymbol{w}_\mathrm{BF} = \arg\min_{\boldsymbol{w}} \boldsymbol{w}^H \mathsf{R}_\mathrm{n} \mathbf{w}, \quad \text{s.t. } \boldsymbol{w}^H \boldsymbol{v}_d = 1. \tag{2.5}$$

Here the objective function measures the power of unwanted signals in the output. The constraint requires the incoming signal from the direction $\boldsymbol{d}$ to remain undistorted in the output signal. The vector $\boldsymbol{v}_d$, called the *steering vector*, indicates the relative phases of the signal impinging on all M microphones from the desired direction $\boldsymbol{d}$, defined as

$$\boldsymbol{v}_d = \begin{bmatrix} e^{-j\frac{\omega}{c}\boldsymbol{d}^T\boldsymbol{p}_1} & \dots & e^{-j\frac{\omega}{c}\boldsymbol{d}^T\boldsymbol{p}_M} \end{bmatrix}^T, \tag{2.6}$$

where c is the speed of sound, and $\boldsymbol{p}_i$ ($i = 1...M$) is the spatial position

of each microphone in the array. The steering vector describes the relative phase difference of a plane wave sound impinging on the microphones from direction $\boldsymbol{d}$. The intuition behind the constraint in (2.5) is that the weights $\boldsymbol{w}$ need to compensate the received phase differences at the microphones from direction $\boldsymbol{d}$ and thereby constructively combine the signals to boost the signal from $\boldsymbol{d}$.

MVDR beamformer is one of the most widely used beamforming techniques. Provided a single steering direction $\boldsymbol{d}$ and an estimation of R_n, it has realtime performance even on a low-budget mobile device, since the weights can be analytically written as

$$\boldsymbol{w}_\mathrm{BF} = \frac{\boldsymbol{v}_d^H \mathsf{R}_\mathrm{n}^{-1}}{\boldsymbol{v}_d^H \mathsf{R}_\mathrm{n}^{-1} \boldsymbol{v}_d}. \tag{2.7}$$

Oftentimes, however, estimation of R_n from microphone recordings is challenging. An approximation is by replacing R_n in (2.5) with the spectral matrix R, as R can be directly estimated using the recorded signals in (2.4). Then, the optimization objective is to minimize the total output power subject to the constraint in (2.5). This is the so-called *Minimum Power Distortionless Response* (MPDR) beamformer, one that lays the foundation of our audiovisual zooming method.

2.3.3 Beamforming Toward a Field of View

Almost all beamforming techniques require to know a steering direction $\boldsymbol{d}$. Indeed, this single steering direction is pivotal for establishing the constraint in MVDR/MPDR formulation (2.5). However, in our work, we wish to enhance

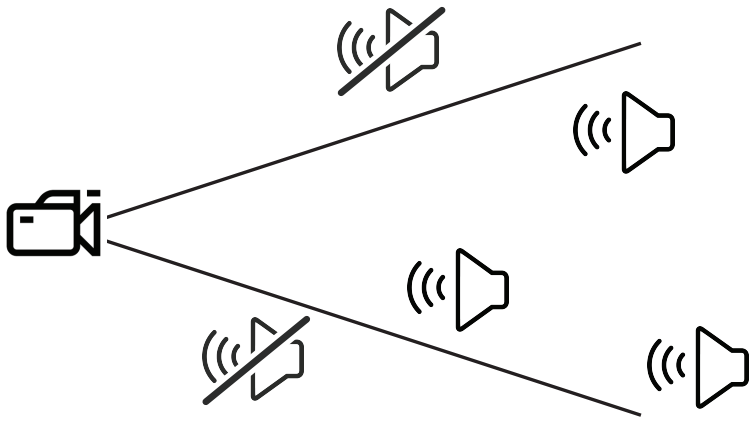


Figure 2.3: Unlike traditional beamforming, our audiovisual zooming system does not rely on a specific target direction. Any sound sources captured by the camera's FOV will be enhanced, while those outside of the FOV are suppressed.

signals toward a field of view (FOV), that is, a *continuous set* of steering directions (Figure 2.3). How to incorporate the FOV in microphone array beamforming is the challenge that we need to overcome.

We determine the beamforming FOV based on the camera's FOV (elaborated in §2.5.3). We also note that the beamforming FOV may vary as the user zooms in/out or pans the camera.

2.3.3.1 Generalized eigenvalue formulation

The starting point of our method is also the spectral matrix (recall §2.3.2.1). Yet, for our purpose of beamforing toward an FOV, the signal and noise spectral matrices, R_s and R_n, must be interpreted in a different way. Now, R_s accounts for all signals coming from directions inside the FOV, while R_n includes signals outside of the FOV. Suppose for now we know both R_s and R_n. We can formulate a beamforming optimization problem by maximizing

the output signal-to-noise (SNR) ratio, namely

$$\boldsymbol{w}_{\text{FOV}} = \arg\max_{\boldsymbol{w}} \frac{\boldsymbol{w}^H \mathsf{R}_{\text{s}} \boldsymbol{w}}{\boldsymbol{w}^H \mathsf{R}_{\text{n}} \boldsymbol{w}}. \tag{2.8}$$

Here the numerator and denominator measure the powers of desired signals and unwanted signals, respectively. This formulation is known in traditional beamforming, although not widely used. This is because it needs the estimation of both R_{s} and R_{n}, and when a single steering direction is considered (e.g., when the desired signal is a plane wave along a direction), this formulation is identical to MVDR beamformer (Trees, 2002)—no need to solve (2.8) directly.

But this formulation is significant for our problem, since it requires no steering direction. Indeed, the desire of steering toward an FOV can be expressed by R_{s}, which can include signals from an arbitrary set of directions. If R_{s} and R_{n} can be robustly estimated, then solving for $\boldsymbol{w}_{\text{FOV}}$ amounts to a simple generalized eigenvalue problem (by noticing that the objective in (2.8) is a generalized Rayleigh quotient):

$$\mathsf{R}_{\text{s}} \boldsymbol{w} = \lambda \mathsf{R}_{\text{n}} \boldsymbol{w}. \tag{2.9}$$

The solution $\boldsymbol{w}_{\text{FOV}}$ in (2.8) is the eigenvector of the maximal eigenvalue.

2.3.3.2 Estimation of signal and noise spectral matrices

The remaining question is how to estimate R_{s} and R_{n} that respect the FOV. Some traditional beamforming methods (such as MVDR) also need to estimate R_{n}, for which a popular approach is by estimating a noise mask in T-F domain (Heymann, Drude, and Haeb-Umbach, 2016). There, a common

assumption is that the desired signal is the speech of a single voice. In other words, it assumes that the desired signal has a T-F structure, which can be inferred and used to estimate the mask by a machine learning model trained over a large speech dataset.

In our problem, the desired signals are those received in the FOV. In stark contrast to the single speech assumption, their structure is unclear, as they might include an arbitrary number of speakers, other types of sound, and even ambient noise coming from the FOV. It is too expensive to construct a sufficient training dataset for a machine learning model producing reasonable masks.

We resort to the MPDR beamformer to estimate R_s and R_n. First, consider a direction $\boldsymbol{\theta}$. The MPDR weights $\boldsymbol{w}_\theta$ for enhancing signals from $\boldsymbol{\theta}$ is expressed in (2.7), where the steering vector $\boldsymbol{v}_d = \boldsymbol{v}_\theta$ is defined in (2.6) and the matrix R_n is replaced by the total spectral matrix R estimated using (2.4). Substituting this expression in $\boldsymbol{w}_\theta^H \mathsf{R} \boldsymbol{w}_\theta$ yields the power spectrum of the MPDR output signal,

$$P(\boldsymbol{\theta}) = \left[\boldsymbol{v}_\theta^H \mathsf{R}^{-1} \boldsymbol{v}_\theta\right]^{-1}. \tag{2.10}$$

Recall that the effect of MPDR beamformer is to boost the signal from direction $\boldsymbol{\theta}$ while suppressing signals from other directions. Thus, $P(\boldsymbol{\theta})$ can be viewed as an estimation of the power of a plane wave coming from the direction $\boldsymbol{\theta}$.

Using this estimation, the microphone array spectral matrix resulted from the sound wave *only* from $\boldsymbol{\theta}$ direction is written as $P(\boldsymbol{\theta})\boldsymbol{v}_\theta \boldsymbol{v}_\theta^H$ (see Appendix 2.7 for more details). If we assume that signals from different directions are uncorrelated, then the signal spectral matrix for sound coming from an FOV

is an integral of the single-direction estimation over the entire FOV:

$$\mathsf{R}_\mathrm{s} \approx \int_{\boldsymbol{\Theta}} P(\boldsymbol{\theta}) \boldsymbol{v}_\theta \boldsymbol{v}_\theta^H \mathrm{d}\boldsymbol{\theta}, \tag{2.11}$$

where $\boldsymbol{\Theta}$ is the solid angle area spanned by the camera's FOV, set by the current camera direction and zoom settings. Similarly, the noise spectral matrix R_n can be estimated using the same integral but over the solid angle area $\mathcal{S}_3 \setminus \boldsymbol{\Theta}$, where $\mathcal{S}_3$ denotes the solid angle of an entire 3D sphere. Note that both R_s and R_n are frequency dependent, and thus they are estimated for each individual frequency band. The estimation (2.11) can be applied to an arbitrary FOV, and is agnostic to the sound source distribution in the FOV.

We note that a similar integral has been used for the standard MVDR beamformer (Gu and Leshem, 2012) to estimate the spectral matrix excluding a single sound direction. However, the accuracy of the power spectrum estimation (2.10) and the matrix estimation (2.11) are unclear. In the rest of this section, we theoretically analyze and justify this estimation.

2.3.3.3 Analysis

Our estimation of the single-direction power $P(\boldsymbol{\theta})$ is built on the MPDR beamformer. Traditionally, a beamformer is used to enhance sound from a direction $\boldsymbol{s}$. It has been shown that the MPDR beamformer is identical to the MVDR beamformer when the steering direction $\boldsymbol{d}$ (in MPDR) is chosen to be the true sound source direction $\boldsymbol{s}$, but MPDR beamformer is much less reliable (Brandstein and Ward, 2013): a small mismatch between $\boldsymbol{d}$ and $\boldsymbol{s}$ can degrade significantly the MPDR performance. Fortunately, this is not an

issue in our case, since we have no explicit notion of sound sources. When evaluating the integral (2.11), we treat each direction $\boldsymbol{d}$ in the FOV as a true sound source direction $\boldsymbol{s}$.

Next, we present an analytical understanding of (2.10) and (2.11) for spectral matrix estimation. First, if the recording environment has only (uncorrelated) ambient noise, then the acoustic power is uniform in space, and the spectral matrix R has the form, $\mathsf{R} = \sigma \mathsf{I}_M$, where σ is the ambient noise power, and I_M is an $M \times M$ identity matrix, where M is the number of microphones. We therefore expect the estimated R_s to have power proportional to the FOV area . In this case, $P(\boldsymbol{\theta})$ is a constant σ/M, and R_s in (2.11) indeed has diagonal elements proportional to the FOV area. Now, consider the general case of estimating the signal power $P(\boldsymbol{\theta})$ for the direction $\boldsymbol{\theta}$. Assuming signals from different directions are uncorrelated, then the (true) spectral matrix can be decomposed into two,

$$\mathsf{R} = \mathsf{R}_\mathsf{c} + m_\theta \boldsymbol{v}_\theta \boldsymbol{v}_\theta^H, \tag{2.12}$$

where R_c accounts for the sound signals from all directions but $\boldsymbol{\theta}$, and the second term is the contribution of a plane wave coming from $\boldsymbol{\theta}$: m_θ is its power, and $\boldsymbol{v}_\theta$ is a vector defined in (2.6) (see Appendix 2.7 for more explanation of the plane wave contribution). To see how well the estimation (2.10) approximates the true power m_θ, we express $P(\boldsymbol{\theta})$ analytically by applying

the matrix inversion lemma (Meyer, 2000) on R and obtain

$$\begin{aligned} P(\boldsymbol{\theta}) &= \left[\boldsymbol{v}_\theta^H \mathsf{R}_\mathrm{c}^{-1} \boldsymbol{v}_\theta - \frac{m_\theta}{1 + m_\theta \boldsymbol{v}_\theta^H \mathsf{R}_\mathrm{c}^{-1} \boldsymbol{v}_\theta} (\boldsymbol{v}_\theta^H \mathsf{R}_\mathrm{c}^{-1} \boldsymbol{v}_\theta)^2 \right]^{-1} \\ &= \left(a - \frac{m_\theta a^2}{1 + m_\theta a} \right)^{-1} \\ &= m_\theta + \frac{1}{a}, \end{aligned} \tag{2.13}$$

where a denotes $\boldsymbol{v}_\theta^H \mathsf{R}_\mathrm{c}^{-1} \boldsymbol{v}_\theta$ for short. It is evident the estimated power $P(\boldsymbol{\theta})$ differs from the true power m_θ by a constant $1/a$. In fact, $1/a$ is the noise power in the output signal from the MVDR beamformer (i.e., using R_c in (2.5) and computing $\boldsymbol{w}_\mathrm{FB}^H \mathsf{R}_\mathrm{c} \boldsymbol{w}_\mathrm{FB}$), and the MVDR beamformer is designed to minimize exactly this noise power ($1/a$). Here noise is all the signals *not* from direction $\boldsymbol{\theta}$. Thus, from (2.13), we conclude that the estimation accuracy of $P(\boldsymbol{\theta})$ depends on $m_\theta a$, the output SNR ratio of the MVDR beamformer.

Furthermore, we show that the estimation of R_s in (2.11) has a bounded error. Formally, we rewrite (2.11) as

$$\mathsf{R}_\mathrm{s} = \int_{\boldsymbol{\Theta}} m_\theta \boldsymbol{v}_\theta \boldsymbol{v}_\theta^H \mathrm{d}\boldsymbol{\theta} + \Delta. \tag{2.14}$$

The first term here is the true signal spectral matrix, and Δ is the error residual introduced by the estimator (2.11). As derived in Appendix 2.8, Δ is bounded from above and below:

$$\frac{\lambda_\mathrm{min}}{M} \int_{\boldsymbol{\Theta}} \boldsymbol{v}_\theta \boldsymbol{v}_\theta^H \mathrm{d}\boldsymbol{\theta} \leq \Delta \leq \frac{\lambda_\mathrm{max}}{M} \int_{\boldsymbol{\Theta}} \boldsymbol{v}_\theta \boldsymbol{v}_\theta^H \mathrm{d}\boldsymbol{\theta}, \tag{2.15}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimal and maximal eigenvalues of R_c, respectively. This derivation indicates that the residual is bounded from above, proportional to the power of the strongest signal direction other than $\boldsymbol{\theta}$ and inversely proportional to the microphone array size.

In light of this, a simple strategy for improving estimation accuracy of R_s is improving the MVDR's output SNR ratio or increasing the number of microphones (i.e., M). In the next section, we provide more guidance on microphone array design for audiovisual zooming through numerical simulations.

2.4 Empirical Studies of Array Designs

We implement our audiovisual zooming system on a microphone array, which has many design parameters. Yet, there is no golden rule to set those parameters; they depend on specific applications (Lai, Nordholm, and Leung, 2017). We conduct a series of simulation experiments to understand the design parameters tailored for our applications, wherein the mobile device such as a smartphone is the form factor that we will restrict the microphone array to fit in. Concretely, we explore the following questions:

- How does beamforming change with frequency?
- How big should the array be?
- What number of microphones should we use?
- How should we sample the spatial directions in (2.11)?

The first three questions are to understand microphone array configuration, while the last is for efficient implementation of our audiovisual zooming algorithm. Because the audiovisual zooming is based on MVDR beamforming, we must understand how the beamforming performance changes with respect to the array's design parameters. Therefore, the studies here are not meant to evaluate our audiovisual zooming method. Rather, we examine MVDR beamforming under different setups to understand design parameter choices. While there have been plenty of empirical studies of microphone array parameters (e.g., see Rabinovich and Alexandrov, 2013; Gannot et al., 2017; *Microphone Array Beamforming* 2013), our primary goal of conducting these studies is to inform our specific algorithm.

In this section, we present the conclusions we learned from the empirical studies. The details of our simulations and their results are in Appendix 2.9 of the supplementary.

The setup we consider is that of a circular array consisting of a number of omnidirectional microphones evenly placed on a circle in the X-Y plane (see Figure 2.7 in Appendix 2.9) centered at the origin. We choose this configuration because it matches the off-the-shelf physical array that we will use. We also place six sound sources throughout the space: four orthogonal sources in the X-Y plane at 0°, 90°, 180°, and 270°. The other two are placed along the positive and negative Z-axes, respectively. The environment is filled with ambient Gaussian noise. The MVDR beamformer aims to enhance the sound coming from the positive Z-direction, while suppressing everything else.

2.4.1 Frequency dependence

MVDR performance is frequency dependent. Our experiments focus on the frequency range of typical human speech (i.e., 300-3420Hz). The results are visualized as MVDR beam patterns in Figure 2.7 of Appendix 2.9 at 300Hz, 1860Hz, and 3420Hz. Towards the steering direction, the beamformer always has unit gain, thanks to the distortionless constraint in (2.5), but its shape varies across frequencies. In general, *beamforming performance increases as frequency increases*. The beam pattern at 3420Hz (Figure 2.7-d) also shows some side lobs near the X-Y plane—a phenomenon known as *spatial aliasing* occurring at high frequencies.

2.4.2 Array size

Next, we study the effect of the overall size of the array: the number of microphones is fixed and the inter-microphone spacing is changed. Figure 2.8 in Appendix 2.9 shows the details of our studies. In general, the simulations show that *better directionality requires a larger array size*—but not too large. For example, once the array size reaches 50cm, we get *non-trivial spatial aliasing effects*. Though there is a strong gain toward the target direction, there are also many unwanted secondary gains in other directions. One way to avoid spatial aliasing is to increase the spatial sampling rate. This brings up the next natural question: how many microphones should we use?

2.4.3 Number of microphones

We simulate the beamformer response as we change the number of microphones while fixing the overall size (i.e., 5cm in radius). The results are shown in Figure 2.9 in Appendix 2.9. When we increase the number of microphones, we obtain better suppression of the interference relative to the target. However (and perhaps somewhat surprisingly), the performance plateaus once we have sufficient microphones. Figure 2.9 shows that 16 microphones become superfluous, yielding no improvement over 8. In other words, *more microphones improve performance, but have diminishing returns*.

For the 8- and 16-microphone cases, there are "indentations" in the directions of the X-Y plane interferers (bottom row of Figure 2.9), indicating so-called *null* responses toward those directions—this is the desired effect. Although there are reasonably larger gains near the areas of those indentations, no sounds comes from those directions in our setup, and so no suppression is needed. This is the advantage of adaptive beamformers (like MVDR): they work to rearrange the gain distribution to best nullify interferers while distributing energy in places where no sound is thought to be.

2.4.4 Sampling density

Our audiovisual zooming algorithm estimates spectral matrices R_n and R_s in (2.11) through Monte Carlo integration. In practice, we need to sample directions within a desired FOV $\boldsymbol{\Theta}$ and at its outside $\mathcal{S}_3 \setminus \boldsymbol{\Theta}$. The sampling density of the directions should not be arbitrary because for each target we beamform towards, there is an effective main lobe with a non-trivial width,

meaning that although the gain in the direction of the target is maximal, there is also non-zero gain from directions nearby the target direction. As shown in Figure 2.10 in Appendix 2.9, the gain falls off as the sound incoming direction deviates from the target direction. We determine an acceptable reduction in dB (i.e., 1.8dB) for nearby sounds and Figure 2.10 suggests sampling directions with an angular separation of 20°.

2.4.5 Discussions: extending to 3D arrays

Thus far, most microphone arrays have a 2D planar configuration. We note that there is inherently a symmetry. For the sound wave coming from an elevation angle θ in spherical coordinates, no 2D planar array can disambiguate it from the wave coming from the same but negated elevation angle $-\theta$ (and the same azimuthal angle). For our applications, this is not a significant problem, as the sound waves from behind the array are often blocked by the user who is holding the camera to capture or the table on which the array is placed (see Figure 2.4). Nevertheless, here we study the performance of a 3D array for future extension. We add an additional microphone at the center of the array and gradually move it along the negative Z-axis to break the planar symmetry. We then examine how this affects the beam pattern. As shown in Figure 2.11, this additional microphone indeed helps to break the symmetry. As it moves further away from the microphone plane, the interferer behind the array attenuates more. However, such a 3D array is much more bulky than the 2D array. Today, the form factor of a mobile device is one of the most decisive factors for its use on a daily basis. It is unclear if a 3D array is worth

equipping on mobile devices.

2.5 Experiments and Results

We demonstrate our results via experiments on both synthetic and real data: first, we use synthetic mixtures of clean speech sources in various configurations to evaluate audiovisual zooming enhancement (§2.5.1) using the following quantitative metrics:

1. **Signal-to-Distortion-Ratio (SDR)** (Vincent, Gribonval, and Fevotte, 2006): SDR evaluation takes as input the enhanced signal and the reference signal it should ideally match. It first decomposes the enhanced signal into four components: a target component coming from the reference signal, an interferer component containing other unwanted sources' contributions, a noise component encapsulating sensor noises and an artifact component capturing distortions from other sources (like forbidden distortions of the sources and/or "burbling" artifacts). SDR is then calculated as the logarithmic ratio (in dB) of the energy in the target component to the energy in the unwanted components.

2. **Signal-to-Noise-Ratio (SNR)**: SNR is defined here to be the logarithmic ratio (in dB) of the energy of the enhanced signal to that of the noise signal, the latter of which is defined at each time point as the difference between the enhanced signal and the reference signal.

3. **Waveform Amplitude Distribution Analysis SNR (WADA-SNR)** (Kim and Stern, 2008): This metric evaluation assumes that clean speech has

an amplitude distribution well approximated by the Gamma distribution with a shaping parameter of 0.4, and that the additive noise signal is Gaussian. It is calculated by studying the amplitude distribution of the enhanced signal. As the reference signal we are attempting to recover in our experiments is oftentimes speech, we use this metric to measure enhancement quality that better correlates with our task.

4. **Short-Time Objective Intelligibility (STOI)** (Taal et al., 2010): Popular objective measures such as SDR and SNR above often do not reflect well the speech intelligibility–how easily the resulting signals can be understood by humans. The STOI score is designed to bridge that divide.

5. **Perceptual Evaluation of Speech Quality (PESQ)** (Rix et al., 2001): Similar to STOI, common measures like SDR and SNR do not correlate well with voice quality evaluation results from humans. PESQ was developed to model these subjective tests better, and is a widely used industry standard for objective voice quality testing.

Next, we perform real experiments using audio loud speakers in various configurations to compute SDR, SNR, WADA-SNR, STOI and PESQ enhancements. Finally, we show qualitative performance using two different hardware platforms to demonstrate feasibility in mobile settings. For all experiments, we tested our method against the MVDR beamforming approach as a basis-of-comparison.

2.5.1 Synthetic Mixture of Speech

As there are no public datasets for audiovisual zooming, we generated data by mixing clean speech tracks in different (virtual) geometric configurations. We performed randomized trials to span the space around a given microphone array, varying the number of speakers and the solid angle over which we wished to enhance the sound. For each target solid angle, different numbers of speech signals were randomly placed within to differentiate from traditional beamforming scenarios where only one sound source is targeted.

We use the same 6-microphone hexagonal array configuration as in §2.4 and simulate sound source directions by delaying the audio signal at each microphone appropriately. In all cases, clean speech signals are obtained from real recordings. For each trial, we do as follows: a) starting from a selection of 10 clean speech sound tracks, we randomly choose between 2-10 overall speakers, of which 1-4 are randomly chosen to be targets and the rest are interferers; b) we randomly select a solid angle between 10° and 120° in both azimuth and elevation as well as a randomly-chosen direction-to-focus; c) given these setups, we randomly place the targets within the solid angle target-FOV as well as randomly place the interferers elsewhere. We run 500 random trials, applying both our method as well as MVDR (directed towards the center of the target FOV), and compute averaged metrics. The results are shown in Table 2.1.

Qualitatively, because MVDR (and other beamforming methods) only enhances sound from a single direction, when a conic angle of space contains

	MVDR	Our Method
SDR [dB]	-2.96793	**-0.0095**
SNR [dB]	-0.86467	**1.80908**
WADA-SNR [dB]	5.1604	**7.55923**
STOI	0.66414	**0.71992**
PESQ	1.75125	**2.04402**

Table 2.1: Comparison of our method against MVDR. Our method consistently outperforms MVDR.

multiple sounds, a relatively *muffled* sound enhancement results when pointing towards the FOV center. In contrast, our method integrates all sounds coming from within the desired FOV and attempts to enhance all equally, resulting in a more *crisp* sound enhancement.

2.5.2 Audio Speaker Experiments

We perform real experiments using four loudspeakers, playing individual sound tracks through each in various geometric configurations. The speakers are placed about a circular round-table, at 90°, 45°, 30°, and 15° (see Figure 2.4).

In each scenario, a single sound track is produced from each speaker and all speakers play simultaneously from different directions. We then cycle through the speakers and select either two or three adjacent speakers as the *targets*, while all others serve as the *interferers*. Again, we focus on more than one target sound at a time so as to differentiate from traditional beamforming scenarios. In some experiments, all speakers play clean speech tracks whereas in others, one of the speakers plays either a soft music track (e.g., jazz) or a pre-recorded "crowd noise" (e.g., recording from a crowded restaurant). Never is the music or crowd chosen as the target: these serve only to provide

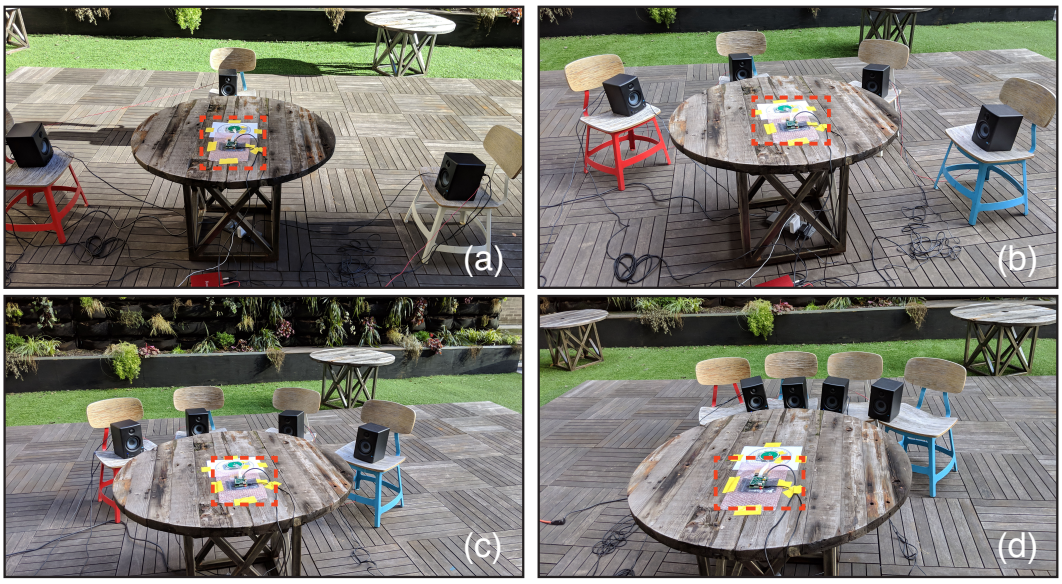


Figure 2.4: Loudspeaker experiments. Four audio loudspeakers play individual sounds in configurations of 90° (a), 45° (b), 30° (c), and 15° (d) angular separations. The microphone array is placed on the table (indicated by the orange boxes). We then select anywhere between 2-3 speakers to simultaneously enhance while attenuating all other speaker sounds (see supplementary video to the paper).

interference signals.

In each experiment, once the target and interferers are chosen, we play the sounds twice: a) first, all are played together to mimic a "noisy" environment; b) second, we play only the target sounds alone to serve as the "ground truth" against which we can compute SNR/SDR metrics. We present our results in each of the angular cases separately in Table 2.2. By all the metrics, our method out-performs MVDR.

2.5.3 Use Case Demonstration

Finally, we demonstrate Audiovisual Zooming on two mobile platforms: a smartphone and a 360° camera, both attached to a 6-microphone hexagonal

Metric	Method	90°	45°	30°	15°
SDR [dB]	MVDR	-4.08	-1.10	-3.54	-1.98
	Ours	**-2.11**	**0.08**	**-2.56**	**-1.29**
SNR [dB]	MVDR	-0.55	0.78	-0.52	-0.19
	Ours	**0.75**	**1.86**	**0.48**	**0.79**
WADA-SNR [dB]	MVDR	-0.24	1.73	1.13	1.60
	Ours	**1.30**	**3.39**	**3.42**	**4.26**
STOI	MVDR	0.46	0.58	0.50	0.53
	Ours	**0.59**	**0.63**	**0.54**	**0.56**
PESQ	MVDR	1.72	1.95	1.86	1.71
	Ours	**2.00**	**2.17**	**2.07**	**1.80**

Table 2.2: Comparison of our method against MVDR for real loudspeaker experiments shown in Figure 2.4.

array (see Figure 2.2).

We refer to our supplementary video for the audiovisual zooming demonstration using both platforms. Using the smartphone, we demonstrate a scenario in which a user captures two persons speaking simultaneously. When both persons are captured in the camera's FOV (see Figure 2.1-a), their voices are mixed together. As the user zooms in the camera's FOV to focus on one person (see Figure 2.1-b), her voice stands out while the other's voice is suppressed. Next, the user shifts the camera's FOV to another person (see Figure 2.1-c), and consequently his voice gradually becomes clear while the other fades out. We note that in this process the change of audio signal is fully synchronized with the change of the camera's FOV, thanks to our audiovisual zooming technique—for example, the sound gradually changes from one person's voice to another voice as the camera pans.

To demonstrate the 360° camera, four people sit around a round table and simultaneously converse. The 360° camera with the microphone array

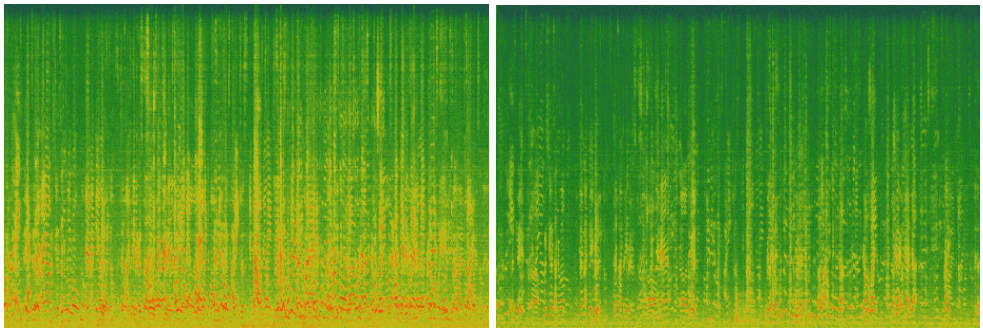


Figure 2.5: Speaker separation on a Ricoh camera. Using a mobile microphone array attached to a 360° camera, we perform audiovisual zooming on 4 people seated around a table at roughly 90° angular offsets from one another. In this scenario, 2 pairs of people are having simultaneous conversations and we use our method to focus in on one conversation. The left shows the raw noisy spectrogram as recorded in one of the microphones in the array. On the right, we show the spectrogram after sound enhancement using our method, which is noticeably *cleaner* (see supplementary video).

is placed on the table and pointed upwards. It is difficult to distinguish the individual speakers in the raw audio. Since the 360° camera captures all speakers, the user can set the camera's FOV on individual speakers to boost their voice relative to the others'. As the camera's FOV switches from one speaker to another, the boosted voice switches correspondingly. Consequently, the user can choose to see and hear individual speakers (see supplementary video).

In these scenarios, it was not possible to obtain ground truth (e.g., target-only sounds that perfectly match the raw, noisy signals), and so here we show our results qualitatively via spectrograms before and after enhancement (see Figure 2.5).

2.6 Conclusion

In this work, we extend the concept of the camera's FOV to enhance audio recording. Traditionally, camera's FOV defines only the visual frustum through which the visual content is captured by the camera. A fundamental limitation is the inconsistency between captured visual and auditory content—the sound is captured regardless of the FOV setup. To address this limitation, we have introduced an audiovisual zooming technique by leveraging the microphone array and augmenting classic beamforming methods. We have presented a method that estimates the sound spectral matrices which accounts for the desired sound signals within the FOV and those outside of the FOV. The estimated spectral matrices allow us to enhance sound coming within the FOV by solving a generalized eigenvalue problem. Our method requires no analysis of captured video frames. It can enhance however many sound sources within the FOV, and the captured imagery is in tandem with the resulting sound signal.

A limitation of our approach is that in a reflective environment, a sound source outside of the FOV may emit sound waves that arrive to the microphone from within the FOV through reflections. In this case, our audiovisual zooming method will still enhance those received sound signals. In the future, we plan to investigate this limitation by estimating the room acoustics, which might require the analysis of captured video frames to understand the environment geometry and acoustic properties (e.g., Li, Langlois, and Zheng, 2018).

2.7 Appendix: Spectral Matrix of Sound from a Direction θ

Consider a plane wave impinging on a microphone array (see Figure 2.6). Let $\boldsymbol{p}_i$ denote the position of individual microphones, and the plane wave comes from the direction $\boldsymbol{d}$ with an intensity A. The angle between the sound incoming direction and the microphone array's facing direction is θ. Then, the sound waves received at individual microphone is expressed as

$$s_i = A^{\frac{1}{2}} e^{-j\left(\frac{\omega}{c}\boldsymbol{d}^T \boldsymbol{p}_i + \omega t\right)}.$$

Here $\frac{\omega}{c}\boldsymbol{d}^T\boldsymbol{p}_i$ is the (relative) phase at the microphone i. Putting all s_i into a vector $\boldsymbol{s} = [s_1 \ \ldots \ s_M]^T$, we can compute the spectral matrix by its definition (Oppenheim, 1999) as

$$\mathsf{R}(\omega) = \mathrm{FFT}\{\boldsymbol{s}\boldsymbol{s}^T\} = A\boldsymbol{v}_{\boldsymbol{d}}\boldsymbol{v}_{\boldsymbol{d}}^H, \tag{2.16}$$

where $\boldsymbol{v}_{\boldsymbol{d}}$ is the steering vector defined in (2.6). This expression (2.16) is what we used in (2.11).

2.8 Appendix: Derivation of Error Bound (2.15)

First, we substitute (2.13) into the R_s estimation (2.11) and obtain the expression of Δ in (2.14),

$$\Delta = \int_\Theta \frac{1}{a}\boldsymbol{v}_\theta\boldsymbol{v}_\theta^H = \int_\Theta \frac{1}{\boldsymbol{v}_\theta^H\mathsf{R}_\mathrm{c}^{-1}\boldsymbol{v}_\theta}\boldsymbol{v}_\theta\boldsymbol{v}_\theta^H. \tag{2.17}$$

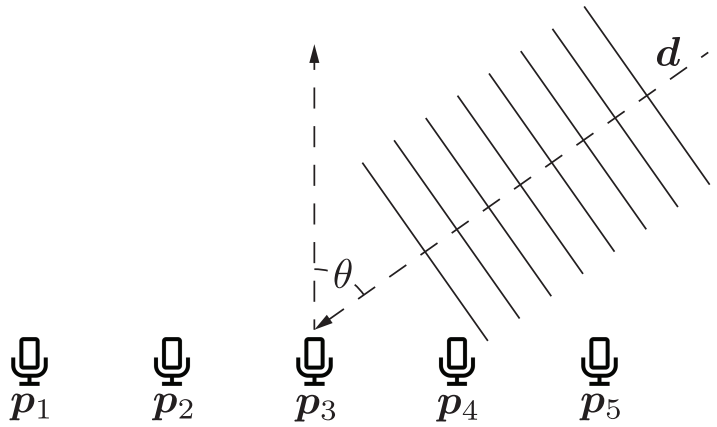


Figure 2.6: Consider a microphone array in which each microphone is located at position $\boldsymbol{p}_i$. A single sound comes from the direction $\boldsymbol{d}$ as a plane wave. The angle between the microphone array's facing direction and the sound incoming direction is θ.

Here $\boldsymbol{v}_\theta^H \mathsf{R}_\mathsf{c}^{-1} \boldsymbol{v}_\theta$ is bounded by the maximum and minimum eigenvalue of $\mathsf{R}_\mathsf{c}^{-1}$. Also, notice that $\boldsymbol{v}_\theta$ is the steering vector, which has a specific form (2.6). Therefore, we have

$$\lambda_{\min}^{\hat{c}} \boldsymbol{v}_\theta^H \boldsymbol{v}_\theta^H = \lambda_{\min}^{\hat{c}} M \leq \boldsymbol{v}_\theta^H \mathsf{R}_\mathsf{c}^{-1} \boldsymbol{v}_\theta \leq \lambda_{\max}^{\hat{c}} \boldsymbol{v}_\theta^H \boldsymbol{v}_\theta^H = \lambda_{\max}^{\hat{c}} M, \tag{2.18}$$

where M is the number of microphones; $\lambda_{\max}^{\hat{c}}$ and $\lambda_{\min}^{\hat{c}}$ are the maximum and minimum eigenvalues of $\mathsf{R}_\mathsf{c}^{-1}$, respectively. They are related to the eigenvalues of R_c through

$$\lambda_{\max}^{\hat{c}} = \frac{1}{\lambda_{\min}} \text{ and } \lambda_{\min}^{\hat{c}} = \frac{1}{\lambda_{\max}}.$$

Combing this expression with (2.17) and (2.18), we obtain the error bound of Δ as shown in (2.15).

2.9 Appendix: Details of Empirical Studies

In our empirical studies, the MVDR beamformer aims to enhance the sound coming from the positive Z-direction, while suppressing everything else. Since we know precisely what the unwanted signals are in our simulation, we can directly compute the noise spectral matrix R_n, which is in turn used in (2.7) to evaluate $\boldsymbol{w}_\mathrm{BF}$. We visualize MVDR performance by evaluating its *beam pattern* across a range of frequencies. The beam pattern describes the effective gain for signals coming from individual directions $\boldsymbol{\theta}$ when the beamformer is set to enhance toward a direction $\boldsymbol{\theta}_0$. It is defined as

$$g(\boldsymbol{\theta}; \boldsymbol{\theta}_0) = |\boldsymbol{w}_{\mathrm{BF},\boldsymbol{\theta}_0}^H \boldsymbol{v}_\theta|, \tag{2.19}$$

where $\boldsymbol{w}_{\mathrm{BF},\boldsymbol{\theta}_0}^H$ is the MVDR weights from (2.7) when the steering direction is set to be $\boldsymbol{\theta}_0$, and $\boldsymbol{v}_\theta$ is defined in (2.6).

2.9.1 Frequency dependence

To study the frequency dependence of the microphone array's performance, we use sound sources that produce sound signals at a fixed frequency, and the frequency is varied to ascertain the beamforming performance with respect to frequency change. This is seen in Figure 2.7 as tighter main lobes in the target's direction for higher frequencies, meaning the interfering sounds are better suppressed.

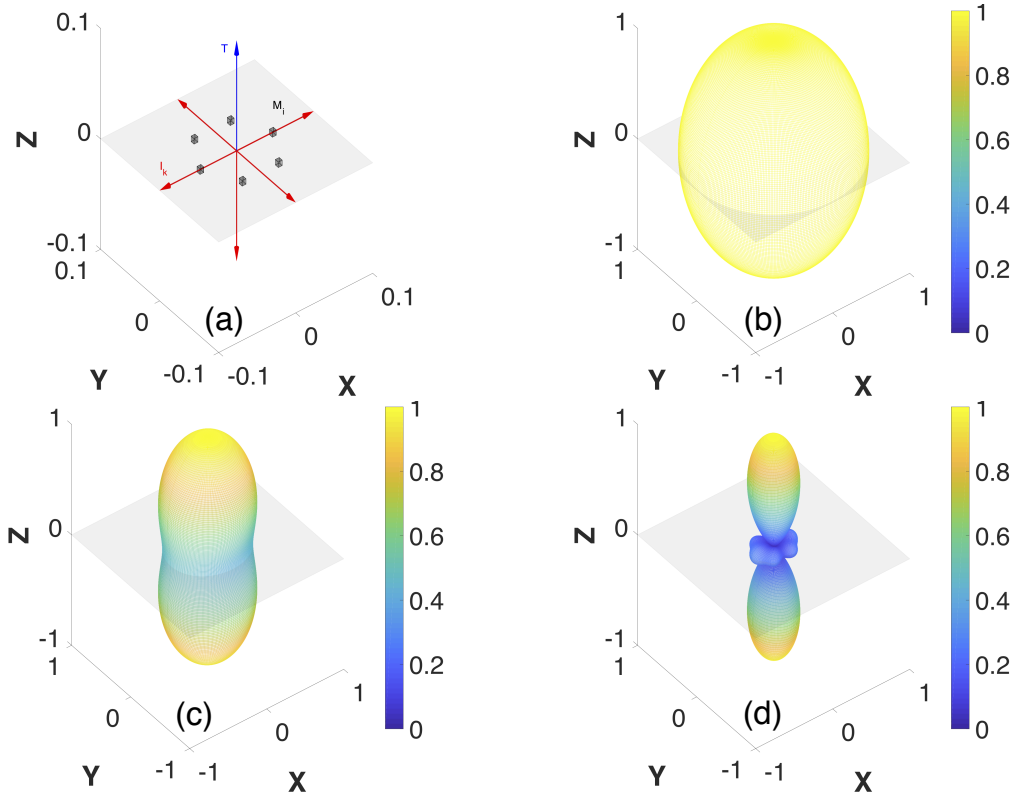


Figure 2.7: Frequency dependence. We visualize the frequency dependence of the MVDR beam patterns. (a) The array consists of 6 microphones shown as gray cubes on the X-Y plane, where the microphones are spaced evenly 5cm from one another. The 6 sound sources are spread throughout the space: 4 interfering sources are shown in red on the X-Y plane along with another interfering source in the negative z-axis. The target is shown in blue in the positive z-axis. (b-d) The MVDR beam patterns at three different frequencies, 300Hz (b), 1860Hz (c), and 3420Hz (d), are shown both in the shape of the surface and as the color (yellow as 1 and blue as 0).

2.9.2 Array size

Figure 2.8 shows the change in *average beam pattern* across the human speech frequency range (300-3420Hz) as the microphone spacing is changed from 0.5cm to 5cm and then to 50cm. The more microphones we have, the better we can sample (spatially) with larger array sizes. The smallest spacing setting of 0.5cm gives almost no directionality, with an omni-directional gain response.

As we increase to 5cm, the directionality improves with better suppression of the interference sources relative to the target.

2.9.3 Number of microphones

The simulation setup and results are shown in Figure 2.9 and its caption.

2.9.4 Sampling density

Figure 2.10 (bottom) shows a plot of average gain (y-axis) within the human frequency range as a function of elevation (x-axis) angle (e.g., offset from the target direction). Note that we ignore the variation of azimuth angle because it has no effect for the given array configuration and target direction, as shown in Figure 2.7. Therefore, we consider the microphone array scenario as shown on the top (circular 6-sensor array with 5cm spacing). The simulation shows that the gain falls off from the target direction for any nearby sounds within a small FOV of the target. We use this to determine a reasonable sampling rate for our sphere integration approach, as discussed in the main text (in §2.4). Here, we convert the gain as expressed in (2.19) to dB as: $g_{dB}(\theta) = 20 * log_{10}(g(\theta))$.

2.9.5 Extension to 3D arrays

We explore the effect of a 3D microphone array as a future extension. 3D array is able to break the symmetry that a 2D array suffers from, although it is much more bulky and might not be compatible with the small form factor of most mobile devices. The result and simulation details are shown in Figure 2.11 and its caption.

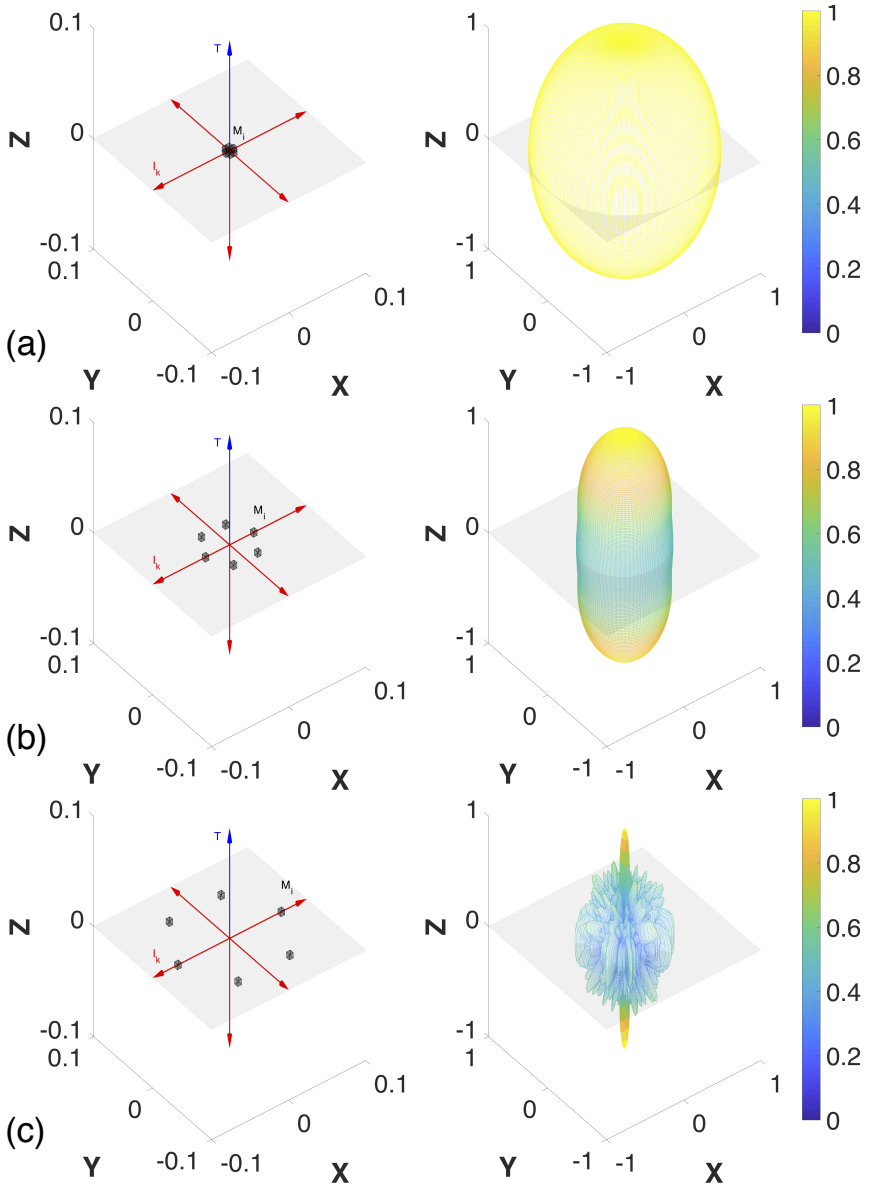


Figure 2.8: Role of array size. We use a 6-element circular array but vary the inter-microphone spacing to adjust the overall array size. On the left of each subplot is the spatial configuration: the target, interferers, and ambient noise are the same as before (Figure 2.7), and the spacing of the microphones (in the x-y plane) changes from 0.5cm (a) to 5cm (b) to 50cm (c). On the right of each subplot is the average beam pattern across the frequency range of human speech, to indicate the average performance of the beamformer in that range.

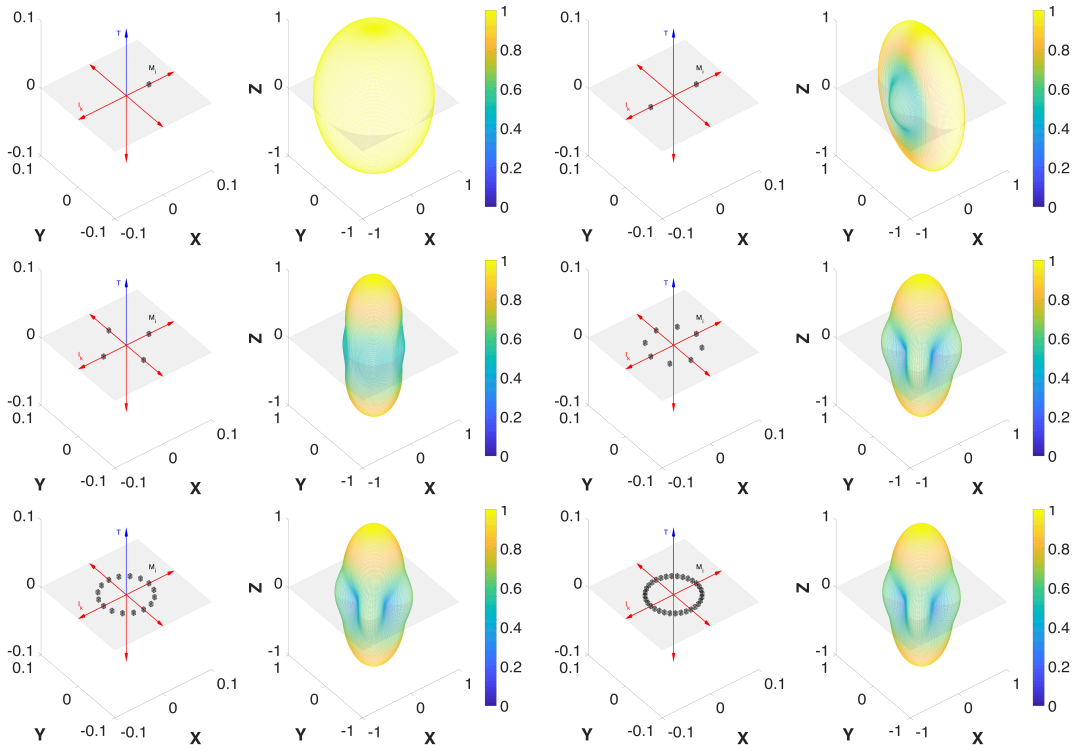


Figure 2.9: Number of microphones. Here the microphone array geometry is a circle with a fixed 5cm radius in the X-Y plane. We examine how changing the number of microphones on this circle affects the average beam pattern of the beamformer. The definition of beam pattern is presented in Appendix 2.9. [Top-Left] A single microphone yields an omnidirectional response. [Top-Right] Two microphones improves directionality by suppressing two side interferers, but not the others. [Middle-Left] Four microphones improves directionality further. [Middle-Right] Eight microphones are better, and the performance plateaus as 16 [Bottom-Left] or 32 [Bottom-Right] microphones yield no clear improvement.

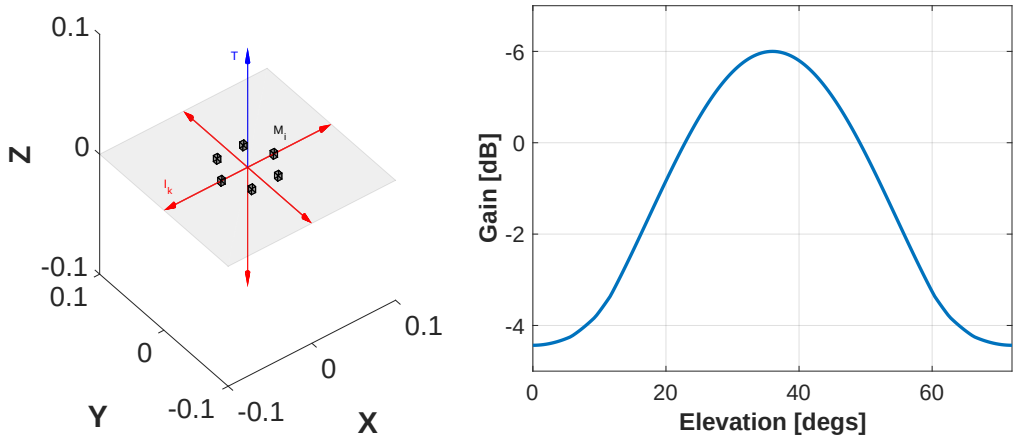


Figure 2.10: Target proximity sensitivity. When beamforming at a target in the presence of interferers, it is important to know how the gain falls off from the direction of the target for nearby interferers. We show this for a given 6-microphone array configuration with 5cm spacing [left] as a 3D surface plot of average gain (across the human frequency range) as a function of azimuth and elevation angular offset from the desired target direction [right]. This allows us to better understand how close sounds can be before they are not sufficiently separable.

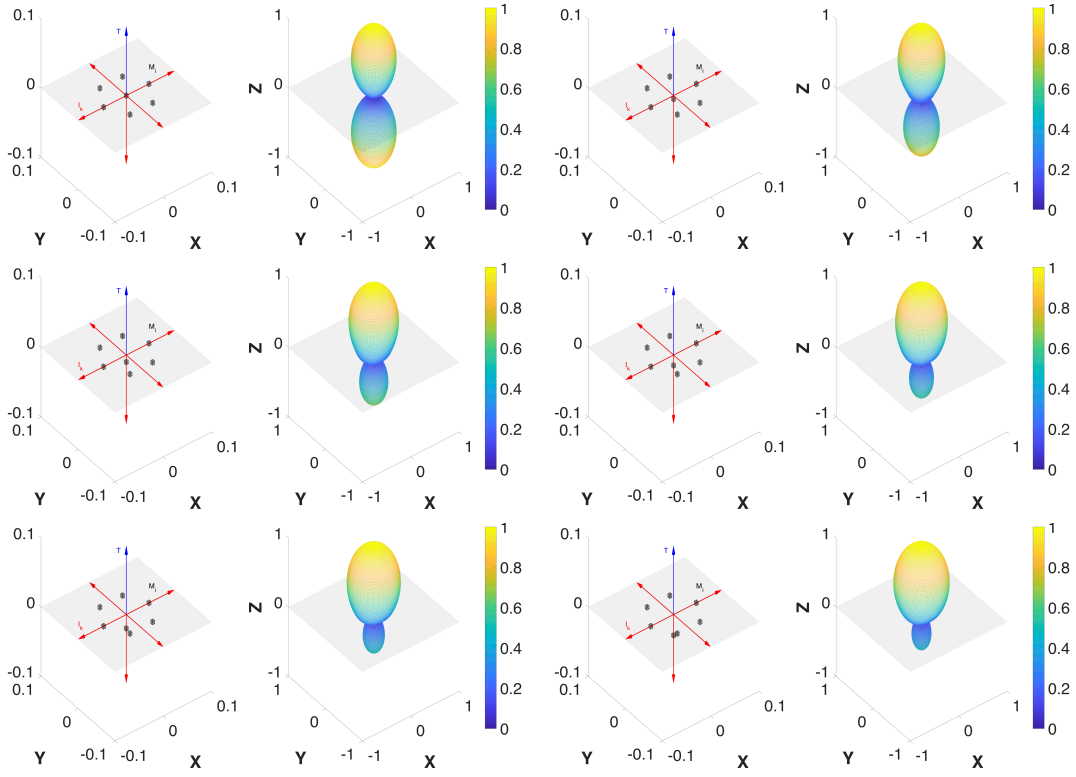


Figure 2.11: 3D asymmetries. Effect of MVDR beamforming as a microphone as added to the third dimension. We consider our baseline 6-microphone circular planar array and we add an extra microphone at the center. We then move the extra mic along the negative z-dimension to break the 2D symmetry and observe how this affects the gain for the previously-ambiguous interference behind the array. [Top-Left] The extra mic is at z=0, and so the symmetry remains. [Top-Right] The extra mic moves down the negative z-axis by 5mm, and the gain in the direction of the interferer subsides. As the microphone moves further along the axis by 10mm [Middle-Left], 15mm [Middle-Right], 20mm [Bottom-Left], and 30mm [Bottom-Right], the gain in the direction of the interferer attenuates more and more while the gain in the direction of the target remains maximal.

References

Nair, Arun Asokan, Austin Reiter, Changxi Zheng, and Shree Nayar (2019). "Audiovisual zooming: what you see is what you hear". In: *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 1107–1118.

Gannot, S., E. Vincent, S. Markovich-Golan, and A. Ozerov (2017). "A Consolidated Perspective on Multimicrophone Speech Enhancement and Source Separation". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25.4, pp. 692–730.

Ephrat, A., I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W.T. Freeman, and M. Rubinstein (2018). "Looking to Listen at the Cocktail Party: A Speaker-Independent Audio-Visual Model for Speech Separation". In: *ACM Transactions on Graphics (SIGGRAPH)*.

Zhao, Hang, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba (2018). "The Sound of Pixels". In: *The European Conference on Computer Vision (ECCV)*.

Owens, Andrew and Alexei A. Efros (2018). "Audio-Visual Scene Analysis with Self-Supervised Multisensory Features". In: *ECCV*. Springer, pp. 639–658.

Billingsley, John and R Kinns (1976). "The acoustic telescope". In: *Journal of Sound and Vibration* 48.4, pp. 485–510.

Michel, Ulf et al. (2006). "History of acoustic beamforming". In: *Berlin Beamforming Conference, Berlin, Germany, Nov*, pp. 21–22.

Veen, B.D. Van and K.M. Buckley (1988). "Beamforming: A versatile approach to spatial filtering". In: *IEEE Acoust., Speech, Signal Process. Mag.* 5.2, pp. 4–24.

Teutsch, Heinz (2007). *Modal array signal processing: principles and applications of acoustic wavefield decomposition*. Vol. 348. Springer.

Capon, Jack (1969). "High-resolution frequency-wavenumber spectrum analysis". In: *Proceedings of the IEEE* 57.8, pp. 1408–1418.

Stoica, Petre, Randolph L Moses, et al. (2005). "Spectral analysis of signals". In:

Trees, H.L. Van (2002). *Detection, Estimation, and Modulation Theory, Part IV: Optimum Array Processing*. New York: Wiley.

Li, Jian, Petre Stoica, and Zhisong Wang (2003). "On robust Capon beamforming and diagonal loading". In: *IEEE transactions on signal processing* 51.7, pp. 1702–1715.

Griffiths, L.J. and C.W. Jim (1982). "An alternative approach to linearly constrained adaptive beamforming". In: *IEEE Trans. Antennas Propag.* 30.1, pp. 27–34.

Hung, Eric KL and Ross M Turner (1983). "A fast beamforming algorithm for large arrays". In: *IEEE Transactions on Aerospace and Electronic Systems* 4, pp. 598–607.

Yu, Jung-Lang and Chien-Chung Yeh (1995). "Generalized eigenspace-based beamformers". In: *IEEE Transactions on Signal Processing* 43.11, pp. 2453–2461.

Warsitz, E. and R. Haeb-Umbach (2007). "Blind Acoustic Beamforming Based on Generalized Eigenvalue Decomposition". In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.5, pp. 1529–1539.

Thiergart, O., K. Kowalczyk, and E.A.P. Habets (2014). "An Acoustical Zoom Based on Informed Spatial Filtering". In: *International Workshop on Acoustic Signal Enhancement (IWAENC)*.

Ruochen, W., Z. Yuhong, and Z. Wei (2014). "Acoustic Zooming Based on Real-Time Metadata Control". In: *Proceedings of IC-NIDC*.

Duong, N.Q.K., P. Berthet, S. Zabre, M. Kerdranvat, A. Ozerov, and L. Chevallier (2017). "Audio Zoom for Smartphones Based On Multiple Adaptive Beamformers". In: *International Conference on Latent Variable Analysis and Signal Separation*.

Mendat, Daniel R, James E West, Sudarshan Ramenahalli, Ernst Niebur, and Andreas G Andreou (2017). "Audio-Visual beamforming with the Eigenmike microphone array an omni-camera and cognitive auditory features". In: *2017 51st Annual Conference on Information Sciences and Systems (CISS)*. IEEE, pp. 1–4.

VisiSonics 5/64 Audio Visual Camera. http://www.thasar.com/site/wp-content/uploads/2018/04/VisiSonics_AudioCamera.Pamphlet.pdf.

Li, Zhiyun and Ramani Duraiswami (2007). "Flexible and optimal design of spherical microphone arrays for beamforming". In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.2, pp. 702–714.

Feng, Weijiang, Naiyang Guan, Yuan Li, Xiang Zhang, and Zhigang Luo (2017). "Audio visual speech recognition with multimodal recurrent neural networks". In: *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, pp. 681–688.

Mroueh, Youssef, Etienne Marcheret, and Vaibhava Goel (2015). "Deep multimodal learning for audio-visual speech recognition". In: *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, pp. 2130–2134.

Rivet, Bertrand, Wenwu Wang, Syed Mohsen Naqvi, and Jonathon Chambers (2014). "Audiovisual speech source separation: An overview of key methodologies". In: *IEEE Signal Processing Magazine* 31.3, pp. 125–134.

Hershey, John R and Michael Casey (2002). "Audio-visual sound separation via hidden Markov models". In: *Advances in Neural Information Processing Systems*, pp. 1173–1180.

Afouras, Triantafyllos, Joon Son Chung, and Andrew Zisserman (2018). "The Conversation: Deep Audio-Visual Speech Enhancement". In: *arXiv preprint arXiv:1804.04121*.

Brandstein, Michael and Darren Ward (2013). *Microphone arrays: signal processing techniques and applications*. Springer Science & Business Media.

Heymann, J., L. Drude, and R. Haeb-Umbach (2016). "Neural Network Based Spectral Mask Estimation For Acoustic Beamforming". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Gu, Y. and A. Leshem (2012). "Robust Adaptive Beamforming Based on Interference Covariance Matrix Reconstruction and Steering Vector Estimation". In: *IEEE Transactions on Signal Processing* 60.7, pp. 3881–3885.

Meyer, Carl D (2000). *Matrix analysis and applied linear algebra*. Vol. 71. Siam.

Lai, Chiong Ching, Sven Erik Nordholm, and Yee Hong Leung (2017). *A Study Into the Design of Steerable Microphone Arrays*. Springer.

Rabinovich, V. and N. Alexandrov (2013). "Typical Array Geometries and Basic Beam Steering Methods". In: *Antenna Arrays and Automotive Applications*. New York: Springer Science+Business Media. Chap. 2, pp. 23–54.

Microphone Array Beamforming (2013). Tech. rep. InvenseSense, Inc.

Vincent, E., R. Gribonval, and C. Fevotte (2006). "Performance Measurement in Blind Audio Source Separation". In: *Transactions on Audio, Speech and Language Processing* 14.4, pp. 1462–1469.

Kim, Chanwoo and Richard M Stern (2008). "Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis". In: *Ninth Annual Conference of the International Speech Communication Association*.

Taal, Cees H, Richard C Hendriks, Richard Heusdens, and Jesper Jensen (2010). "A short-time objective intelligibility measure for time-frequency weighted noisy speech". In: *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, pp. 4214–4217.

Rix, Antony W, John G Beerends, Michael P Hollier, and Andries P Hekstra (2001). "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs". In: *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*. Vol. 2. IEEE, pp. 749–752.

Li, Dingzeyu, Timothy R. Langlois, and Changxi Zheng (2018). "Scene-Aware Audio for 360° Videos". In: *ACM Trans. Graph.* 37.4.

Oppenheim, Alan V (1999). *Discrete-time signal processing*. Pearson Education India.

Chapter 3

Single Channel Speech Enhancement Using Deep Learning

In this chapter, we demonstrate how deep learning can be used to improve the perceptual qualities of degraded speech to make it more pleasing to human listeners. Speech enhancement aims to improve speech quality by eliminating noise and distortions. While most speech enhancement methods address signal independent additive sources of noise, several degradations to speech signals are signal dependent and non-additive, like speech clipping, codec distortions, and gaps in speech. Here, we first systematically study and achieve state of the art results on each of these three distortions individually. Next, we demonstrate a neural network pipeline that cascades a time domain convolutional neural network with a time-frequency domain convolutional neural network to address all three distortions jointly. We observe that such a cascade achieves good performance while also keeping the action of each neural network component interpretable.

While the previous chapter dealt with multi-channel (i.e., multi microphone) speech, here, we concern ourselves only with single channel speech

which might be either the signal recorded my a single microphone or the output from a prior beamforming step. In addition, the angle from which we approach the problem hinges on a topic – the phase of the speech – often traditionally ignored (Wang and Lim, 1982) in single channel speech enhancement but one which forms the foundation upon which all of beamforming is built. Through our investigation, we demonstrate scenarios in which speech phase, crucial in multi-channel speech enhancement, is also essential to successful single channel speech enhancement, an observation confirmed by recent state of the art results in speech enhancement (Hu et al., 2020; Isik et al., 2020) . Thus, even though beamforming is not the direct focus here, it is our hope that a better understanding of the role of speech phase in audio quality and developing phase-aware enhancement pipelines will aid us in designing better machine learning driven audio beamforming pipelines. The work presented in this chapter has been accepted for publication in Nair and Koishida, 2021.

3.1 Introduction

Speech in the real world is almost always contaminated by noise. The primary goal of speech enhancement (Loizou, 2013) is to improve the intelligibility and perceptual quality of speech by reducing (ideally, eliminating) the presence of unwanted noise signals. Successful speech enhancement helps us humans understand and communicate not only with each other but also with machines (e.g, automatic speech recognition (Lee, 1988))

Traditionally, speech enhancement was achieved through signal processing methods (Wiener, 1950; Boll, 1979; Ephraim and Malah, 1984). Lately, deep

neural network (DNN) approaches to speech enhancement have demonstrated superior performance (Kumar and Florencio, 2016). Solutions based on a wide variety of DNN architectures – fully connected Networks (FCNs) (Xu et al., 2013; Xu et al., 2014), denoising autoencoders (DAEs) (Lu et al., 2013), convolutional neural networks (CNNs) (Park and Lee, 2017), recurrent neural networks (RNNs) (Weninger et al., 2015), and generative adversarial networks (GANs) (Pascual, Bonafonte, and Serra, 2017) – have all been proposed for the speech enhancement task.

DNN methods mostly fall into two families depending on the input data – either the input is in the time domain (Pascual, Bonafonte, and Serra, 2017; Stoller, Ewert, and Dixon, 2018; Luo and Mesgarani, 2019) or the time-frequency domain (Bulut and Koishida, 2020; Park and Lee, 2017; Williamson and Wang, 2017). Time-frequency networks largely operate on magnitude spectrograms and combine the noisy phase of the input with the enhanced output magnitude spectrogram to reconstruct speech (Bulut and Koishida, 2020). In contrast, time domain networks can operate on and enhance the phase information as well (Paliwal, Wójcicki, and Shannon, 2011) as the time domain input contains both magnitude and phase information.

The majority of existing speech enhancement methods largely address scenarios of signal independent additive noise removal – such as the removal of an air conditioner's hum from a video call. However, several degradations, especially in telecommunication applications, are signal dependent. Clipping, codec distortions, and gaps are three common speech degradations in telecommunications. Speech clipping is a non-linear signal distortion that

occurs when the speech signal exceeds the recording microphone's dynamic range. The vast majority of declipping methods are based on signal processing (Záviška et al., 2020) with deep learning only very recently being used (Kashani et al., 2019; Mack and Habets, 2019). Codec distortions occur when speech is encoded by a lossy codec for transmission. Due to limited bandwidth, a low bitrate codec might be used which results in the encoded speech being of poor quality. Various deep learning architectures have been studied for improving the quality of the encoded speech (Biswas and Jia, 2020; Zhao, Liu, and Fingscheidt, 2018; Deng et al., 2020). Gaps in speech arise as a result of poor network conditions. Due to packet loss or jitter in the network, some speech packets are missed and the corresponding speech is not reconstructed. To tackle this, gap filling methods have been developed (Mohamed, Nessiem, and Schuller, 2020).

While each of clipping, codec distortions, and gaps is challenging to address in its own right, in practice these distortions occur together. Our contribution is jointly addressing the three distortions. First, we start by studying each distortion individually, achieving state of the art results. Next, guided by the observation that clipping and gaps are better addressed in time and codec distortions in time-frequency, we propose a novel convolutional neural network (CNN) pipeline based on the UNet (Ronneberger, Fischer, and Brox, 2015) architecture that cascades a time domain UNet (T-UNet) with a time-frequency UNet (TF-UNet) to achieve the best results on the task. A benefit of our pipeline is the function of each component network remains interpretable while still achieving good performance.

The work is organized as follows: Section 3.2 presents clipping, codec distortions, and gaps in speech in more detail and also elaborates on the network architectures used. Section 3.3 describes the dataset and experiments conducted. Section 3.4 presents the findings from our experiments . Section 3.5 concludes the work.

3.2 Method

Let $[x_1, ..., x_n] = \mathbf{x} \in \mathbb{R}^n$ denote the original (clean signal) vector. Here, x_k is the k-th speech sample. Let $[y_1, ..., y_n] = \mathbf{y} \in \mathbb{R}^n$ denote the degraded signal vector.

3.2.1 Speech Degradations - Clipping, Codec Distortions, Gaps in Speech

The first speech degradation modeled is speech clipping (Záviška et al., 2020). Clipping is a non-linear distortion that occurs when the speech signal exceeds the dynamic range of the recording microphone and can be expressed as the element-wise function:

$$y_n = \begin{cases} x_n & \text{if } |x_n| < \theta \\ \theta.\text{sgn}(x_n) & \text{if} |x_n| \geq \theta \end{cases} \tag{3.1}$$

where the threshold θ is called the clipping threshold.

The second speech degradation we model is codec distortion. Speech is commonly encoded prior to transmission (or storage) and decoded only when it needs to be played to reduce the amount of data that needs to be transmitted (or stored). A codec (portmanteau of coder-decoder) is a software written to

compress speech for transmission or storage. To achieve better compression rates, codecs (especially at low bitrates) are lossy (i.e., the decoded data is not an exact match for the original data prior to encoding). This can result in artifacts that are audible to the human ear. Mathematically, we can express the action of a codec as:

$$\mathbf{y} = \boldsymbol{\psi}(\mathbf{x}) \tag{3.2}$$

where the function $\boldsymbol{\psi}(.)$ represents the action of an element-wise codec (e.g. $\mu-$ law compression) or a frame-wise codec (e.g. Adaptive Multi-Rate Wideband (AMR-WB) or MP3).

The third speech degradation modeled is gaps in speech. Poor network conditions result in speech packets being dropped due to packet loss or jitter (Mohamed, Nessiem, and Schuller, 2020). This can be expressed mathematically as a vector Hadamard product:

$$\mathbf{y} = \mathbf{m} \odot \mathbf{x} \tag{3.3}$$

where the mask $\mathbf{m}$ contains 0s and 1s. Contiguous subsets of mask samples with sizes matching the packet size must either be missing together (all 0s) or observed together (all 1s).

3.2.2 Network Architectures

We make extensive use of two U-Net (Ronneberger, Fischer, and Brox, 2015) architectures in our study – T-UNet (see Fig. 3.1), a time domain U-Net for time domain experiments, and TF-UNet (see Fig. 3.2), a time-frequency domain network for time-frequency domain experiments

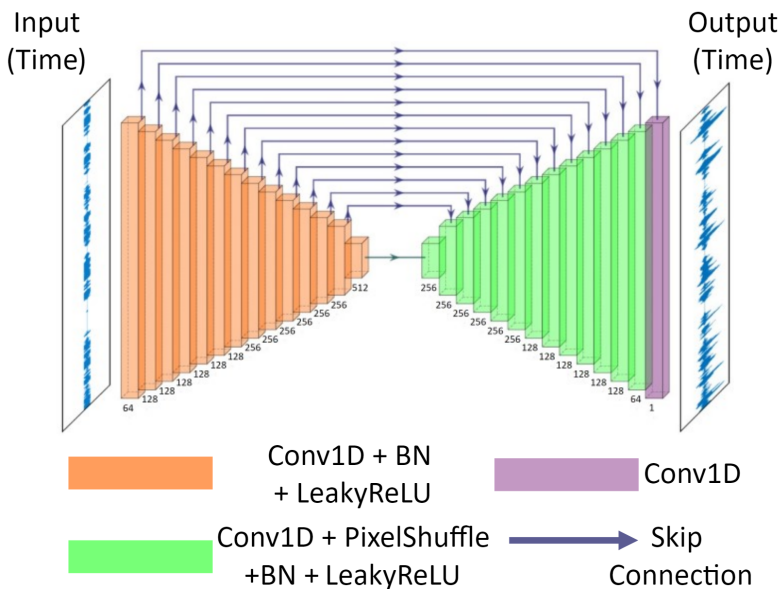


Figure 3.1: T-UNet network architecture used for speech enhancement in this work

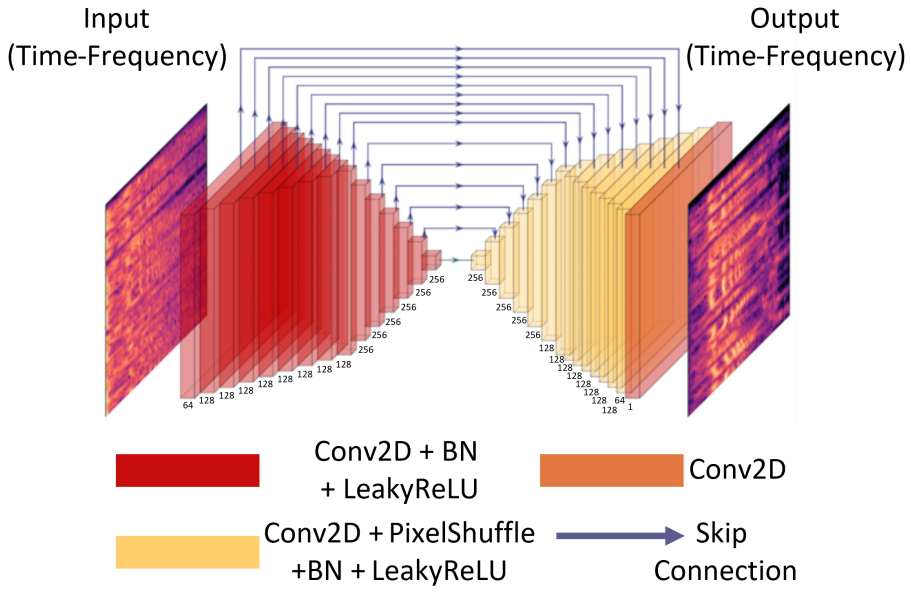


Figure 3.2: TF-UNet network architecture used for speech enhancement in this work

T-UNet was designed based on the structure of the generator in the popular SEGAN (Pascual, Bonafonte, and Serra, 2017) architecture with some key differences – it has smaller convolutional kernels of width 5 and stride 2, is fully deterministic (no Gaussian noise injection in the bottleneck layer), uses LeakyReLU non-linearities and has sub-pixel convolutions in the decoder branch which are demonstrated to work better (Eskimez, Koishida, and Duan, 2019). Further structural details such as number of layers, number of feature maps in each layer etc. can be observed in Fig. 3.1.

The TF-UNet uses the 2D UNet network architecture proposed in Bulut and Koishida, 2020. Key features of it include separable convolutions in the encoder (first downsampling is performed in frequency, then in time), pixel shuffling in the decoder layers, and training with log-spectral distance (LSD) loss. Each of these modifications improve performance on speech tasks (Bulut and Koishida, 2020). See Fig. 3.2 for more details.

As comparing approaches in time and time-frequency for each distortion is of interest, model complexity in both T-UNet and TF-UNet were matched by ensuring the number of learnable parameters are comparable (11,283,585 for T-UNet vs. 9,711,361 for TF-UNet).

3.3 Experiments

3.3.1 Dataset and Degradation Modeling

The Interspeech 2020 Deep Noise Suppression (DNS) Challenge dataset (Reddy et al., 2020) was used to train and test the models. Clean data in the training set of the corpus was corrupted and used as training data. For testing, the

synthetic non-reverberant test data of the corpus was similarly corrupted and used. There is no speaker overlap between training and test sets. All data was sampled at 16 kHz.

For creating clipped data, the parameter θ in Eqn. 3.1 was set as $\theta = a \times x_{\max}$, where $x_{\max}$ is the maximum value of the clean speech $\mathbf{x}$ loaded from a single wav file and the multiplier a is randomly chosen from a truncated random normal distribution with mean 0, standard deviation 0.2, lower threshold of 0.01 and upper threshold of 0.5. This results in a typical speech signal having between 0.36% and 62% of its samples clipped. This generating model was chosen to present more tougher examples to the network while still presenting easy examples so that performance on easy cases does not degrade.

To model codec distortions, the FFmpeg software (*FFmpeg A complete, cross-platform solution to record, convert and stream audio and video.*) was used to degrade clean speech with either the 8-bit $\mu-$law, AMR-WB, or MP3 codec. For the latter two codecs, only a subset of possible bitrates (6.60kbit/s and 23.85kbit/s for AMR-WB, 8kbit/s, 16kbit/s, 24kbit/s, 40kbit/s, and 96kbit/s for MP3) were used for training but testing was done on all codec bitrates to check generalization of the trained networks.

To create gaps in speech, a fixed packet size of 16ms was used because voice packet sizes for VOIP applications are typically 10–20ms long (Mohamed, Nessiem, and Schuller, 2020). Each clean speech utterance was partitioned into 16ms frames and each frame was either set to zero or observed as is according to Eqn. 3.3. The probability of the frame being set to zero was set as

10%, modeling severe VOIP degradation (Kenneth, Mansfield, and Antonakos, 2010).

3.3.2 Data Preprocessing and Network Training Details

For the T-UNet, input is of size 16,384 time samples. This corresponds to $\approx$ 1s of speech considering the dataset is sampled at 16kHz. During test time, the degraded test signal is partitioned into 1s segments which are processed by the network individually before being stitched back together

For the TF-UNet, degraded speech is passed through a 512-point (32 ms) STFT with 50% overlap. The magnitudes of the 257 unique STFT coefficients obtained are squared and log compressed to give the log-power spectrogram (LPS) of the signal. The bin corresponding to the highest frequency STFT coefficient is then removed to get an input with 256 frequencies, a power of 2 as required by the TF-UNet architecture (Bulut and Koishida, 2020). Along the time axis of the LPS, the signal is partitioned into blocks of 64 frames ($\approx$ 1s), making the input to the network of size 256×64. During testing, the enhanced LPS is transformed back into STFT magnitudes before it is combined with the noisy phase of the input and the inverse STFT is applied.

All T-UNets and TF-UNets networks are trained with the Adam optimizer (Kingma and Ba, 2014) with a batch size of 64, a learning rate of 0.0001 and decay rates of $\beta_1 = 0.5$ and $\beta_2 = 0.9$.

3.3.3 Preprocessing Speech Gap Regions

A simple gap identification method using normalized cross correlation (NCC) was developed to aid with gap filling. As packet size (16ms) is assumed known, NCC between $\mathbf{1} - |(\mathbf{y})|$ (where $\mathbf{y}$ is the input signal with speech gaps in time domain) and a vector of all ones of length 16ms should yield a NCC value of 1, enabling gap identification. However, silence regions were also falsely detected as gaps, so the silence detector in Librosa (McFee et al., 2015) was used to suppress them. This simple gap detection algorithm achieved an impressive mean Dice Similarity Coefficient of 0.99 on the test data. After gap identification, a mask $\hat{\mathbf{m}}$ can be constructed which is an estimate of the true gap mask $\mathbf{m}$ in Eqn. 3.3. The estimated mask can then be stacked onto $\mathbf{y}$ along the channel dimension and fed in to the neural network as shown in Fig. 3.3 (a). It is straightforward to use the time domain mask estimate $\hat{\mathbf{m}}$ to create a time-frequency domain mask estimating time-frequency bins affected by gaps by taking the STFT of $\mathbf{1} - \hat{\mathbf{m}}$ and labeling time-frequency bins with energy content as affected by a gap and those without energy as not affected by gaps.

In addition, different initialization methods for the gap regions were studied. Either the gap was initialized as is (i.e., filled with zeros), filled with the average of the previous and next packets in time (T-init), or the gap-affected LPS frame was initialized with the average of the previous and next LPS frames (TF-init).

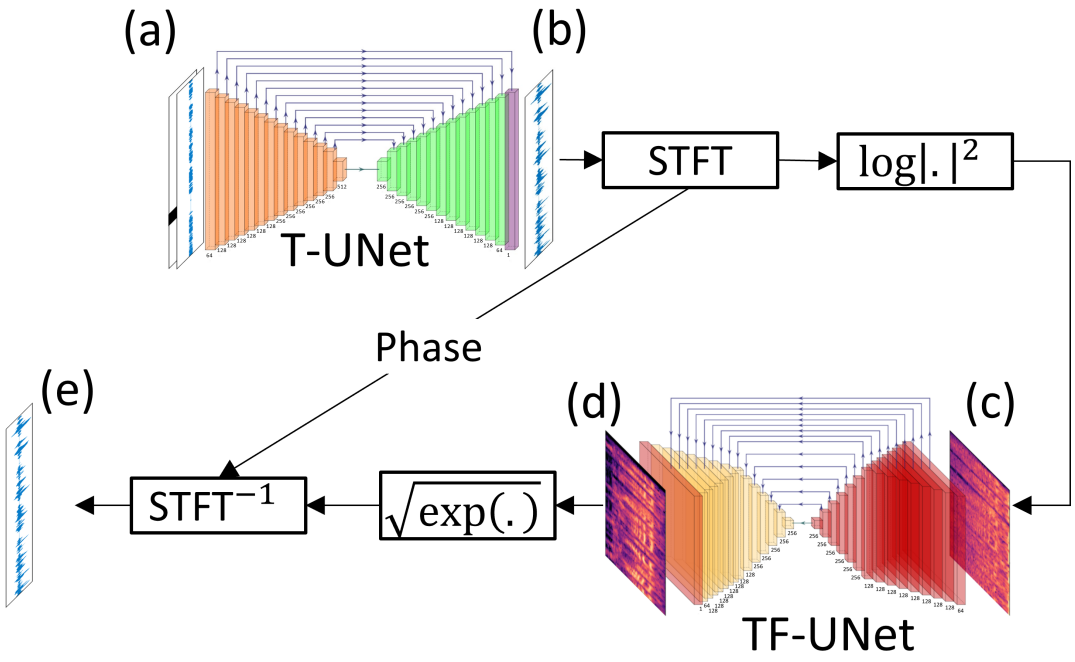


Figure 3.3: Proposed T-UNet + TF-UNet pipeline for jointly addressing clipping, codec distortions, and gaps. Speech with clipping, codec distortions, and gaps along with an optional gap mask (a) is input to a T-UNet trained to remove clipping and gaps and output speech with only codec distortions (b). This speech (b) is transformed into an LPS (c) which is input to a TF-UNet trained to remove codec distortions and produce a clean LPS (d) which is combined with the phase of (b) to produce enhanced speech (e).

3.3.4 Unified T-UNet + TF-UNet Pipeline Training

The proposed T-UNet + TF-UNet pipeline illustrated in Fig. 3.3 contains both a time-domain UNet and a time-frequency domain UNet to leverage the processing strengths of each domain. The training phase for the pipeline is partitioned into two stages. In the first stage, the two networks are trained individually – the T-UNet is trained to perform gap filling and declipping on input data corrupted by all the three degradations of clipping, codec distortions, and gaps (Fig. 3.3 (a)). The target output is data with only codec distortions (Fig. 3.3 (b)) – i.e., the target output is itself degraded. Concurrently,

the TF-UNet is trained to perform codec distortion removal on LPS data with codec distortions (Fig. 3.3 (c)). The target output is the clean speech LPS (Fig. 3.3 (d)). Once both the T-UNet and TF-UNet are trained, the second stage is to fine-tune the TF-UNet on the output of the T-UNet to address any domain shift between the output of the T-UNet and data with only codec distortions.

3.3.5 Studying Phase Distortion Introduced By Clipping, Codec Distortions, and Gaps in Speech

An advantage of T-UNet over TF-UNet is that while TF-UNet only operates on magnitude spectrogram information, T-UNet can operate on phase information as well. This suggests that the more severe the phase distortion, the better T-UNet should perform compared to TF-UNet. The phase distortion introduced by each of clipping, codec distortions, and gaps was studied by combining distorted phase caused by each of the three degradations with clean magnitude information, and measuring the degradation in speech quality.

3.3.6 Evaluation Metrics

Objective evaluation of the enhanced speech was done using the PESQ, CSIG, CBAK, and COVL measures. Each measure compares the enhanced speech with a corresponding ground truth clean speech signal. PESQ, or Perceptual Evaluation of Speech Quality, is a common assessment measure of the speech quality as experienced by a user of a telephony system and returns a value between -0.5 and 4.5. CSIG, CBAK, and COVL are objective measures that aim to predict the subjective Mean Opinion Score (MOS), or how well would an average human listening to the (enhanced speech, clean speech) pair rate

Table 3.1: Declipping Performance

	PESQ	CSIG	CBAK	COVL
Noisy	1.93	3.79	3.74	2.90
Baseline (UNet) (Kashani et al., 2019)	3.68	4.81	4.39	4.41
T-UNet	**3.97**	**4.93**	**4.80**	**4.66**
TF-UNet	3.83	4.89	4.58	4.54

the quality of the enhanced speech, on three different criteria – CSIG predicts signal distortion MOS, CBAK predicts background-noise intrusiveness and COVL predicts overall signal quality MOS. All three produce MOS values from 1 to 5. For all the metrics, a higher score corresponds to better quality enhanced speech.

3.4 Results

3.4.1 Clipping

Performance of the T-UNet and TF-UNet on the declipping task are compared it with a state of the art declipper based on the UNet network (Kashani et al., 2019) trained on our data in Table 3.1. Compared to the baseline, the modifications in the TF-UNet make it better for modeling speech and it consequently performs better. However, the best performing approach to declipping is the proposed T-UNet approach, achieving superior performance according to all measures. These results suggests declipping is a problem best addressed in time domain.

Table 3.2: Codec Distortion Removal Performance

	PESQ	CSIG	CBAK	COVL
Noisy	3.76	4.03	4.20	3.93
Baseline (SEGAN) (Biswas and Jia, 2020)	3.75	4.34	4.32	4.12
T-UNet	3.85	4.41	**4.38**	4.22
TF-UNet	**4.10**	**4.88**	4.37	**4.65**

Table 3.3: Gap Filling Performance

	PESQ	CSIG	CBAK	COVL
Noisy	1.89	4.18	4.29	3.08
Baseline (SEGAN) (Shi et al., 2019)	3.08	4.88	4.80	4.05
T-UNet	3.26	4.93	**4.82**	4.18
+ mask	3.24	4.92	4.77	4.16
+ mask + T-init	**3.38**	4.96	4.70	**4.28**
TF-UNet	2.96	4.84	4.71	3.96
+ mask	2.96	4.84	4.70	3.96
+ mask + T-init	3.25	**4.97**	**4.82**	4.19
+ mask + TF-init	2.95	4.83	4.70	3.95

3.4.2 Codec Distortions

T-UNet and TF-UNet approaches to codec distortion removal were compared with a state of the art SEGAN baseline (Biswas and Jia, 2020) trained on our data in Table 3.2. In comparison, our T-UNet and TF-UNet achieve better results. Best overall results are obtained with the TF-UNet which suggests codec distortion removal is best done in the time-frequency domain.

3.4.3 Gaps in Speech

The performance of the different networks trained to fill in speech gaps is compared with a state of the art SEGAN baseline (Shi et al., 2019) trained on our data in Table 3.3. T-UNet outperforms the baseline, while TF-UNet

Table 3.4: Performance addressing clipping, codec distortions, and gaps in speech jointly

	PESQ	CSIG	CBAK	COVL
Noisy	1.35	2.75	2.61	2.05
T-UNet	2.61	3.84	3.60	3.26
TF-UNet	2.76	4.22	3.41	3.52
T-UNet (Large)	2.56	3.80	3.55	3.21
TF-UNet (Large)	2.76	4.26	3.41	3.54
T-UNet+TF-UNet (w/o fine-tune)	2.62	3.23	3.36	2.94
T-UNet+TF-UNet (w/ fine-tune)	**3.46**	**4.63**	**3.87**	**4.10**

does not. On studying example outputs from both networks, it was observed that the biggest issue with the enhanced outputs was not incorrect speech reconstruction but a lack of reconstruction of the missing speech in gap regions. Consequently, as described in Section 3.3.3, explicit supervision was provided to the network on gap locations by inputting a gap mask $\hat{\mathbf{m}}$ as well as better initializing gap regions by filling in the average of neighboring samples either in time (T-init) or time-frequency (TF-init). From the experiments, using both a gap mask and T-init together works best for both T-UNet and TF-UNet, with the former performing as well or better than the latter (except on the CBAK measure), suggesting the T-UNet is overall better suited to performing gap filling.

3.4.4 Jointly Addressing Clipping, Codec Distortions, and Gaps

We observe the performance of networks trained to jointly address all three distortions – clipping, codecs and gaps – in Table 3.4. Compared to training a single T-UNet or TF-UNet to handle all three distortions, better performance

can be obtained by incorporating the knowledge gained in Sections 3.4.1–3.4.3 that clipping and gaps are best addressed in time and codec distortions in time-frequency. A pipeline tailored to the task was designed that cascades a T-UNet with a TF-UNet and trained as described in section 3.3.4. A first attempt of training a T-UNet and TF-UNet in parallel and connecting them together (T-UNet + TF-UNet (w/o fine-tune) in Table 3.4) was unsuccessful because of the domain shift between the output of the T-UNet trained to remove clipping and gaps and data corrupted only by codec distortions. On fine-tuning the TF-UNet on the output of the T-UNet, this domain shift was bridged and significantly better results were obtained ((T-UNet + TF-UNet (w/ fine-tune) in Table 3.4). The performance of the proposed pipeline remained superior even when larger networks (denoted as T-UNet (Large) and TF-UNet (Large) in Table 3.4) with more filters per layer to match the number of parameters in our T-UNet + TF-UNet pipeline were trained on the joint distortion removal task.

3.4.5 Discussion

We attempted to understand why T-UNet performs better than TF-UNet on the declipping and gap filling tasks by studying phase distortion as described in Section 3.3.5, using PESQ to evaluate speech quality. We observed that codec distortions produced less phase distortion, lowering PESQ by an average of 0.26, while clipping and speech gaps caused more phase distortion, lowering PESQ by an average of 0.32 and 1.25, respectively, and are the two cases where T-UNet outperformed TF-UNet. This observation confirms that T-UNet

performs better relative to TF-UNet in the presence of more severe phase distortion as T-UNet enhances phase information as well, and highlights the importance of choosing the right domain (time vs. time-frequency) in which to process each distortion.

3.5 Conclusion

In this work we systematically study the problem of enhancing speech suffering from three degradations – clipping, codec distortions, and gaps – using DNNs in the time and time-frequency domains. We achieve state of the art performance on each degradation before developing a neural network pipeline consisting of cascaded time domain and time-frequency domain UNets to address all three distortions together. The cascaded pipeline developed nears the performance ceiling set by the most challenging single distortion of gaps in speech while simultaneously allowing the function of each component network to remain interpretable.

References

Wang, Dequan and Jae Lim (1982). "The unimportance of phase in speech enhancement". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 30.4, pp. 679–681.

Hu, Yanxin, Yun Liu, Shubo Lv, Mengtao Xing, Shimin Zhang, Yihui Fu, Jian Wu, Bihong Zhang, and Lei Xie (2020). "DCCRN: Deep Complex Convolution Recurrent Network for Phase-Aware Speech Enhancement". In: *Proc. Interspeech 2020*, pp. 2472–2476.

Isik, Umut, Ritwik Giri, Neerad Phansalkar, Jean-Marc Valin, Karim Helwani, and Arvindh Krishnaswamy (2020). "PoCoNet: Better Speech Enhancement with Frequency-Positional Embeddings, Semi-Supervised Conversational Data, and Biased Loss". In: *Proc. Interspeech 2020*, pp. 2487–2491.

Nair, Arun Asokan and Kazuhito Koishida (2021). "Cascaded Time + Time-Frequency UNET for Speech Enhancement: Jointly Addressing Clipping, Codec Distortions, and Gaps". In: *2021 IEEE International conference on acoustics, speech and signal processing (ICASSP)*. IEEE.

Loizou, Philipos C (2013). *Speech enhancement: theory and practice*. CRC press.

Lee, Kai-Fu (1988). *Automatic speech recognition: the development of the SPHINX system*. Vol. 62. Springer Science & Business Media.

Wiener, Norbert (1950). *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. MIT press.

Boll, Steven (1979). "Suppression of acoustic noise in speech using spectral subtraction". In: *IEEE Transactions on acoustics, speech, and signal processing* 27.2, pp. 113–120.

Ephraim, Yariv and David Malah (1984). "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator". In: *IEEE Transactions on acoustics, speech, and signal processing* 32.6, pp. 1109–1121.

Kumar, Anurag and Dinei Florencio (2016). "Speech Enhancement In Multiple-Noise Conditions using Deep Neural Networks". In:

Xu, Yong, Jun Du, Li-Rong Dai, and Chin-Hui Lee (2013). "An experimental study on speech enhancement based on deep neural networks". In: *IEEE Signal processing letters* 21.1, pp. 65–68.

Xu, Yong, Jun Du, Li-Rong Dai, and Chin-Hui Lee (2014). "A regression approach to speech enhancement based on deep neural networks". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.1, pp. 7–19.

Lu, Xugang, Yu Tsao, Shigeki Matsuda, and Chiori Hori (2013). "Speech enhancement based on deep denoising autoencoder." In: *Interspeech*. Vol. 2013, pp. 436–440.

Park, Se Rim and Jin Won Lee (2017). "A Fully Convolutional Neural Network for Speech Enhancement". In: *Proc. Interspeech 2017*, pp. 1993–1997.

Weninger, Felix, Hakan Erdogan, Shinji Watanabe, Emmanuel Vincent, Jonathan Le Roux, John R Hershey, and Björn Schuller (2015). "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR". In: *International Conference on Latent Variable Analysis and Signal Separation*. Springer, pp. 91–99.

Pascual, Santiago, Antonio Bonafonte, and Joan Serra (2017). "SEGAN: Speech enhancement generative adversarial network". In: *arXiv preprint arXiv:1703.09452*.

Stoller, Daniel, Sebastian Ewert, and Simon Dixon (2018). "Wave-u-net: A multi-scale neural network for end-to-end audio source separation". In: *arXiv preprint arXiv:1806.03185*.

Luo, Yi and Nima Mesgarani (2019). "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation". In: *IEEE/ACM transactions on audio, speech, and language processing* 27.8, pp. 1256–1266.

Bulut, Ahmet E and Kazuhito Koishida (2020). "Low-Latency Single Channel Speech Enhancement Using U-Net Convolutional Neural Networks". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 6214–6218.

Williamson, Donald S and DeLiang Wang (2017). "Speech dereverberation and denoising using complex ratio masks". In: *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, pp. 5590–5594.

Paliwal, Kuldip, Kamil Wójcicki, and Benjamin Shannon (2011). "The importance of phase in speech enhancement". In: *speech communication* 53.4, pp. 465–494.

Záviška, Pavel, Pavel Rajmic, Alexey Ozerov, and Lucas Rencker (2020). "A survey and an extensive evaluation of popular audio declipping methods". In: *arXiv preprint arXiv:2007.07663*.

Kashani, Hamidreza Baradaran, Ata Jodeiri, Mohammad Mohsen Goodarzi, and Shabnam Gholamdokht Firooz (2019). "Image to Image Translation based on Convolutional Neural Network Approach for Speech Declipping". In: *arXiv preprint arXiv:1910.12116*.

Mack, Wolfgang and Emanuël AP Habets (2019). "Declipping Speech Using Deep Filtering". In: *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, pp. 200–204.

Biswas, Arijit and Dai Jia (2020). "Audio codec enhancement with generative adversarial networks". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 356–360.

Zhao, Ziyue, Huijun Liu, and Tim Fingscheidt (2018). "Convolutional neural networks to enhance coded speech". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.4, pp. 663–678.

Deng, Jun, Björn Schuller, Florian Eyben, Dagmar Schuller, Zixing Zhang, Holly Francois, and Eunmi Oh (2020). "Exploiting time-frequency patterns with LSTM-RNNs for low-bitrate audio restoration". In: *Neural Computing and Applications* 32.4, pp. 1095–1107.

Mohamed, Mostafa M, Mina A Nessiem, and Björn W Schuller (2020). "On Deep Speech Packet Loss Concealment: A Mini-Survey". In: *arXiv preprint arXiv:2005.07794*.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241.

Eskimez, Sefik Emre, Kazuhito Koishida, and Zhiyao Duan (2019). "Adversarial training for speech super-resolution". In: *IEEE Journal of Selected Topics in Signal Processing* 13.2, pp. 347–358.

Reddy, Chandan KA, Vishak Gopal, Ross Cutler, Ebrahim Beyrami, Roger Cheng, Harishchandra Dubey, Sergiy Matusevych, Robert Aichner, Ashkan Aazami, Sebastian Braun, et al. (2020). "The INTERSPEECH 2020 Deep Noise Suppression Challenge: Datasets, Subjective Testing Framework, and Challenge Results". In: *arXiv preprint arXiv:2005.13981*.

FFmpeg A complete, cross-platform solution to record, convert and stream audio and video. https://ffmpeg.org/.

Kenneth, C, Jr Mansfield, and JL Antonakos (2010). *Computer Networking from LANs to WANs: Hardware, Software, and Security*.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto (2015). "librosa: Audio and music signal analysis in python". In: *Proceedings of the 14th python in science conference*. Vol. 8, pp. 18–25.

Shi, Yupeng, Nengheng Zheng, Yuyong Kang, and Weicong Rong (2019). "Speech Loss Compensation by Generative Adversarial Networks". In: *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, pp. 347–351.

Chapter 4

Robust Short-Lag Spatial Coherence Imaging

In this chapter, we demonstrate how the performance of Short-Lag Spatial Coherence (SLSC) imaging (Lediju et al., 2011) can be improved using Robust Principal Component Analysis (RPCA) (Candès et al., 2011). SLSC imaging displays the spatial coherence between backscattered ultrasound echoes instead of their signal amplitudes and is more robust to noise and clutter artifacts when compared to traditional delay-and-sum (DAS) B-mode imaging. However, SLSC imaging does not consider the content of images formed with different lags, and thus does not exploit the differences in tissue texture at each short lag value. Our proposed method improves SLSC imaging by weighting the addition of lag values (i.e., M-weighting) and by applying RPCA to search for a low dimensional subspace for projecting coherence images created with different lag values. The RPCA-based projections are considered to be de-noised versions of the originals that are then weighted and added across lags to yield a final Robust Short-Lag Spatial Coherence (R-SLSC) image. Our approach was tested on simulation, phantom, and *in vivo* liver data. Relative

to DAS B-mode images, the mean contrast, signal-to-noise ratio (SNR), and contrast-to-noise ratio (CNR) improvements with R-SLSC images are 21.22 dB, 2.54 and 2.36 respectively, when averaged over simulated, phantom, and *in vivo* data and over all lags considered which corresponds to mean improvements of 96.4%, 121.2% and 120.5% respectively. When compared to SLSC images, the corresponding mean improvements with R-SLSC images were 7.38 dB, 1.52 and 1.30, respectively, (i.e., mean improvements of 14.5%, 50.5% and 43.2%, respectively). Results show great promise for smoothing out the tissue texture of SLSC images and enhancing anechoic or hypoechoic target visibility at higher lag values which could be useful in clinical tasks such as breast cyst visualization, liver vessel tracking, and obese patient imaging. The work presented in this chapter was published earlier in Nair, Tran, and Bell, 2017.

4.1 Introduction

Displaying the spatial coherence of backscattered ultrasound waves is a promising alternative to generate ultrasound image contrast when compared to traditional, amplitude-based delay-and-sum (DAS) beamforming. This alternative is motivated by the van Cittert Zernike (VCZ) theorem applied to ultrasound (Cittert, 1934; Zernike, 1938; Goodman, 2015), which states that for an incoherent source and a spatially incoherent medium, the expected spatial coherence is the squared Fourier transform of the product of the transmit beam intensity distribution and the reflectivity profile of the insonified medium.

The VCZ theorem supported ultrasound-based investigations by Mallart

and Fink, 1991, Liu and Waag, 1995, and Bamber, Mucci, and Orofino, 2002, and led to the development of short-lag spatial coherence (SLSC) (Lediju et al., 2011) imaging. SLSC imaging has since demonstrated remarkable improvements over traditional ultrasound B-mode imaging when visualizing liver tissue (Jakovljevic et al., 2013), endocardial borders (Bell et al., 2013a), fetal anatomical features (Kakkad et al., 2013), and point-like targets in the presence of noise (Bell, Dahl, and Trahey, 2015). A suite of traditional ultrasound transducer arrays (i.e., linear (Lediju et al., 2011), curvilinear (Jakovljevic et al., 2013), phased (Bell et al., 2013a), and 2D matrix (Hyun et al., 2014; Jakovljevic et al., 2014) arrays) were demonstrated to be compatible with SLSC imaging. This new imaging method was additionally extended to photoacoustic imaging to improve the visibility of prostate brachytherapy seeds (Bell et al., 2013b), to improve signal contrast when imaging with low-energy, pulsed laser diodes (Bell et al., 2014) and to potentially guide minimally invasive surgeries (Gandhi et al., 2017). Additional work in this area has weighted SLSC images with traditional DAS images (Alles, Jaeger, and Bamber, 2014) and utilized SLSC beamforming to reduce clutter and sidelobes in photoacoustic images (Pourebrahimi et al., 2013).

SLSC imaging is implemented by computing the spatial correlation between received signals at various element separations (or lags), then summing across the lags to generate the final output image. In doing so, SLSC imaging inherently weights all lags equally and does not consider differences in tissue texture appearances when SLSC images are formed with various combinations of lag values. One possibility to consider texture differences is to apply

uneven weighting to the lag images prior to summation. Another possibility is to apply linear dimensionality reduction.

Principal component analysis (PCA) (Jolliffe, 1986) is a popular method for linear dimensionality reduction, with wide-ranging domains of application that include data mining (Han, Kamber, and Pei, 2011), neuroscience (Turk and Pentland, 1991), and linear control systems (Moore, 1981). PCA finds the orthogonal directions of highest variance by taking the singular value decomposition of a data matrix and preserving the subspace corresponding to the largest singular values. Assuming that data is corrupted by dense, low-magnitude, Gaussian noise, PCA returns the maximum likelihood estimate for an underlying subspace (Bishop, 2006). Projecting data onto this low-dimensional, underlying subspace, then re-projecting to a high dimensional space is generally a useful denoising technique that eliminates spurious directions of variance corresponding to noise in the data.

PCA was successfully applied to various ultrasound imaging tasks, including motion estimation (by leveraging its signal separation capabilities to reject decorrelation and noise) (Mauldin, Viola, and Walker, 2010) and on-line classification of arterial stenosis intensity (Prytherch et al., 1982). However, one limitation of PCA is that it lacks robustness (Wright et al., 2009) and displays a high sensitivity to outliers.

Robust Principal Component Analysis (RPCA) (Wright et al., 2009; Lin, Chen, and Ma, 2010; Lin et al., 2009) was developed to recover a low rank matrix from a matrix of corrupted observations, particularly when the errors are arbitrarily large. In addition, as stated in Wright et al., 2009, in most cases

the low rank matrix can be recovered from most common corruptions by solving a convex optimization problem. In the context of ultrasound imaging, RPCA was utilized to automatically classify acoustic radiation force impulse (ARFI) displacement profiles in the presence of high variance outlier profiles (Mauldin Jr et al., 2008) and to implement motion-based clutter reduction (Lediju et al., 2009).

In this work, we propose a modification to the SLSC algorithm to explicitly consider the content of coherence images formed with different lags by applying RPCA to first search for a low dimensional subspace, then project individual coherence images onto this low dimensional subspace. We assume that this approach enables us to denoise the observations at higher lags and incorporate them in our imaging pipeline. The projections are denoised versions of the originals that are then weighted and summed across the lags to yield the final Robust Short-Lag Spatial Coherence (R-SLSC) image. We also consider the effect of weighting without applying RPCA.

Our work is organized as follows: Section 4.2 details the background that motivated this work, specifically the SLSC algorithm and the RPCA algorithm. Section 4.3 describes our proposed R-SLSC method in detail. Section 4.4 provides details about our simulation, phantom and experimental data and related evaluation metrics. Section 4.5 presents the results of our study, while section 4.6 discusses the strengths and limitations of the proposed algorithm. We conclude our work in section 4.7.

4.2 Background

4.2.1 Short-Lag Spatial Coherence (SLSC) Imaging

SLSC beamforming (as discussed extensively in (Lediju et al., 2011; Bell, Dahl, and Trahey, 2015; Dahl et al., 2011)) computes and displays the spatial coherence between backscattered ultrasound echoes at different short lag values, and thereby removes clutter artifacts. The ultrasound channel data consists of echoes received by N equi-spaced detector elements of an array. Assuming s_i is the time-delayed, zero mean data received by the i^{th} detector element, let a measurement corresponding to the n^{th} depth (in samples) of this data be the signal $s_i(n)$. The spatial covariance across the face of the aperture is evaluated as:

$$\hat{C}(m) = \frac{1}{N-m} \sum_{i=1}^{N-m} \sum_{n=n_1}^{n_2} s_i(n) s_{i+m}(n) \tag{4.1}$$

where m is the lag (in number of elements) between two detector elements of the array. The size of the correlation kernel (i.e., $n_2 - n_1$) is fixed to be approximately one wavelength in order to maintain an axial resolution similar to that of DAS B-mode images without compromising the stability of the calculated coherence functions.

Eq. (4.1) is normalized by the individual variances of the two scan lines being considered, and the spatial correlation $\hat{R}$ at lag m is:

$$\hat{R}(m) = \frac{1}{N-m} \sum_{i=1}^{N-m} \frac{\sum_{n=n_1}^{n_2} s_i(n) s_{i+m}(n)}{\sqrt{\sum_{n=n_1}^{n_2} s_i^2(n) \sum_{n=n_1}^{n_2} s_{i+m}^2(n)}} \tag{4.2}$$

which results in a spatial coherence function. We integrate this spatial coherence function over the first M lags to achieve a SLSC image pixel:

$$R_{sl} = \int_{m=1}^{M} \hat{R}(m)dm \approx \sum_{m=1}^{M} \hat{R}(m) \tag{4.3}$$

Eqs. (4.1)-(4.3) are repeated at various axial and lateral positions to generate a SLSC image.

The coherence functions scale with the size of the aperture, thus M is expressed in terms of a quantity Q, which is defined to be the percentage fraction of the receive aperture over which we are summing, i.e.:

$$Q = \frac{M}{N} \times 100\% \tag{4.4}$$

in order to standardize across various receive aperture sizes.

4.2.2 Robust Principal Component Analysis (RPCA)

RPCA (Wright et al., 2009; Lin, Chen, and Ma, 2010) is implemented by finding a low-rank approximation A of a noisy observation matrix D, which can be expressed as:

$$D = A + E + N \tag{4.5}$$

where A is the low-rank ground truth matrix, E is an error matrix which is considered to be sparse but allowed to have high magnitude errors, while N contains dense, low-magnitude errors. The main objective is to calculate the lowest rank A that approximates the data subject to the outlier errors being sparse i.e. $\|E\|_0 \leq K$ for some appropriately chosen threshold K (where $\|.\|_0$ is the L_0 norm, which counts the number of non-zero entries in E). Writing out

the Lagrangian formulation, we obtain:

$$\min_{A,E} \text{Rank}(A) + \lambda \|E\|_0 \text{ subject to } D = A + E + N \approx A + E \tag{4.6}$$

where λ is a penalty factor based on the quantity of outliers present in data. Note that Eq. (4.6) is difficult to optimize as it is non-convex. Relaxing the rank constraint to a nuclear norm constraint and the L_0 norm constraint to an L_1 norm constraint, we rewrite Eq. (4.6) as:

$$\min_{A,E} \|A\|_* + \lambda \|E\|_1 \text{ subject to } D \approx A + E \tag{4.7}$$

where the nuclear norm, $\|.\|_*$, is the sum of the singular values of a matrix. This relaxation is reasonable because the solution to (4.7) is almost always the same as the solution to (4.6), as proved in Wright et al., 2009.

To solve Eq. (4.7), we utilized a numerical optimization method based on the Augmented Lagrangian Multiplier (ALM) (Lin, Chen, and Ma, 2010) method. This solver relaxes Eq. (4.7) by solving for the minimum of the Lagrangian $L(A, E, Y, \mu)$ of the problem, where $L(A, E, Y, \mu)$ is defined as:

$$L(A, E, Y, \mu) = \|A\|_* + \lambda \|E\|_1 + \langle Y, D - A - E \rangle$$

$$+ \frac{\mu}{2} \|D - A - E\|_F^2$$

We used the MATLAB inexact ALM solver based on Lin, Chen, and Ma, 2010 and hosted at *https://people.eecs.berkeley.edu/ yima/matrix-rank/sample_code.html* to perform RPCA.

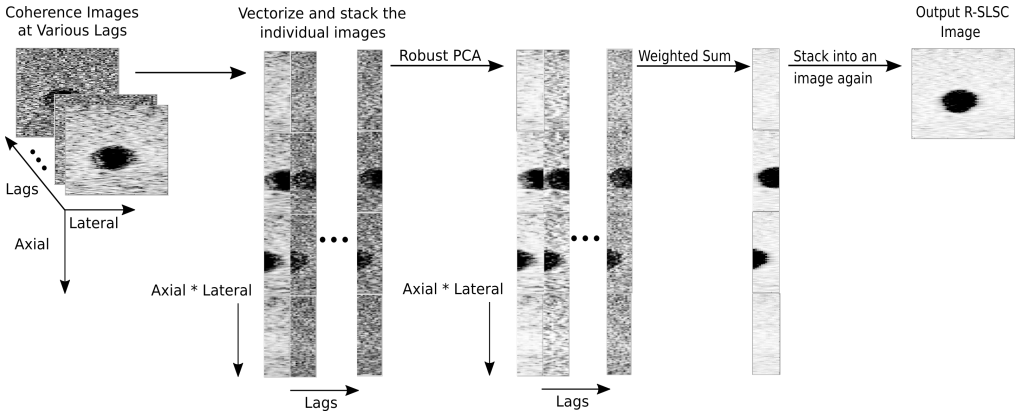
4.3 Proposed Algorithm

4.3.1 Robust Short-Lag Spatial Coherence (R-SLSC) Imaging

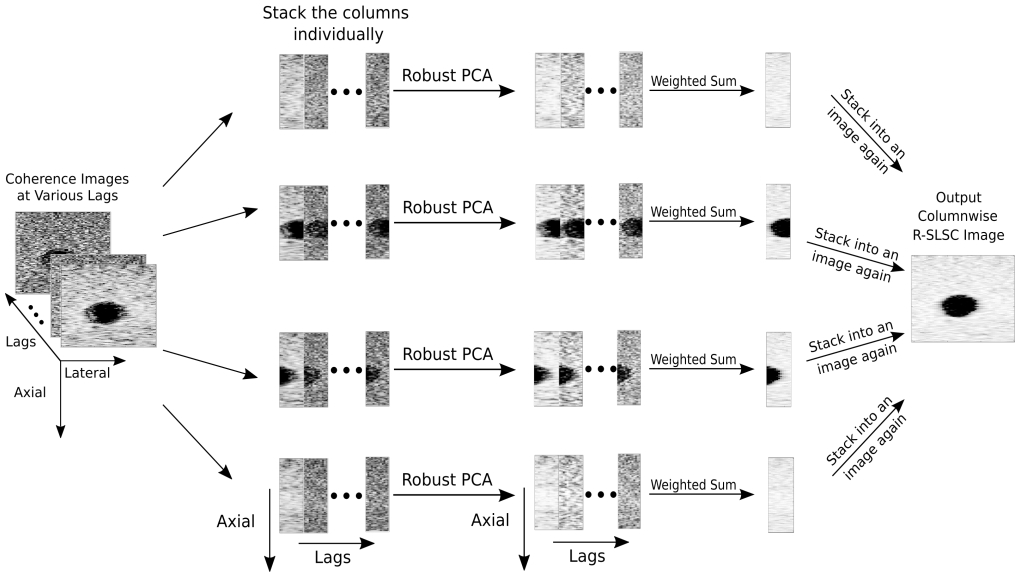
If we define outliers in SLSC images as pixels with coherence values that differ significantly from their surroundings and from their values at other lags, we observe that SLSC images formed with higher lags tend to have more outliers (Bell, 2012). These outliers adversely affect contrast, and thus reduce the diagnostic utility of SLSC imaging. Consequently, we hypothesize that filtering out these coherence outliers is an important step in order to consider the additional information that is provided at higher lag values.

We also hypothesize that because each image corresponds to an observation of the same ground truth, we can treat the images at the different lags as noisy, corrupted versions of this ground truth, each affected differently by clutter and coherence outliers. We can thus reformulate finding the optimal summation of the coherence images as a RPCA application (Wright et al., 2009; Lin, Chen, and Ma, 2010; Lin et al., 2009) and we call this combination R-SLSC.

The first step of R-SLSC is to perform SLSC beamforming and generate the coherence images at various lags. Each of these lag images is then vectorized as illustrated in Fig. 4.1a. The vectorized lag images (up to a specific lag M) are stacked horizontally to form the noisy data matrix D. This matrix D is then fed into the RPCA algorithm, which returns a low rank estimate that corresponds to A in Eq. (4.7), which is the denoised data matrix, with both coherence outliers (stored in E) and low magnitude dense noise (stored in N) removed. We then apply a weighted sum across the columns to generate



(a)



(b)

Figure 4.1: (a) Summary of the the whole-image R-SLSC imaging process. The individual coherence images up to a specific lag M are vectorized and stacked into a matrix. RPCA is performed on this data matrix, and the denoised coherence images are weighted and summed across the lag dimension. Finally, the vectorization is reversed to yield the output R-SLSC image at lag M. (b) Columnwise R-SLSC imaging is similar, with the exception that the whole image is subdivided into individual columns for the denoising step. Patchwise R-SLSC imaging (not shown) denoises individual patches rather than columns.

the vectorized output R-SLSC image corresponding to lag M. The weighting applied could be uniform (as in traditional SLSC imaging), but we apply a linearly decreasing weighting scheme (weight 1 to lag image 1, weight $\frac{M-1}{M}$ to lag image 2, ..., weight $\frac{1}{M}$ to lag image M) to enforce our prior knowledge that SLSC image characteristics such as Contrast, CNR, SNR are superior in the short-lag region. We call this weighting scheme linear M-weighting. With linear M-weighting, the higher lag value observations are primarily used to refine our estimate of the data subspace for A in Eq. (4.7). The final step involves reshaping the vectorized image to obtain the output R-SLSC image corresponding to lag M.

We additionally note that we can vary the λ parameter (see Eqn. 4.7) to apply a penalty factor to the quantity of coherence outliers present. The λ value reported throughout this chapter is multiplied by $\frac{1}{\sqrt{size(D,1)}}$, where D is the data matrix being considered. We chose λ to equal 1 unless otherwise stated.

4.3.2 Columnwise and Patchwise R-SLSC Imaging

With the addition of RPCA to SLSC imaging, one expected concern with R-SLSC imaging is the additional processing time. While real-time SLSC imaging has previously been demonstrated (Hyun, Trahey, and Dahl, 2013; Hyun, Trahey, and Dahl, 2015), performing real-time R-SLSC on the entire image is not possible as currently implemented.

The bottleneck in R-SLSC processing times is the Singular Value Decomposition (SVD) step of the RPCA algorithm. The time complexity, O, of SVD

is generally $O(min(mn^2, m^2n))$, where m is the number of rows of the data matrix D and n is the number of columns (Holmes, Gray, and Isbell, 2007). Thus, we hypothesize that subdividing the large SVD problem into smaller SVDs, each solved independently using parallel computing, will increase algorithm speed.

We experimented with two methods for subdividing our problem:

- Columnwise R-SLSC (summarized in Fig. 4.1b)
- Patchwise R-SLSC

To implement columnwise R-SLSC, the first step entails performing SLSC beamforming and generating the coherence images at the various lags. However, instead of vectorizing the images, we extract a specific column from each of these lag images (up to a specific lag M) and stack these extracted columns horizontally to form the noisy data matrix D as illustrated in Fig. 4.1b. We repeat this process across all columns to achieve n independent RPCA subproblems (where n is the number of columns). The RPCA subproblems are then solved, and the results from each are combined to obtain the final columnwise R-SLSC image corresponding to lag M.

The process for patchwise R-SLSC is similar, with the exception that the independent subproblems correspond to patches and not columns.

Table 4.1: Ultrasound Transducer and Image Acquisition Parameters

	Experiments	PICMUS
Aperture Width	19.2 mm	38.4 mm
Element Width	0.24 mm	0.27 mm
Number of Receive Elements	64	128
Pitch	0.30 mm	0.30 mm
Transmit Frequency	8 MHz	5.208 MHz
Sampling Frequency	40 MHz	20.832 MHz
Pulse Bandwidth	61%	67%

4.4 Evaluation Methods

4.4.1 Simulation Data

Field II (Jensen, 1996; Jensen and Svendsen, 1992) was used to generate a numerical phantom of width 50 mm, height 60 mm (located between 30 mm and 90 mm depth) and transverse width 10 mm. A total of 3,141,360 scatterers (corresponding to 20 scatterers per resolution cell) were randomly placed in this volume, with amplitudes that were randomly drawn from a standard normal distribution. An anechoic cyst of diameter 4 mm was centered at a depth of 60mm. Focused transmits with dynamic receive were used to image the cyst. The parameters of the simulated probe matched those of the Alpinion L3-8 linear array transducer which was used to acquire experimental data (see Table 4.1 for transducer and image acquisition parameters). The sampling frequency was 40 MHz, and the center frequency was 8.0 MHz. Additive white Gaussian noise of SNR -10 dB was added to the channel data and the summed signal was bandpass filtered with cutoff frequencies equal to the -6 dB cutoff frequencies of the ultrasound transducer in order to simulate acoustic noise received by the transducer (Bell, Dahl, and Trahey, 2015; Dahl

et al., 2011).

4.4.2 Experimental Phantom and In Vivo Data

Ultrasound data was acquired with an Alpinion E-Cube 12R connected to an L3-8 linear ultrasound transducer. An 8mm diameter cylindrical anechoic cyst target of a CIRS Model 054GS ultrasound phantom at a depth of 4cm was insonified. The sampling frequency of the probe was 40 MHz and the center frequency for the transmission was 8.0 MHz. The probe possessed 128 elements, with only 64 allowed to receive simultaneously at any point in time. Additional transducer and image acquisition parameters are listed in Table 4.1.

Using the same ultrasound system, a 4mm diameter vessel located at a depth of 34mm in the liver of a healthy female was imaged with approval from the Johns Hopkins University Institutional Review Board (Protocol HIRB00005688). The patchwise and columnwise R-SLSC methods were only applied to this *in vivo* dataset. CPU parallelization was performed using the *parfor* subroutine in MATLAB on an Intel(R) Core(TM) i7-4720HQ CPU with a clock speed of 2.60 GHz. This *in vivo* dataset was additionally used to experiment with the direct display of M-weighted SLSC images without applying RPCA and to experiment with the optimal λ parameter for R-SLSC imaging.

4.4.3 Plane Wave Data

In addition to simulation and experimental data acquired with focused transmits, we tested our algorithm on the publicly available plane wave experimental data provided through the Plane-Wave Imaging Challenge in Medical Ultrasound (PICMUS) (Liebgott et al., 2016), which was organized for the 2016 IEEE International Ultrasonics Symposium. The data consisted of 75 steered plane wave sequences with an angular range of -16 degrees to +16 degrees, acquired with a Verasonics Vantage 256 research scanner and a L11 probe (Verasonics Inc., Redmond WA). The probe specifications and acquisition parameters are reported in Table 4.1.

A CIRS Multi-Purpose Ultrasound Phantom (Model 040GSE) was imaged using this setup. Specifically, the region corresponding to a -3dB and a +3dB cyst set against a speckle background with a pair of anechoic targets was recorded. Both cysts are located at a depth of 3cm and have diameters of 8 mm, while the anechoic targets are located at depths of 15mm and 45mm, and are smaller with a diameter of 3mm. The anechoic target located at 45mm depth was the focus of our study, as highlighted by the red box in Fig. 4.2.

4.4.4 Image Quality Metrics

The contrast, signal-to-noise ratio (SNR) and contrast-to-noise ratio (CNR) metrics were calculated for each data set, as:

$$Contrast = 20\log_{10}\left(\frac{S_i}{S_o}\right) \tag{4.8}$$

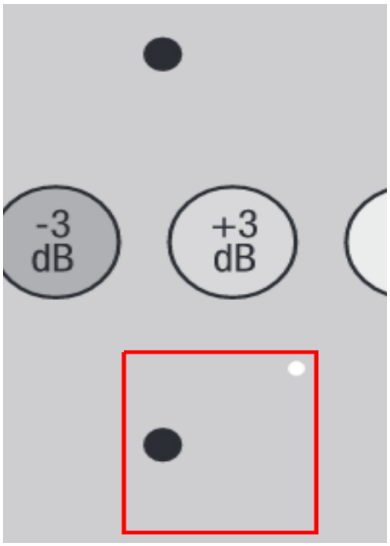


Figure 4.2: Schematic diagram of phantom used for the plane wave data. The red rectangle shows the anechoic target of interest for our study.

with S_i and S_o representing the mean signal intensities inside and outside selected regions of interest (ROIs) at the same image depth.

$$SNR = \frac{S_o}{\sigma_o} \tag{4.9}$$

where σ_o is the standard deviation of the background ROI.

$$CNR = \frac{|S_i - S_o|}{\sqrt{\sigma_i^2 + \sigma_o^2}} \tag{4.10}$$

where σ_i is the standard deviation of the signal in the chosen ROI.

Note that SLSC images can contain negative pixels due to potential negative correlations from signals that are out of phase. However, we observed that these negative values mostly appear in anechoic or hypoechoic regions, and they are not significant (i.e., they are closer to 0 than -1). When log

compressing an image with negative values, the negative correlations are converted to positive values that degrade the image quality. Hence, our approach when calculating our quality metrics and displaying our images was to set all negative SLSC image pixels to zero.

To evaluate the PICMUS data and to enable past and future users of the PICMUS dataset to compare their results with our method, we additionally report a modified version of the contrast evaluation script provided by the PICMUS challenge organizers. The modified script calculates contrast as:

$$PICMUS\ Contrast = 20\log_{10}\left(\frac{|S_i - S_o|}{\sqrt{\frac{\sigma_i^2+\sigma_o^2}{2}}}\right) \tag{4.11}$$

All data analysis and beamforming was performed in MATLAB (MathWorks Inc., Natick, MA).

4.5 Results

4.5.1 Correlation Curves

The VCZ theorem predicts that when imaging diffuse scatterers like tissue, the expected spatial correlation across the receive aperture is a triangle, with a peak of 1 at lag 0 and a minimum of 0 at lag $N-1$, where N is the total number of elements in the transmit aperture. However, when imaging anechoic or hypoechoic regions (like the cyst or the vessel), the spatial correlation is expected to significantly drop from 1 to 0 in the short-lag region, with low magnitude oscillations about 0 as lag increases beyond the initial drop (Lediju et al., 2011).

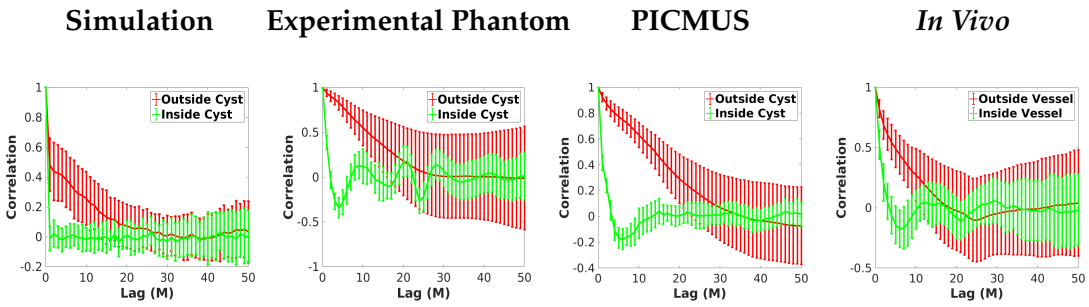


Figure 4.3: Measured spatial coherence within regions of interest (ROIs) inside and outside anechoic or hypoechoic targets. The lines show the means and the error bars show $\pm$ one standard deviation of the measured spatial correlation within each ROI. The locations of the ROIs relative the cyst are shown in Figs. 4.4, 4.6, and 4.7 for the simulated, phantom, and PICMUS data, respectively.

We measured the spatial correlation for a pair of rectangular windows (one in the background, and the other within the target), resulting in the correlation curves shown in Fig. 4.3. The lines correspond to the mean value measured within each ROI, while the errorbars display $\pm$ one standard deviation of the measured correlation within each ROI.

The experimental correlation curves generally agree with our expectations. One notable difference between the simulated and experimental coherence curves is the significant decrease in coherence at lag 1 in simulation, which occurs because of the presence of noise in the simulation (Pinton, Trahey, and Dahl, 2014; Bottenus and Trahey, 2015). We additionally note that the standard deviations (represented by the amplitude of the error bars) appear to increase as we increase lag both inside and outside anechoic regions. This increase is generally greater outside rather than inside the anechoic region with the exception of the simulation result. Fig. 4.3 provides evidence that noise and outliers increase as lag increases, which is one primary motivation

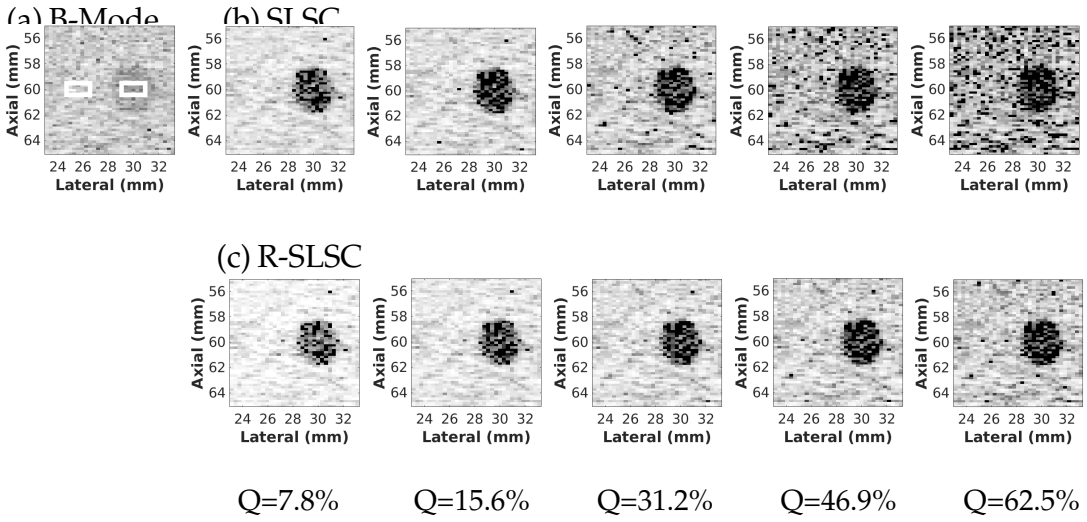


Figure 4.4: (a) DAS B-mode image of an anechoic cyst simulated with Field II (Jensen, 1996; Jensen and Svendsen, 1992). The white rectangles show the ROIs used to calculate Contrast, SNR, CNR, and the correlation curves in Fig. 4.3a. (b) SLSC images corresponding to Q-values of 7.8%, 15.6%, 31.2%, 46.9% and 62.5%, respectively. (c) Corresponding R-SLSC images created with the same Q-values. All images are displayed with 60 dB dynamic range.

for pursuing R-SLSC imaging, as we assume that the ground truth for each correlation estimate lies somewhere within the error bars.

4.5.2 Simulation Results

B-mode, SLSC, and R-SLSC images of the simulated anechoic cyst target are displayed in Fig. 4.4. The rectangles in the B-mode image (Fig. 4.4a) correspond to the regions inside and outside the cyst used to calculate contrast, SNR and CNR, and they were maintained for all performance metrics calculated for this phantom. Fig. 4.4b shows the SLSC beamformed outputs corresponding to Q-values of 7.8%, 15.6%, 31.2%, 46.9 % and 62.5%, respectively, while Fig. 4.4c shows the R-SLSC beamformed outputs for the same

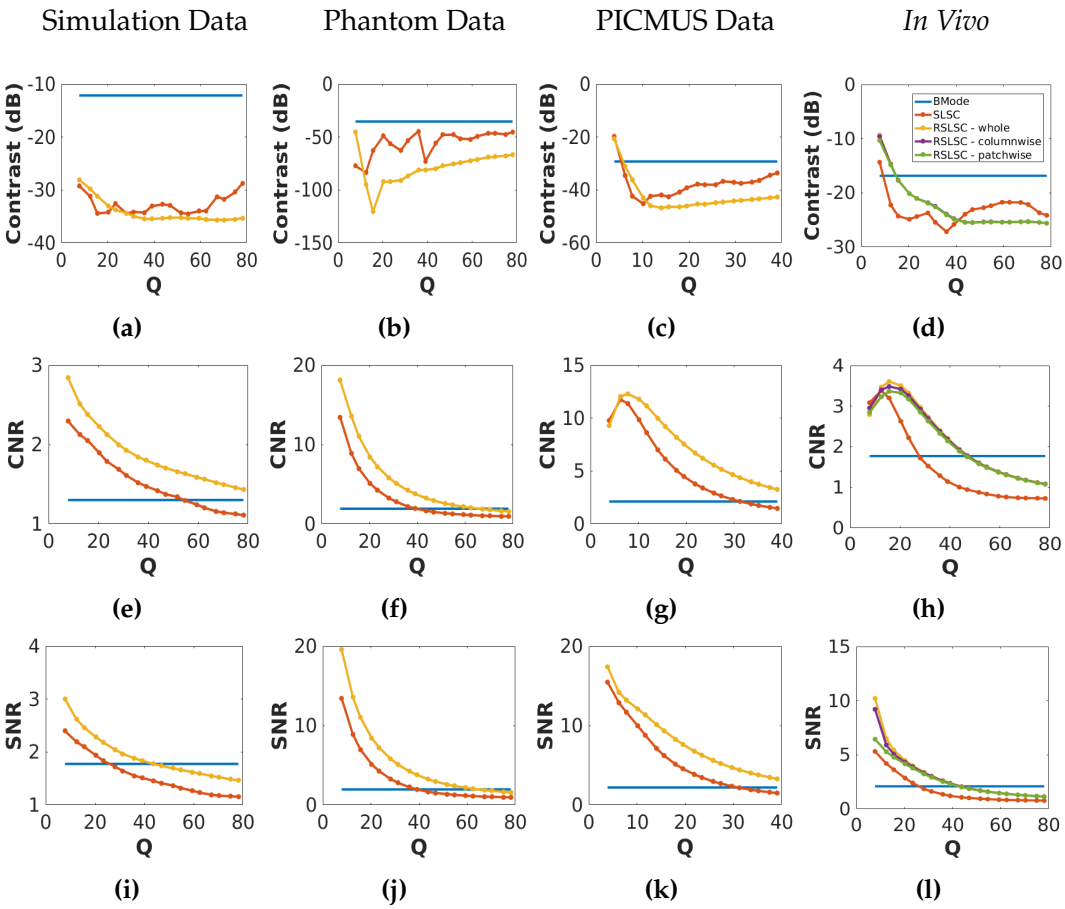


Figure 4.5: Comparison of B-mode, SLSC, and R-SLSC Contrast, CNR and SNR measurements and their variation with Q, as measured in (a, e, i) simulated data with -10dB channel noise, (b , f, j) experimental phantom data acquired with focused transmit beams, (c, g, k) experimental phantom data acquired with plane wave transmission, and (d, h, l) *in vivo* liver data. For the *in vivo* liver data, the patchwise and columnwise results overlap the results obtained with R-SLSC applied to the whole image in most cases. B-mode images were created with the entire receive aperture, and the Q values do not apply to the B-mode results.

Q-values. All images are displayed with a 60 dB dynamic range.

The mean gain in R-SLSC contrast (for all Q values considered) is 1.48 dB, when compared to that of SLSC, which corresponds to a mean gain of 4.53%. The mean gains in R-SLSC SNR and CNR (when compared to SLSC SNR and

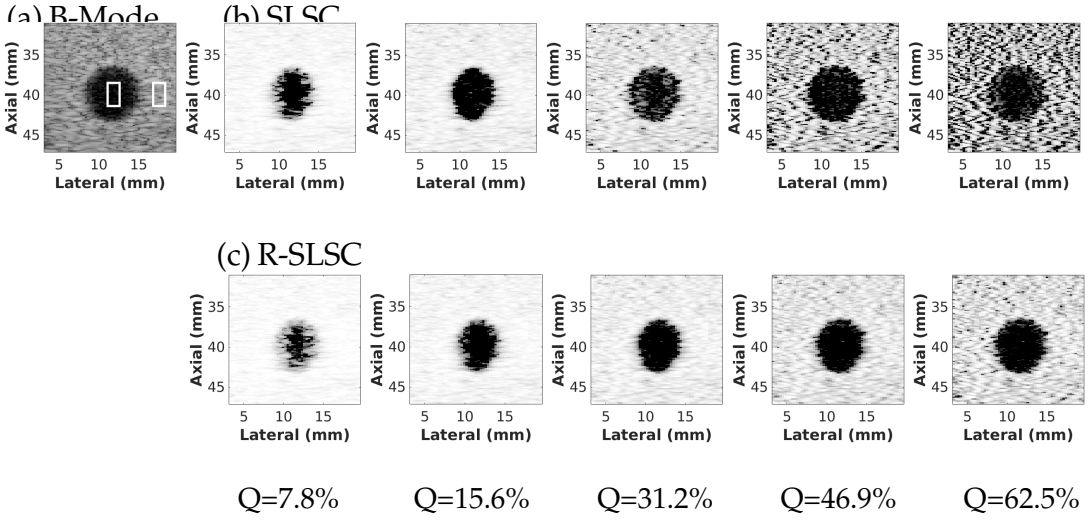


Figure 4.6: (a) DAS B-mode image of an anechoic cyst in a CIRS 054GS experimental phantom. The white rectangles show the ROIs used to calculate Contrast, SNR, CNR, and the correlation curves in Fig. 4.3b. (b) SLSC images corresponding to Q-values of 7.8%, 15.6%, 31.2%, 46.9% and 62.5%, respectively. (c) Corresponding R-SLSC images created with the same Q-values. All images are displayed with 60 dB dynamic range.

CNR) are 0.35 and 0.35, respectively, which correspond to improvements of 22.72% and 22.87%. The contrast and CNR of SLSC and R-SLSC generally outperform DAS B-Mode in this simulation result, as shown in Fig. 4.5 (left), particularly at the higher lag values.

4.5.3 Experimental Phantom Results

A B-mode image of the anechoic cyst phantom target is displayed in Fig. 4.6a with white rectangles that demarcate the regions inside and outside the cyst being considered when evaluating contrast, SNR and CNR. The same ROIs are used for all performance metrics calculated with this phantom. SLSC and R-SLSC images of this phantom are displayed in Fig. 4.6b and 4.6c, respectively (created with Q-values equal to 7.8%, 15.6%, 31.2%, 46.9 % and 62.5 %).

The mean gain in R-SLSC contrast (for all Q-values considered) is 23.91 dB when compared to that of SLSC, which corresponds to a mean gain of 43.18%. The mean gains in R-SLSC SNR and CNR (when compared to SLSC SNR and CNR) are 2.10 and 2.03, respectively, which correspond to improvements of 65.30% and 63.16%. R-SLSC contrast, CNR, and SNR generally outperform B-Mode imaging for the majority of Q-values considered, as shown in the second column of Fig. 4.5.

Qualitatively, for this phantom data, we observe that at the lower lags, boundary delineation for R-SLSC is worse than that of SLSC, likely because R-SLSC does not have sufficient data to estimate a suitable subspace. However, this boundary delineation is improved at higher lags when compared to lower-lag R-SLSC images and when compared to comparable-lag SLSC images. We additionally observe that at lower lags the poor boundary definition results in seemingly smaller cyst sizes. This is related to the finite width of the ultrasound beam and the lower lags containing only local information, which is insufficient to produce a good boundary estimate. However, at higher lags, the cyst size returns closer to its original size because the algorithm incorporates the higher resolution information that is contained within the higher element separations. The tissue texture surrounding the cyst also appears smoother at the higher-lag R-SLSC images when compared to the higher-lag SLSC images.

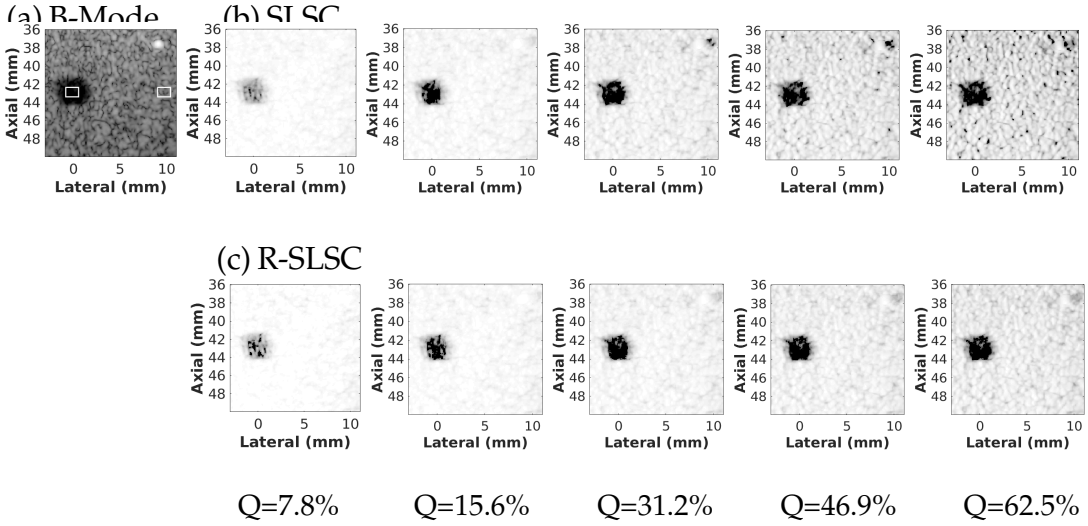


Figure 4.7: (a) DAS B-mode image constructed from from the PICMUS (Liebgott et al., 2016) experimental data of an anechoic target in a CIRS 040GSE phantom. The white rectangles show the ROIs used to calculate Contrast, SNR, CNR, and the correlation curves in Fig. 4.3c. (b) SLSC images corresponding to Q-values of 7.8%, 15.6%, 31.2%, 46.9% and 62.5%, respectively. (c) Corresponding R-SLSC images created with the same Q-values. All images are displayed with 60 dB dynamic range.

4.5.4 Application to Plane Wave Imaging

B-mode, SLSC and R-SLSC images of the plane wave data are displayed in Fig. 4.7. The rectangles in the DAS image (Fig. 4.7a) correspond to the target and background ROIs used to evaluate contrast, SNR and CNR and they are maintained for this phantom. Fig. 4.7b shows SLSC images corresponding to Q-values of 7.8%, 15.6%, 23.4%, 31.2% and 39.0%, while Fig. 4.7c shows corresponding R-SLSC images.

Based on the metrics shown in Fig. 4.5 for the PICMUS data, R-SLSC has a mean contrast gain (averaged over all Q-values considered) of 4.62 dB (12.28%) when compared to SLSC, with gains in SNR and CNR of 2.37 (42.41%) and 2.14 (41.50%), respectively. Similar to the previous phantom results achieved

with focused transmits, R-SLSC imaging outperforms B-Mode imaging for this PICMUS data obtained with plane wave transmits, particularly at higher lags, as evident in Figs. 4.5c, 4.5g, and 4.5k.

We were unable to obtain meaningful results when directly implementing the contrast evaluation script provided by PICMUS organizers because the zero-value pixels in R-SLSC images returned $-\infty$ values after applying the log operation step provided in the script. We therefore made one change to the evaluation script and measured performance prior to log compression, resulting in a contrast of 7.90 dB for the DAS B-Mode image and a mean contrast (averaged over all Q-values considered) of 11.95 dB for the R-SLSC images, which confirms our observations that R-SLSC imaging produces better anechoic cyst contrast (4.05 dB greater) than B-mode imaging.

We additionally note that the hyperechoic point target, which is clearly observable in the DAS B-mode image, is difficult to visualize in both the SLSC and R-SLSC images. Generally, SLSC is known to perform poorly with point target visualization (Lediju et al., 2011) (except in the presence of noise (Bell, Dahl, and Trahey, 2015)). We see that this is also true for R-SLSC imaging with plane wave transmissions. There are also a few coherence outliers within the cyst that are not removed with R-SLSC imaging, although the corresponding location of these outliers have lower amplitudes and are less pronounced in the B-mode image.

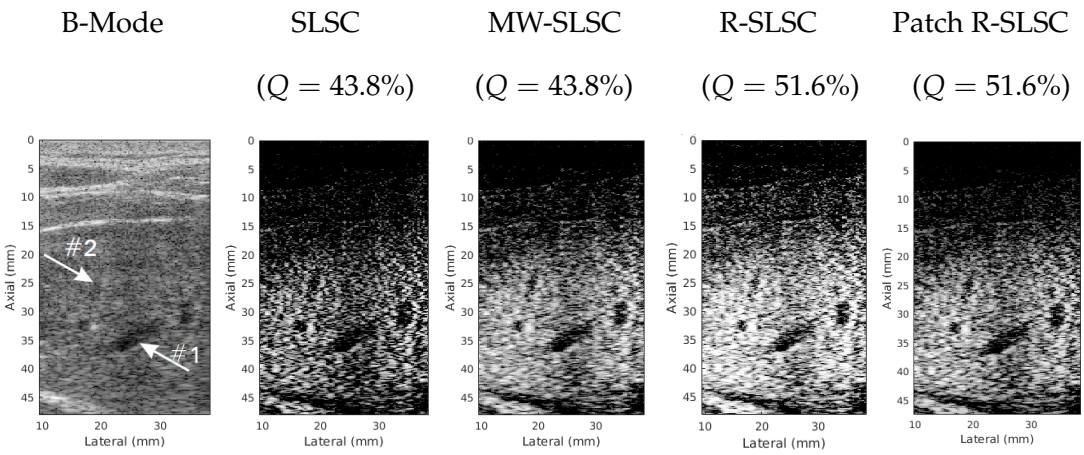


Figure 4.8: *In Vivo* images of hypoechoic blood vessels in a healthy liver. (a) B-mode image, (b) traditional SLSC image created with $Q = 43.8\%$, (c) M-weighted SLSC image (without RPCA), (d) whole-image R-SLSC created with $Q = 51.6\%$ and $\lambda = 0.6$, (e) Patchwise R-SLSC image created with $Q = 51.6\%$ and $\lambda = 0.6$. The dynamic range for each image was chosen to best visualize the data (i.e, 60 dB for the B-mode image and 30 dB for the SLSC, M-weighted SLSC, and R-SLSC images). Arrow #1 points to the ROI used to calculate contrast, CNR, and SNR, while arrow #2 points to a vessel that is noticeably improved with SLSC, M-weighting, and R-SLSC.

4.5.5 In Vivo Liver Data

B-mode, SLSC, and R-SLSC images of a hypoechoic vessel target in an *in vivo* liver are shown in Figs. 4.8a, 4.8b, and 4.8d, respectively. Although rectangles corresponding to the ROIs used to evaluate contrast, SNR and CNR were omitted to improve vessel visibility, they correspond to the largest vessel at a the transmit focal depth of 35mm, located between lateral positions 20 and 30mm (see arrow #1). We also note that the top of these *in vivo* SLSC and R-SLSC images are dark because they are outside of the focal zone.

The mean R-SLSC contrast loss (averaged over all Q-values shown in the last column of Fig. 4.5) is 0.48 dB when compared to that of SLSC, which

corresponds to a 2% decrease. When we exclude the lower lags from this comparison and only consider the higher lags ranging from $Q = 43.75\%$ to $Q = 78.12\%$ (where we see the most contrast improvement), we achieve a higher mean contrast gain of 2.69dB (11.86%) for R-SLSC images compared to SLSC images. The mean SNR and CNR gains (averaged over all Q values) are 1.26 and 0.67, respectively, corresponding to improvements of 71.62% and 45.26%. Similar to phantom data, R-SLSC imaging outperforms B-Mode imaging for this *in vivo* case, as shown in Figs. 4.5d, 4.5h, and 4.5l. The additional lines seen in this last column of Fig. 4.5 are explained in Section 4.5.6.

Qualitatively, there are several additional aspects of these R-SLSC *in vivo* images that are improved over SLSC and B-mode images. For example, clutter obscures the appearance of the vessel located from depth 20 mm to 30 mm in the B-mode image (see arrow #2), but this vessel is more clearly visualized in the SLSC and R-SLSC images. The tissue within the transmit focal zone is additionally brighter overall in R-SLSC images (when compared to SLSC images created with similar lag values). Similar to the phantom and simulated data, the tissue texture also appears to be smoother with R-SLSC images. This smoothing of tissue texture helps with discerning the hypoechoic vessels from their surroundings and reduces the speckle-like texture of the images.

4.5.6 Parallelization

After calculating delays and computing a SLSC image, the average additional computation time required to calculate the robust principal components

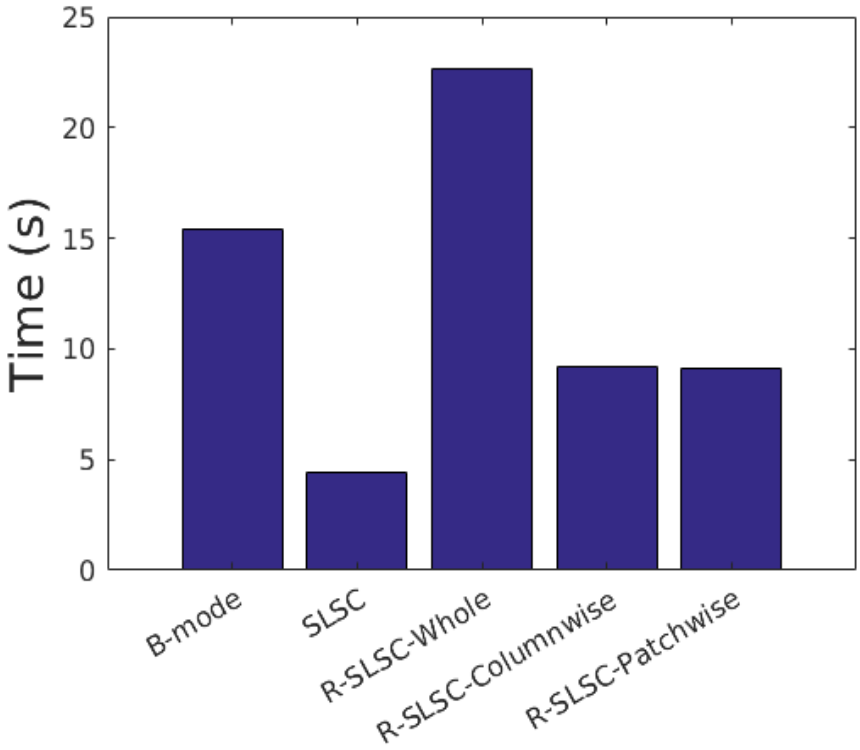


Figure 4.9: Calculation times to obtain B-mode and SLSC images with the computer described in Section 4.4.2, compared to calculation times for the RPCA step required to obtain R-SLSC images with and without patchwise and columnwise parallelization. The calculation time for R-SLSC is reduced by a factor of 2.6 with parallelization.

is 23 seconds per R-SLSC image (using the computer described in Section 4.4.2). One approach to reduce the R-SLSC image computation time is to subdivide the RPCA computation for parallel processing as illustrated in Fig. 4.1b. We successfully implemented this alternative using the same number of columns as scanlines (i.e., 128 columns) for the columnwise implementation and using 64 pixel x 64 pixel patches (i.e. 88 patches total each of size 19.2mm (lateral) $\times$ 1.23mm (axial)) for the patchwise implementation, thereby reducing our RPCA computation times to 9s each. For comparison, Fig. 4.9 shows the calculation times for these various R-SLSC implementations

alongside the calculation times for SLSC correlation calculations and B-mode imaging obtained with the computer described in Section 4.4.2.

A patchwise R-SLSC image of the *in vivo* liver is shown in Fig. 4.8e. When comparing the process for creating this image with that of the corresponding R-SLSC image obtained without parallelization (Fig. 4.8d), we note that this patchwise image excludes the black region at the top of the image when imaging the vessels closer to the image focus. This exclusion results in slightly less clutter inside vessel # 1 which is close to the focus, although the performance metrics in Fig. 4.5 are not affected. In addition, the patchwise image slightly reduces the overall image brightness (when compared to the R-SLSC image without parallelization) because this image is based on the local estimates within each patch. Otherwise, the reduction in computation times achieved with parallelization has minimal impact on image quality. This observation is particularly true at the higher lags, which can be confirmed quantitatively by noting that the two additional lines in Figs. 4.5d, 4.5h, and 4.5l (representing the columnwise and patchwise implementations) overlap the whole-image R-SLSC implementation at the higher lags.

4.5.7 Effect of the λ Parameter and M-Weighting

As speckle SNR is an important characteristic of ultrasound images, the $Q-$values of the *in vivo* R-SLSC images in Fig. 4.8 were chosen to closely match the speckle SNR of DAS images. Our specific selections are represented by the open circles in Fig. 4.10a, which shows the results of our investigations to determine the optimal λ parameter for R-SLSC imaging. While the SLSC

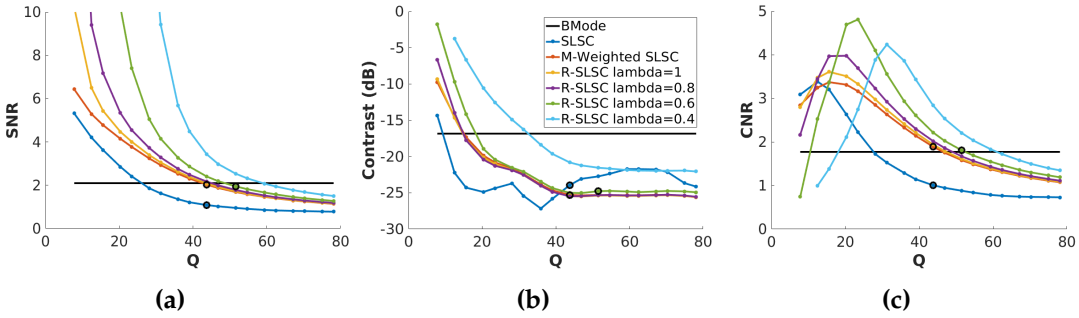


Figure 4.10: (a) SNR, (b) Contrast, and (c) CNR of *in vivo* B-mode, SLSC, M-weighted SLSC, and R-SLSC images. The R-SLSC image metrics are calculated with $\lambda = 1.0, 0.8, 0.6$ and 0.4. Note that R-SLSC images can be tuned to provide similar tissue SNR to B-mode images by adjusting the λ parameter, an option that is not possible with SLSC imaging. The black circles correspond to the lags displayed in Fig. 4.8(b), Fig. 4.8(c) and Fig. 4.8(d). B-mode images were created with the entire receive aperture, and the Q values do not apply to the B-mode results.

images possess high SNR (in most cases higher than B-mode), we find that we can control the SNR more directly in R-SLSC imaging by adjusting the λ parameter.

Fig. 4.10 shows contrast, CNR, and SNR for B-mode, traditional SLSC, and R-SLSC with λ equal to $1.0, 0.8, 0.6$ and 0.4. We observe from Fig. 4.10 that decreasing the λ parameter results in applying less penalty to labeling pixels as outliers, and as a result more coherence values are labeled as outliers to be discarded (which effectively increases the SNR). These changes in SNR generally have minimal impact on image contrast, except when λ=0.4 (see Fig. 4.10b).

When comparing R-SLSC ($\lambda = 1$) to SLSC images created with the linear M-weighting described in Section 4.3.1 (applied without RPCA), we observe that the majority of the improvements obtained with R-SLSC are primarily due to this weighting step. For example, an M-weighted SLSC image without

the application of RPCA is shown in Fig. 4.8c, and it looks strikingly similar to the R-SLSC image achieved with the same $Q-$value (43.8%) and $\lambda = 1$, which is confirmed quantitatively in Fig. 4.10b, as M-weighted SLSC images obtained with different Q-values have similar contrast to R-SLSC ($\lambda = 1$) images. The SNR and CNR of these two image types are also similar at higher lag values (Figs. 4.10a and 4.10c). This observation is true not only for the *in vivo* data, but also for the phantom and simulated data (although images are not shown without RPCA applied for these data). Thus, M-weighting is a major step towards improving SLSC image quality and incorporating the information from higher lags.

Despite this similarity between M-weighted SLSC images and R-SLSC images achieved with $\lambda = 1$ (and the significantly reduced processing time required for M-weighted SLSC compared to R-SLSC imaging), R-SLSC imaging can potentially be considered more advantageous because we can use RPCA to incorporate up to 8% more lags (i.e. 43.8% vs. 51.6%, which corresponds to 10 additional element separations for a 128-element aperture) and achieve similar SNR to B-mode images by decreasing the λ parameter, as shown quantitatively in Fig. 4.10 with an example image displayed in Fig. 4.8d. Although the number of coherence outliers are greater at higher lags, it appears that more of them are rejected with lower values of λ. This data-dependent adjustment of the λ parameter effectively allows us to utilize more lags, achieve similar speckle SNR to B-mode images, and obtain greater improvements in contrast and CNR when compared to traditional SLSC images achieved with the same Q-values.

4.6 Discussion

There are four key contributions of this work. First, we applied both linear M-weighting and RPCA to the traditional SLSC imaging method in order to incorporate previously discarded information from higher lags. With M-weighting, it appears that the short lags provide more structural information (i.e., general cyst location) while the longer lags provide more boundary information, and both contributions work together to improve image quality for anechoic and hypoechoic targets after incorporating more lags with more weight applied to the short lag region. Additional weighting schemes could be applied in the future to explore the optimal weights for a range of imaging targets and anatomical structures. R-SLSC could be considered as a more advanced weighting scheme that improves image quality by both rejecting coherence outliers and taking advantage of the demonstrated benefits of M-weighting. Our second contribution highlights the data-dependent performance of R-SLSC, which can be tuned to provide similar tissue SNR to B-mode images by adjusting the λ parameter. Third, we showed that the processing times for R-SLSC can be reduced by subdividing the image data. Finally, we demonstrated that R-SLSC imaging outperforms traditional SLSC imaging (defined as improved SNR, CNR, and contrast of anechoic or hypoechoic regions) at higher lags when applied to data acquired with both focused and plane wave transmissions.

When anechoic and hypoechoic targets are barely discernible in B-mode images due to low contrast and clutter, we expect SLSC and R-SLSC to clearly distinguish these targets from their surroundings, particularly in high-noise

environments as represented by the simulation results in Fig. 4.4 and the *in vivo* results in Fig. 4.8. R-SLSC experiences additional improvements over SLSC as lag increases in all example cases shown in this work (simulation, phantom, and *in vivo*), as demonstrated in Fig. 4.5. This improvement at higher lags is caused by a combination of applying both linear M-weighting and the RPCA algorithm, which develops a better subspace estimate as the amount of data available to the algorithm increases. Therefore, rejection of the noise and outliers is more prevalent at the higher lags, leading to an image with smoother tissue texture. This smoothing of tissue texture helps to discern anechoic and hypoechoic structures from their surroundings and reduces the speckle-like texture of the images, which is generally beneficial for boundary detection (e.g., similar to spatial compounding (Trahey, Smith, and Von Ramm, 1986; Entrekin et al., 1999)), but could potentially limit the diagnostic information typically provided by the presence of speckle. We can potentially recover some of this diagnostic value by adjusting the λ parameter, which we envision being controlled by an additional knob on an ultrasound scanner, similar to existing options like focal depth or time gain compensation that are currently used to enhance ultrasound image quality. These results imply that both R-SLSC and M-weighting will perform well in high-noise clinical scenarios where anechoic or hypoechoic target visualization is critical. Possible clinical applications include breast cyst visualization (Stavros, 2004), liver vessel tracking (De Luca et al., 2015), and obese patient imaging.

One common characteristic between SLSC and R-SLSC images is heightened sensitivity to structural boundaries. For example, when low-amplitude

signals are surrounded by hyperechoic structures with high-amplitude signals and high spatial coherence, the coherence of the lower amplitude signal is reduced relative to that of the higher amplitude signal. While this characteristic is a major strength when detecting cyst-like structures, it is also a limitation when imaging hyperechoic boundaries next to tissue structures. This observation was evident in *in vivo* cardiac images (Bell et al., 2013a), and it is present at the distal liver boundary in Fig. 4.8, where this boundary appears to be separated from the rest of the liver tissue in SLSC and R-SLSC images.

While the processing times for R-SLSC could be considered as an additional limitation of R-SLSC imaging, Fig. 4.9 demonstrates that it is feasible to subdivide the RPCA step to implement parallel processing for real-time imaging. This alteration provides sufficient information to locally estimate a suitable subspace while rejecting appropriate coherence outliers.

When comparing the SLSC contrast curves for simulated and experimental data in Fig. 4.5 to the corresponding coherence curves inside the cyst (Fig. 4.3), the shapes of these curves are similar as a function of Q. While changes in the contrast of SLSC images seems to be correlated with changes in the corresponding coherence curves as a function of Q, the contrast of the R-SLSC images is more stable at higher lags as a result of robustness to coherence outliers. This observation further supports the implementation of R-SLSC imaging.

4.7 Conclusion

This work is the first to re-examine the lag summation step of the SLSC algorithm and achieve additional robustness to coherence outliers through both weighted summation of individual coherence images (i.e., M-weighting) and the application of RPCA. The original SLSC imaging algorithm does not consider the content of the images formed at different lags before summing them, and thus does not exploit tissue texture differences in SLSC images created with various short lag values. In addition, the traditional SLSC beamforming method is somewhat restricted to short lag values when considering the widely varying coherence values present at the longer lags. Our methods improve the original SLSC imaging method by incorporating a linearly decaying weighting scheme to achieve M-weighted SLSC images. RPCA is additionally utilized to search for a low dimensional subspace to the coherence images at different lags. The RPCA projections and consequent denoising of the individual images on this low dimensional subspace are then used to achieve R-SLSC images. Both M-weighted SLSC and R-SLSC imaging enable the use of higher lag information, offer increased contrast, SNR and CNR, and are generally more robust to noise (defined as coherence outliers) when compared to traditional SLSC imaging.

References

Lediju, Muyinatu A, Gregg E Trahey, Brett C Byram, and Jeremy J Dahl (2011). "Short-lag spatial coherence of backscattered echoes: Imaging characteristics". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 58.7, pp. 1377–1388.

Candès, Emmanuel J, Xiaodong Li, Yi Ma, and John Wright (2011). "Robust principal component analysis?" In: *Journal of the ACM (JACM)* 58.3, pp. 1–37.

Nair, Arun Asokan, Trac Duy Tran, and Muyinatu A Lediju Bell (2017). "Robust short-lag spatial coherence imaging". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 65.3, pp. 366–377.

Cittert, Pieter Hendrik van (1934). "Die wahrscheinliche Schwingungsverteilung in einer von einer Lichtquelle direkt oder mittels einer Linse beleuchteten Ebene". In: *Physica* 1.1-6, pp. 201–210.

Zernike, Frederik (1938). "The concept of degree of coherence and its application to optical problems". In: *Physica* 5.8, pp. 785–795.

Goodman, Joseph W (2015). *Statistical optics*. John Wiley & Sons.

Mallart, Raoul and Mathias Fink (1991). "The van Cittert–Zernike theorem in pulse echo measurements". In: *The Journal of the Acoustical Society of America* 90.5, pp. 2718–2727.

Liu, Dong-Lai and Robert C Waag (1995). "About the application of the van Cittert-Zernike theorem in ultrasonic imaging". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 42.4, pp. 590–601.

Bamber, Jeffrey C, Ronald A Mucci, and Donald P Orofino (2002). "Spatial coherence and beamformer gain". In: *Acoustical Imaging*. Springer, pp. 43–48.

Jakovljevic, Marko, Gregg E Trahey, Rendon C Nelson, and Jeremy J Dahl (2013). "In vivo application of short-lag spatial coherence imaging in human liver". In: *Ultrasound in medicine & biology* 39.3, pp. 534–542.

Bell, Muyinatu A Lediju, Robi Goswami, Joseph A Kisslo, Jeremy J Dahl, and Gregg E Trahey (2013a). "Short-lag spatial coherence imaging of cardiac ultrasound data: Initial clinical results". In: *Ultrasound in medicine & biology* 39.10, pp. 1861–1874.

Kakkad, Vaibhav, Jeremy Dahl, Sarah Ellestad, and Gregg Trahey (2013). "In vivo performance evaluation of short-lag spatial coherence and harmonic spatial coherence imaging in fetal ultrasound". In: *2013 IEEE International Ultrasonics Symposium (IUS)*. IEEE, pp. 600–603.

Bell, Muyinatu A Lediju, Jeremy J Dahl, and Gregg E Trahey (2015). "Resolution and brightness characteristics of short-lag spatial coherence (SLSC) images". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 62.7, pp. 1265–1276.

Hyun, Dongwoon, Gregg E Trahey, Marko Jakovljevic, and Jeremy J Dahl (2014). "Short-lag spatial coherence imaging on matrix arrays, Part 1: Beamforming methods and simulation studies". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 61.7, pp. 1101–1112.

Jakovljevic, Marko, Brett C Byram, Dongwoon Hyun, Jeremy J Dahl, and Gregg E Trahey (2014). "Short-lag spatial coherence imaging on matrix arrays, Part II: Phantom and in vivo experiments". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 61.7, pp. 1113–1122.

Bell, Muyinatu A Lediju, Nathanael Kuo, Danny Y Song, and Emad M Boctor (2013b). "Short-lag spatial coherence beamforming of photoacoustic images for enhanced visualization of prostate brachytherapy seeds". In: *Biomedical optics express* 4.10, pp. 1964–1977.

Bell, Muyinatu A Lediju, Xiaoyu Guo, Hyun Jae Kang, and Emad Boctor (2014). "Improved contrast in laser-diode-based photoacoustic images with short-lag spatial coherence beamforming". In: *2014 IEEE International Ultrasonics Symposium*. IEEE, pp. 37–40.

Gandhi, Neeraj, Margaret Allard, Sungmin Kim, Peter Kazanzides, and Muyinatu A Lediju Bell (2017). "Photoacoustic-based approach to surgical guidance performed with and without a da Vinci robot". In: *Journal of Biomedical Optics* 22.12, p. 121606.

Alles, Erwin J, Michael Jaeger, and Jeffrey C Bamber (2014). "Photoacoustic clutter reduction using short-lag spatial coherence weighted imaging". In: *2014 IEEE International Ultrasonics Symposium*. IEEE, pp. 41–44.

Pourebrahimi, Behnaz, Sangpil Yoon, Dustin Dopsa, and Michael C Kolios (2013). "Improving the quality of photoacoustic images using the short-lag spatial coherence imaging technique". In: *Photons Plus Ultrasound: Imaging*

and Sensing 2013. Vol. 8581. International Society for Optics and Photonics, 85813Y.

Jolliffe, Ian T (1986). "Principal components in regression analysis". In: *Principal component analysis*. Springer, pp. 129–155.

Han, Jiawei, Micheline Kamber, and Jian Pei (2011). "Data mining concepts and techniques third edition". In: *The Morgan Kaufmann Series in Data Management Systems* 5.4, pp. 83–124.

Turk, Matthew and Alex Pentland (1991). "Eigenfaces for recognition". In: *Journal of cognitive neuroscience* 3.1, pp. 71–86.

Moore, Bruce (1981). "Principal component analysis in linear systems: Controllability, observability, and model reduction". In: *IEEE transactions on automatic control* 26.1, pp. 17–32.

Bishop, Christopher M (2006). *Pattern recognition and machine learning*. springer.

Mauldin, F William, Francesco Viola, and William F Walker (2010). "Complex principal components for robust motion estimation". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 57.11, pp. 2437–2449.

Prytherch, DR, DH Evans, MJ Smith, and DS Macpherson (1982). "On-line classification of arterial stenosis severity using principal component analysis applied to Doppler ultrasound signals". In: *Clinical physics and physiological measurement* 3.3, p. 191.

Wright, John, Arvind Ganesh, Shankar Rao, and Yi Ma (2009). "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization". In: *Coordinated Science Laboratory Report no. UILU-ENG-09-2210, DC-243*.

Lin, Zhouchen, Minming Chen, and Yi Ma (2010). "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices". In: *arXiv preprint arXiv:1009.5055*.

Lin, Zhouchen, Arvind Ganesh, John Wright, Leqin Wu, Minming Chen, and Yi Ma (2009). "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix". In: *Coordinated Science Laboratory Report no. UILU-ENG-09-2214, DC-246*.

Mauldin Jr, F William, Hongtu T Zhu, Russell H Behler, Timothy C Nichols, and Caterina M Gallippi (2008). "Robust principal component analysis and clustering methods for automated classification of tissue response to ARFI excitation". In: *Ultrasound in medicine & biology* 34.2, pp. 309–325.

Lediju, Muyinatu A, Michael J Pihl, Stephen J Hsu, Jeremy J Dahl, Caterina M Gallippi, and Gregg E Trahey (2009). "A motion-based approach to abdominal clutter reduction". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 56.11, pp. 2437–2449.

Dahl, Jeremy J, Dongwoon Hyun, Muyinatu Lediju, and Gregg E Trahey (2011). "Lesion detectability in diagnostic ultrasound with short-lag spatial coherence imaging". In: *Ultrasonic imaging* 33.2, pp. 119–133.

https://people.eecs.berkeley.edu/ yima/matrix-rank/sample_code.html. https://people.eecs.berkeley.edu/~yima/matrix-rank/sample_code.html.

Bell, Muyinatu Adebisi Lediju (2012). "Improved endocardial border definition with short-lag spatial coherence (SLSC) imaging". PhD thesis. Duke University.

Hyun, Dongwoon, Gregg E Trahey, and Jeremy Dahl (2013). "A GPU-based real-time spatial coherence imaging system". In: *Medical Imaging 2013: Ultrasonic Imaging, Tomography, and Therapy*. Vol. 8675. International Society for Optics and Photonics, 86751B.

Hyun, Dongwoon, Gregg E Trahey, and Jeremy J Dahl (2015). "Real-time high-framerate in vivo cardiac SLSC imaging with a GPU-based beamformer". In: *2015 IEEE International Ultrasonics Symposium (IUS)*. IEEE, pp. 1–4.

Holmes, Michael, Alexander Gray, and Charles Isbell (2007). "Fast SVD for large-scale matrices". In: *Workshop on Efficient Machine Learning at NIPS*. Vol. 58, pp. 249–252.

Jensen, Jørgen Arendt (1996). "Field: A program for simulating ultrasound systems". In: *10TH NORDICBALTIC CONFERENCE ON BIOMEDICAL IMAGING, VOL. 4, SUPPLEMENT 1, PART 1: 351–353*. Citeseer.

Jensen, Jørgen Arendt and Niels Bruun Svendsen (1992). "Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 39.2, pp. 262–267.

Liebgott, Herve, A Rodriguez-Molares, F Cervenansky, Jørgen Arendt Jensen, and Olivier Bernard (2016). "Plane-wave imaging challenge in medical ultrasound". In: *2016 IEEE International ultrasonics symposium (IUS)*. IEEE, pp. 1–4.

Pinton, Gianmarco F, Gregg E Trahey, and Jeremy J Dahl (2014). "Spatial coherence in human tissue: Implications for imaging and measurement". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 61.12, pp. 1976–1987.

Bottenus, Nick B and Gregg E Trahey (2015). "Equivalence of time and aperture domain additive noise in ultrasound coherence". In: *The Journal of the Acoustical Society of America* 137.1, pp. 132–138.

Trahey, Gregg E, Stephen W Smith, and OT Von Ramm (1986). "Speckle pattern correlation with lateral aperture translation: Experimental results and implications for spatial compounding". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 33.3, pp. 257–264.

Entrekin, R, P Jackson, JR Jago, and BA Porter (1999). "Real time spatial compound imaging in breast ultrasound: technology and early clinical experience". In: *medicamundi* 43.3, pp. 35–43.

Stavros, A Thomas (2004). *Breast ultrasound*. Lippincott Williams & Wilkins.

De Luca, V, T Benz, S Kondo, L König, D Lübke, S Rothlübbers, O Somphone, S Allaire, MA Lediju Bell, DYF Chung, et al. (2015). "The 2014 liver ultrasound tracking benchmark". In: *Physics in Medicine & Biology* 60.14, p. 5571.

Chapter 5

Deep Learning for Simultaneous Ultrasound Image Formation and Segmentation

In this chapter, we demonstrate how modern deep learning (Goodfellow et al., 2016) techniques can improve the information extraction pipeline in ultrasound imaging. Specifically, we work with the challenging scenario of single plane wave ultrasound imaging (Montaldo et al., 2009). Single plane wave transmissions are promising for automated imaging tasks requiring high ultrasound frame rates over an extended field of view. However, a single plane wave insonification typically produces sub-optimal image quality. To address this limitation, we explore the use of deep neural networks (DNNs) as an alternative to traditional beamforming. The objectives of this work are to obtain information directly from raw channel data and to simultaneously generate both a segmentation map for automated ultrasound tasks and a corresponding ultrasound B-mode image for interpretable supervision of the automation. We focus on visualizing and segmenting anechoic targets

surrounded by tissue and ignoring or de-emphasizing less important surrounding structures. DNNs trained with Field II simulations were tested with simulated, experimental phantom, and *in vivo* datasets that were not included during training. With unfocused input channel data (i.e., prior to the application of receive time delays), simulated, experimental phantom, and *in vivo* test datasets achieved mean $\pm$ standard deviation Dice similarity coefficients of 0.92 ± 0.13, 0.92 ± 0.03, and 0.77 ± 0.07, respectively, and generalized contrast-to-noise ratios (gCNR) of 0.95 ± 0.08, 0.93 ± 0.08, and 0.75 ± 0.14, respectively. With subaperture beamformed channel data and a modification to the input layer of the DNN architecture to accept these data, the fidelity of image reconstruction increased (e.g., mean gCNR of multiple acquisitions of two *in vivo* breast cysts ranged 0.89-0.96), but DNN display frame rates were reduced from 395 Hz to 287 Hz. Overall, the DNNs successfully translated feature representations learned from simulated data to phantom and *in vivo* data, which is promising for this novel approach to simultaneous ultrasound image formation and segmentation. The work presented in this chapter was published earlier in Nair et al., 2020.

5.1 Introduction

Ultrasound images are widely used in multiple diagnostic, interventional, and automated procedures that range from cancer detection (Pons et al., 2016; Kumar et al., 2018) to ultrasound-based visual servoing (Mebarki, Krupa, and Chaumette, 2010). Despite this wide clinical utility, there are three pervasive challenges. First, the presence of speckle and clutter often complicates

image interpretation (Entrekin et al., 2001), particularly during automated ultrasound-based tasks. Second, speckle, clutter, and other inherent ultrasound image features tend to confuse simple thresholding and filtering algorithms and require the use of more complex procedures to successfully perform automated segmentations (Xian et al., 2018). Third, segmentation tasks are traditionally implemented after image formation (Noble and Boukerroui, 2006; Xian et al., 2018), which further increases the computational complexity of implementing segmentation algorithms to provide a desired segmentation result. These three challenges have the potential to be addressed by simultaneously outputting multiple desired information in parallel, directly from the raw ultrasound channel data, with the assistance of deep learning.

The field of deep learning has traditionally been applied to diagnostic ultrasound tasks, such as classification, segmentation, and image quality assessment (Liu et al., 2019). Recently, there has been growing interest in applying deep neural networks (DNNs) to augment or replace steps of the ultrasound image formation process. For example, there is a class of deep learning approaches that improves data quality obtained from a single plane wave transmission by enhancing the beamformed data (Perdios et al., 2019; Gasse et al., 2017; Zhang et al., 2018; Zhou et al., 2018). Another class of ultrasound-based deep learning approaches produces high-quality images with reduced data sampling in order to increase frame rates (Perdios et al., 2017; Yoon et al., 2017; Yoon et al., 2018; Yoon and Ye, 2018; Khan, Huh, and Ye, 2019b; Khan, Huh, and Ye, 2019a; Vedula et al., 2018b; Huang et al., 2018). Deep learning has also been used to replace portions of the beamforming

process by learning the parameters of a model created during an intermediary beamforming step (Luchies and Byram, 2017; Luchies and Byram, 2018; Luchies and Byram, 2019; Luijten et al., 2019; Vedula et al., 2018a). However, none of these methods provide an end-to-end transformation that learns information directly from raw channel data.

Prior work from our group (Nair et al., 2018b; Nair et al., 2018a; Nair et al., 2019) introduced DNNs that were trained purely with simulated data to successfully extract information directly from raw radiofrequency (RF) single plane wave channel data, prior to the application of time delays or any other traditional beamforming steps. Similarly, Simson et al., 2018 introduced a method to learn the entire beamforming process without applying delays to the input data. This approach trains on real data rather than simulated data and uses focused transmissions rather than plane wave transmissions. With the exception of Nair et al., 2019, no existing methods simultaneously provide ultrasound images and segmentation information directly from raw channel data.

One challenge with learning information directly from raw channel data is the absence of receive focusing delays. Instead, the DNN input has dimensions of time vs. channels, and the DNN output has dimensions of depth vs. width. Thus, the network architecture must account for the mapping of time (recorded on each channel) to depth, as well as the mapping of multiple channels (which includes temporal recordings) to a single pixel in the image width dimension, and the proposed task is therefore not a simple image-to-image transformation. This challenge is not present in other ultrasound-based deep

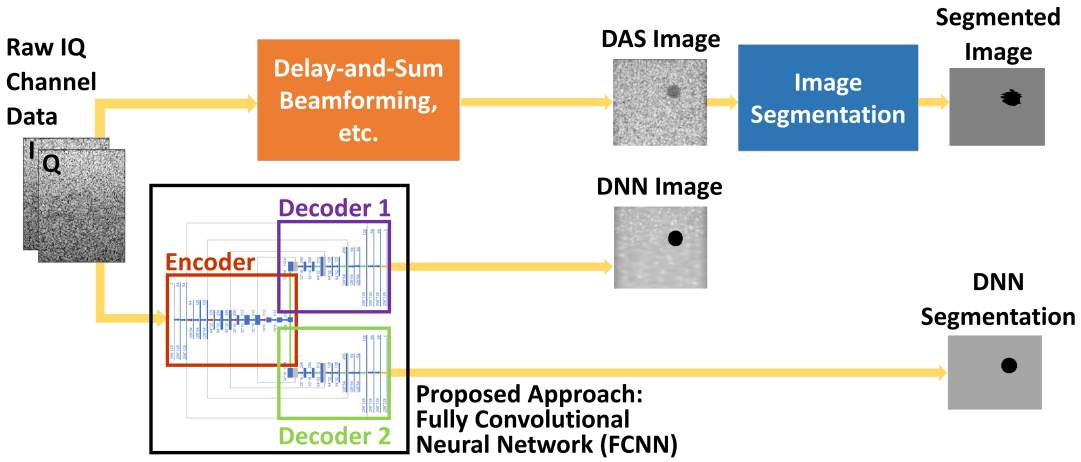


Figure 5.1: Illustration of our proposed DNN goals (bottom) in comparison to the traditional approach (top). Traditionally, raw channel data undergoes delay-and-sum beamforming followed by envelope detection, log compression and filtering to produce an interpretable delay-and-sum (DAS) beamformed image, which is then passed to a segmentation algorithm to isolate a desired segment of the image. We propose to replace this sequential process with a fully convolutional neural network (FCNN) architecture, consisting of a single encoder and two decoders, that simultaneously outputs both a DNN image and a DNN segmentation directly from raw ultrasound channel data received after a single plane wave insonification. The input is in-phase/quadrature (IQ) ultrasound data, presented as a three-dimensional tensor.

learning approaches that learn image-to-image transformations using input and output data that are both represented in the same spatial domain. In addition, our previous work did not take advantage of the lower spatial frequencies available when performing this transformation with raw, complex, baseband, in-phase and quadrature (IQ) data (when compared to the higher spatial frequencies of raw RF ultrasound channel data).

The primary contribution of this work is a description and analysis of a DNN framework that is, to the author's knowledge, the first to replace beamforming followed by segmentation (as illustrated in the top of Fig. 5.1) with parallel B-mode and segmentation results offered as a paired network

output from a single network input of raw IQ data (as illustrated in the bottom of Fig. 5.1). This parallel information may be extracted directly from the recorded echoes received after a single plane wave insonification, either before or after the application of time delays (which can be implemented in hardware), or after receiving channel data from focused transmissions. We compare these three options in this work and show that a simple modification to the input layer of a DNN can be used to accommodate each of these options. These options have the potential to simultaneously benefit both robot-based computer vision tasks (which often discard many of the details in ultrasound B-mode images through post-processing and primarily utilize resulting target segmentation information (Mebarki, Krupa, and Chaumette, 2010; Huang et al., 2019)) and human observers (who may require the more familiar B-mode information to override, supervise, or otherwise interpret the output of automated and image segmentation tasks). Assuming that DNNs can be optimized to be faster than current acquisition rates (Bianco et al., 2018) and provide better than current image quality with single plane wave beamforming, we also provide some guidelines to focus future efforts.

To demonstrate initial proof of principle, we focus on the detection of small, round, anechoic, cyst-like targets. This focus characterizes a range of anatomical targets, including urine-filled renal calyces (which can range from 3 mm to 7 mm in diameter (Cadeddu et al., 1997)), cysts in the breast (which can be as small as 2-3 mm in ultrasound images (Jackson, 1990) with a mean size of 2.0 $\pm$ 1.8 cm (Vargas et al., 2004)), and ovarian follicles (which can range from 10-17 mm in width (Wikland et al., 2001)). We train a task-specific

DNN to target these types of structures and ignore or de-emphasize structures that are not anechoic (considering that this information would otherwise be ignored through image post-processing to achieve the proposed task). One key feature of our training approach is the use of ground truth segmentation masks to produce enhanced beamformed images in order to enhance identification of anechoic targets during network training. In addition, network training in this work is performed in a purely supervised manner using a fully convolutional neural network (FCNN), making the network easier and faster to train when compared to the generative adversarial network (GAN) employed in our previous paper (Nair et al., 2019).

The remainder of this chapter is organized as follows. Section 5.2 describes our network architecture, training, and evaluation methods. Section 5.3 presents our results. Section 5.4 includes a discussion of key insights from our results, and Section 5.5 summarizes our major conclusions.

5.2 Methods

5.2.1 Problem Formulation for Unfocused Input Channel Data

Let I_d be a tensor that contains downsampled IQ channel data of size $d \times w \times q$, where d is the length of each downsampled IQ signal, w is the IQ data image width, which is set to be equivalent to the number of transducer element receive channels, and q has two channels, each representing the in-phase or quadrature component of the recording. Our goal is to produce one DNN beamformed image, D, and one segmentation map prediction, S_p, each with dimensions $d \times w$, using I_d as input. We employ a FCNN with

trainable parameters θ to learn the optimal mapping of $I_d \rightarrow y$ that produces acceptable images for robotic automation and human supervision, where y is the reference for the optimal mapping. This reference consists of a true segmentation map, S_t, and the corresponding enhanced beamformed image, E. Thus, y describes the tuple (E, S_t).

Our DNN architecture, shown in Fig. 5.2, was designed based on the U-Net (Ronneberger, Fischer, and Brox, 2015) architecture for biomedical image segmentation, possessing a single encoder adopting the VGG-13 (Simonyan and Zisserman, 2014) encoder with batch normalization (BatchNorm) (Ioffe and Szegedy, 2015) layers to stabilize training and speed up convergence. There is one encoder, which takes the input and passes it through a series of ten 3x3 convolutional layers and downsamples in the spatial domain using 2x2 max pooling (MaxPool) layers while simultaneously increasing the number of feature channels in the data. This process is followed by two decoders, each with nine convolutional layers. One decoder produces a DNN image, $D(I_d; \theta)$, while the second decoder produces the DNN segmentation image, $S_p(I_d; \theta)$. The structures of the decoders are identical, each having a similar architecture to the encoder but mirrored, with 2x2 up-convolutional (UpConv) layers performing upsampling in the spatial domain and simultaneously decreasing the number of feature channels in the data. Both decoders have a sigmoid non-linearity in the last layer, ensuring the final predicted DNN image or DNN segmentation is restricted to be between 0 and 1. In addition, skip connections (He et al., 2016) are implemented to copy extracted features from the encoder to the decoder at the same scale (as in Ronneberger, Fischer,

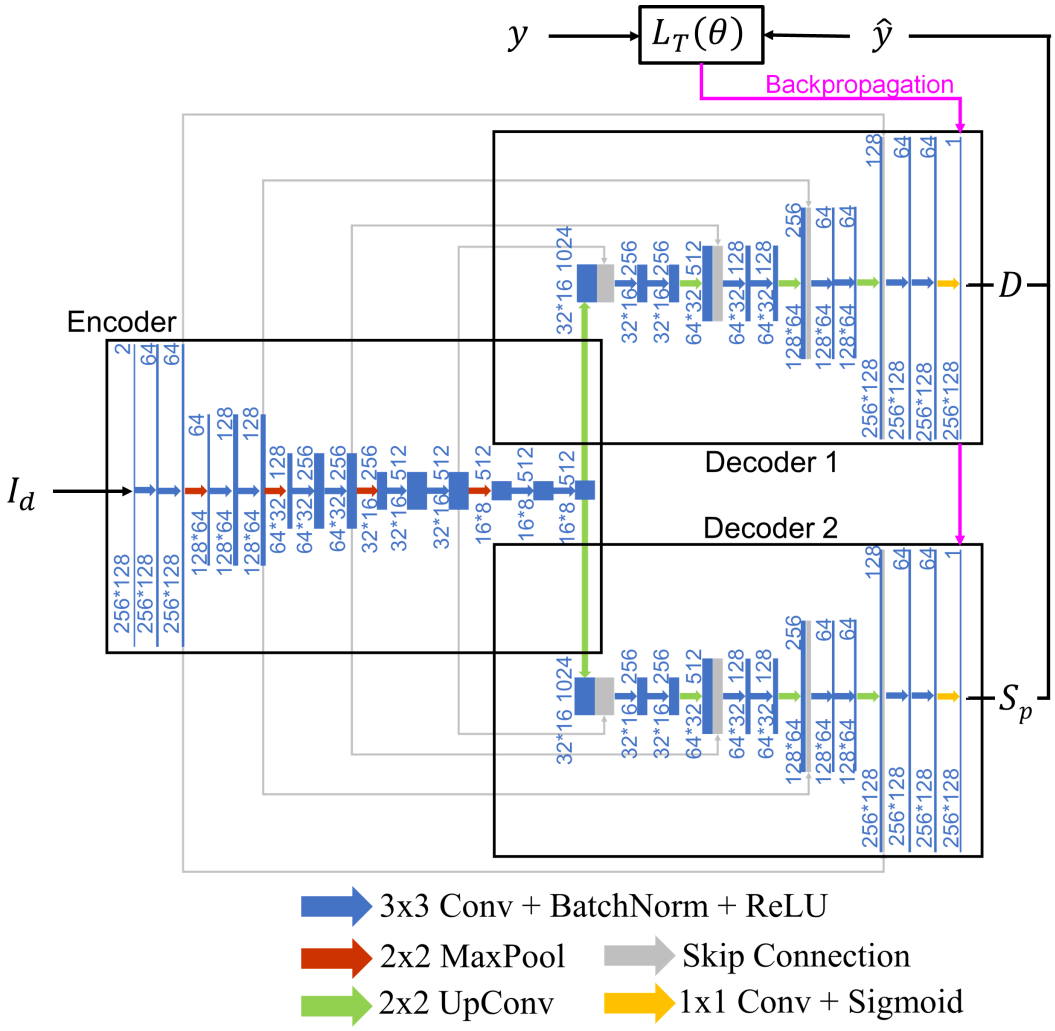


Figure 5.2: FCNN architecture and training scheme for simultaneous DNN image and DNN segmentation generation.

and Brox, 2015). The skip connections enable the network to learn finer details which might otherwise be lost as a result of downsampling, to enhance the flow of information through the network, and to reduce training time and training data requirements (Ronneberger, Fischer, and Brox, 2015; Simonyan and Zisserman, 2014).

5.2.2 Network Architecture

5.2.3 Mapping and Scaling of Network Input and Training Data

In order to consider the time-to-depth mapping described in Section 5.1, each recorded channel data image, I, was downsampled from a grid size of approximately 8,300 pixels x 128 pixels (time samples x receive channel number) to a grid size of 256 pixels x 128 pixels (depth x width) with linear interpolation, satisfying Nyquist criteria and resulting in I_d. To achieve I_d, each axial line in I (i.e., the recorded echo samples) was mapped to a fixed position in space using an input speed of sound value that is either known (for simulated data) or assumed (for experimental data). In general, the reduction of the input data size (e.g., from I to I_d) is necessary to maintain the entire input and corresponding output images, as well as the corresponding gradient information of the DNN, within the GPU memory during training, and to increase training and inference speed. I_d was then normalized by the maximum absolute value to ensure $I_d \in [-1, 1]$, resulting in the network input.

To scale the training data used to obtain the DNN image output, the recorded channel data image I was demodulated to baseband, beamformed, downsampled, filtered to create envelope-detected data, then log-compressed to achieve I_{dB}. The demodulation, beamforming, downsampling, and filtering steps were implemented with the Ultrasound Toolbox (Rodriguez-Molares et al., 2017). I_{dB} was initially displayed on a log scale with a dynamic range of 60 dB (which is a common dynamic range when displaying ultrasound

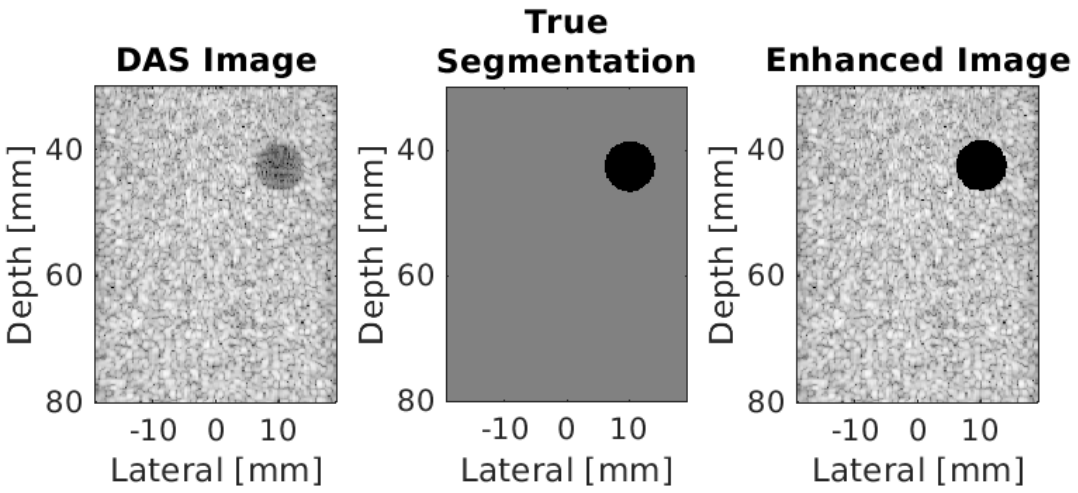


Figure 5.3: From left to right, this example shows a simulated DAS beamformed ultrasound image, I_n, the ground truth segmentation of the cyst from surrounding tissue, S_t, and the corresponding enhanced beamformed image, E.

images). I_{dB} was then rescaled to I_n as follows:

$$I_n = \frac{I_{dB} + 60}{60} \tag{5.1}$$

in order to ensure $I_n \in [0, 1]$. This normalization is an important step for stable DNN training, as neural networks are highly sensitive to data scaling (Ioffe and Szegedy, 2015), and optimal performance is typically achieved when the ranges of the inputs and outputs of the network are normalized.

A final enhancement was applied to I_n to obtain an enhanced B-mode image, E, in efforts to overcome the poor contrast and acoustic clutter limitations of single plane wave transmissions. For example, Fig. 5.3 shows a DAS beamformed image obtained after a single plane wave insonification of an anechoic cyst simulated with Field II (Jensen and Svendsen, 1992; Jensen, 1996), followed by the true segmentation and the enhanced image used during network training only. The rationale for this enhancement is that the cyst is

intrinsically anechoic, but the visualized cyst in the DAS beamformed image contains acoustic clutter (e.g., the sidelobe responses of the scatterers in the surrounding tissue region extending into the anechoic cyst region). Our goal is to ideally obtain better quality images than that of DAS images (and not to simply replicate poor DAS image quality during training). Toward this end, the pixel labels obtained from the input echogenicity map (which is also considered as the true segmentation mask, S_t) were used to set the pixel values of the anechoic regions in I_n to zero while preserving the pixel values of the surrounding tissue, with the intention of removing the clutter observed within the cyst and thereby restoring the desired anechoic appearance of the cyst, as shown in Fig. 5.3. Enhanced beamformed DAS images, E, were only used to train the DNN to learn the mapping function required for estimation of the optimal network parameters θ by minimizing the loss between the reconstructed images $\hat{y}$ and the reference y, where $\hat{y}$ describes the tuple (D, S_p). Note that the procedure described to obtain enhanced images was not applied to alter any of the DNN output images.

5.2.4 Network Training

During training, the total network loss, $L_T(\theta)$, was composed of the weighted sum of two losses. The first loss was the mean absolute error, or L1Loss, between the predicted DNN image, D, and the reference enhanced beamformed image, E, defined as:

$$\text{L1Loss}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{||D_i(I_d;\theta) - E_i||_1}{N} \tag{5.2}$$

where $||\cdot||_1$ is the ℓ_1 norm, D_i and E_i are the vectorized images for each training example, N is the total number of image pixels, and n is the total number of training examples in each mini-batch (i.e., the mini-batch size). The second loss was the Dice similarity coefficient, or DSCLoss, between the predicted DNN segmentation, S_p, and the true segmentation, S_t, defined as:

$$\text{DSCLoss}(\theta) = \frac{1}{n}\sum_{i=1}^{n} 1 - 2\frac{|S_{p,i}(I_d;\theta) \cap S_{t,i}|}{|S_{p,i}(I_d;\theta)| + |S_{t,i}|} \tag{5.3}$$

where $S_{p,i}$ and $S_{t,i}$ are the vectorized segmentation masks for each training example. While the target segmentation mask is binary valued, the predicted segmentation mask is allowed to be continous valued between 0 and 1 (with the range restricted by the sigmoid non-linearity in the final layer). A pixel value of 0 in the predicted segmentation can be interpreted as the pixel being predicted as tissue with 100% confidence, and a value of 1 can be interpreted as the pixel being predicted as cyst with 100% confidence. Thus, the DSCLoss function is implemented as a soft loss, ensuring gradient information can flow backwards through the network. The total network loss was the weighted sum of the two losses defined in Eqs. 5.2 and 5.3, each loss receiving a weight of one, as defined by:

$$\begin{aligned} L_T(\theta) &= \text{L1Loss}(\theta) + \text{DSCLoss}(\theta) \\ &= \frac{1}{n}\sum_{i=1}^{n} \frac{||D_i(I_d;\theta) - E_i||_1}{N} + 1 - 2\frac{|S_{p,i}(I_d;\theta) \cap S_{t,i}|}{|S_{p,i}(I_d;\theta)| + |S_{t,i}|} \end{aligned} \tag{5.4}$$

In summary, the network was trained to learn $\hat{y}$, which was composed of representations of E and S_t from input I_d, to jointly produce both the DNN image, D, and the DNN segmentation, S_p.

Unless otherwise stated, the DNN was trained using the following baseline settings. The Adam (Kingma and Ba, 2014) optimizer used a learning rate of 10^{-5} for 25 epochs, where one epoch is defined as one pass over the entire training dataset (i.e., the entire training dataset is once presented to the network for training). The mini-batch size for the training dataset was set to 16.

Training was performed on a system with an Intel Xeon E7 processor and four Tesla P40 GPUs, each equipped with 24 GB of graphics memory. To relate these computer specifications to a real-time frame rate, the training time for 25 epochs was 100 minutes. However, we contrast this with the inference time for our network to process 51,200 images, as reported in Section 5.3.

5.2.5 Comparisons to Training with Receive Delays Applied

To emphasize the challenge of deep learning from unfocused channel data, the input to the architecture shown in Fig. 5.2 was modified to be focused channel data and the first layer of this network was modified to accept the focused channel data. Specifically, the recorded channel data image, I, was transformed to the focused data tensor, I_f, by applying receive time delays, resulting in a 3D tensor with the new third dimension containing the number of focused scan lines. I_f was then downsampled (using the same downsampling procedure described in Section 5.2.3 to convert I to I_d), followed by the subaperture summation procedure as described in Hyun et al., 2019a, resulting in I_{fds}, which is a tensor of size $d \times w \times q_s$, where q_s is twice the number of subapertures, each representing the in-phase or quadrature component of

the recording. Our modified goal was to input I_{fds} to produce D and S_p, each with dimensions $d \times w$.

To perform subaperture beamforming (Hyun et al., 2019a), the third dimension of I_f (which contains the receive delays for each scan line) was divided into 16 subapertures (i.e., 8 elements per subaperture). The delayed data corresponding to each subaperture was summed, resulting in 16 complex valued images, one for each of the 16 subapertures. The I and Q channels of each subaperture were then grouped together within the third dimension of the tensor to give 32 feature channels in total. Although this subaperture beamforming was performed in software in this work for ease of demonstration of the feasibility of this approach, this subaperture beamforming step can also be implemented in hardware (Santos et al., 2016), which would still result in a raw channel data input to our network (yet has the expected trade-off of increased data transfer rates).

We employed the same FCNN described in Section 5.2.2 with the exception of a modified input layer and updated trainable parameters θ to learn the optimal mapping of $I_{fds} \rightarrow y$. Specifically, the first layer of the architecture shown in Fig. 5.2 was modified to accept 32 feature channels rather than two feature channels due to the subaperture beamforming step. This modified network was then trained as described in Section 5.2.4, after replacing I_d in Eqs. 5.2-5.4 with I_{fds}. The same computer described in Section 5.2.4 was used for training. Training time for this modified network was 315 minutes. However, we contrast this with the inference time for this network to process 51,200 images, as reported in Section 5.3.

Table 5.1: Simulated cyst image data parameters

Parameter	Range	Increment
Radius (r)	2-8 mm	1-2 mm
Speed of Sound (c)	1420-1600 m/s	10 m/s
Lateral position of cyst center (x)	-16 mm - 0 mm	2 mm
Axial position of cyst center (z)	40-70 mm	2.5 mm

5.2.6 Simulated Datasets for Training and Testing

The Field II (Jensen and Svendsen, 1992; Jensen, 1996) ultrasound simulation package was used to generate 22,230 simulations of individual anechoic cysts surrounded by homogenous tissue. We employed simulations in our training approach for two primary reasons. First, simulations enable the generation of large, diverse datasets that are required to train robust DNNs. Second, for segmentation tasks, simulations enable the specification of ground truth pixel labels, allowing one to avoid the expensive and time-consuming step of a human annotator to provide segmentation labels.

The simulated cyst radius (r), lateral and axial center positions of the cyst (x and z, respectively), and speed of sound in the medium (c) were varied using the range and increment sizes defined in Table 5.1. The values of r were 2, 3, 4, 6, and 8 mm, which is within the range of renal calyx, breast cyst, and ovarian follicle sizes (Cadeddu et al., 1997; Jackson, 1990; Vargas et al., 2004; Wikland et al., 2001). These cysts were contained within a cuboidal phantom volume located between an axial depth of 30 mm and 80 mm, with a lateral width of 40 mm, and an elevational thickness of 7 mm. The cysts were modeled as cylinders with the same diameter in each elevational cross section. Each simulation contained a unique speckle realization, enforced by using a

Table 5.2: Transducer parameters

Parameter	Simulated	L3-8	L8-17
Number of Elements	128	128	128
Pitch	0.30 mm	0.30 mm	0.20 mm
Element Width	0.24 mm	0.24 mm	0.11 mm
Kerf	0.06 mm	0.06 mm	0.09 mm
Aperture	38.4 mm	38.4 mm	25.6 mm
Elevational Width	7 mm	7 mm	4 mm
Elevational Focus	35 mm	35 mm	20 mm
Transmit Frequency	4 MHz	4 MHz	12 MHz
Sampling Frequency	100 MHz	40 MHz	40 MHz
Pulse length (cycles)	4	4	1
Center Frequency	5.5 MHz	5.5 MHz	12.5 MHz
Fractional Bandwidth	0.65	0.65	0.65

different seed for the random number generator. A total of 50,000 scatterers were contained within the simulated phantom to ensure fully developed speckle.

In each simulation, a single plane wave at normal incidence was simulated to insonify the region of interest. The simulated ultrasound probe matched the parameters of the Alpinion L3-8 linear array transducer, and its center was placed at the axial, lateral, and elevation center of the phantom (i.e., 0 mm, 0 mm, and 0 mm, respectively). The simulated probe parameters are summarized in Table 5.2. The one exception to matching the real hardware system was a simulated sampling frequency of 100 MHz (rather than the 40 MHz sampling frequency of the Alpinion ultrasound scanner used to acquire the experimental phantom and *in vivo* data described in Sections 5.2.7 and 5.2.8, respectively) in order to improve the Field II simulation accuracy (Jensen and Svendsen, 1992; Jensen, 1996).

A total of 80% of the 22,230 simulated examples was reserved for training, and the remaining 20% were used for network testing. Considering that cysts were purposely simulated to reside on the left side of the phantom (see Table 5.1), data augmentation was implemented by flipping the simulated channel across the $x = 0$ axis to incorporate right-sided cysts in our training and testing.

To investigate the impact of depth-dependent attenuation on network training sensitivity, half of the 22,230 simulated Field II examples were simulated with an attenuation coefficient of 0.5 dB/cm-MHz, and the remaining half did not include attenuation. One DNN was trained with attenuated data, a second DNN was trained with non-attenuated data, and a third DNN was trained with the combined dataset. Each network was trained for 27,625 iterations. Therefore, for this investigation, one epoch was considered to be either one pass over the combined dataset (i.e., for the third DNN) or two passes over either dataset with or without attenuation (i.e., for the first or second DNN, respectively), as each of these datasets is half the size of the combined dataset. Using these updated definitions, the three networks were trained for 25 epochs. Unless otherwise stated (i.e., when not investigating attenuation), results are reported for networks trained with the combined dataset.

5.2.7 Phantom Datasets

Channel data from a cross sectional slice of two anechoic cylinders in a CIRS 054GS phantom located at depths of 40 mm and 70 mm were acquired using an Alpinion L3-8 linear array ultrasound transducer attached to an Alpinion

E-Cube 12R research scanner. Two independent 80-frame sequences were acquired. The anechoic targets were consistently in the left or right half of the image for each acquisition sequence, achieved by manually flipping the ultrasound probe. In addition, the channel data corresponding to each of the 80 frames in each sequence was flipped from left to right, producing a dataset consisting of 320 total images in order to test the generalizability of the trained networks. The ground truth for this phantom dataset was specified by manually annotating pixels in the beamformed ultrasound image as cyst or tissue. When quantitatively evaluating these phantom examples, the mean result for the two anechoic cysts in the same image is reported, unless otherwise stated.

5.2.8 In Vivo Data

An 80-frame sequence of *in vivo* data from a simple anechoic cyst surrounded by breast tissue (denoted as Cyst #1) was acquired using an Alpinion L3-8 linear array transducer with parameters summarized in Table 5.2. Each plane wave acquisition was flipped from left to right to double this *in vivo* test dataset size. The ground truth for this *in vivo* dataset was specified by manually annotating pixels in the beamformed ultrasound image as cyst or tissue. In addition, the channel data input, I_d, was cropped to minimize the presence of bright reflectors that were not included during training. Because bright reflectors were not similarly prevalent after subaperture beamforming, the channel data input, I_{fds}, was not cropped until after images were created in order to match the field of view for more direct comparisons to the results

obtained with input I_d.

To highlight the versatility of the DNN trained with I_{fds}, this DNN was evaluated with a 10 frame sequence of an *in vivo* simple cyst surrounded by breast tissue (denoted as Cyst #2), which was originally acquired for the separate study reported in Wiacek et al., 2018. These data were acquired with focused (rather than plane wave) transmissions, using an Alpinion L8-17 linear array transducer with parameters for the acquisition listed in Table 5.2. We include this acquisition in this work to demonstrate that plane wave input data is not a requirement for the DNN trained with focused data. The ultrasound probe also has a range of different parameters (including transmit frequency) when compared to the L3-8 linear array, which was simulated and used to train the DNN, as reported in Table 5.2.

In addition to the channel data described above, clinical screenshots of the two *in vivo* cysts were additionally acquired with the Alpinion E-Cube 12R to assist with manual annotations of the cyst boundaries for ground truth segmentations. For Cyst #1, a noticeable deformation occurred between the acquisitions due to the sequential acquisition of clinical reference images followed by plane wave data acquisitions. Therefore, the clinical B-mode image was stretched and scaled and only used to help guide the segmentation boundary definition. The acquisition of all *in vivo* data was performed after informed consent with approval from the Johns Hopkins Medicine Institutional Review Board.

5.2.9 Comparison with Sequential Approaches

Results obtained with the trained DNNs were compared against four alternative and sequential approaches, namely DAS beamforming followed by non-local means (NLM), binary thresholding, NLM combined with binary thresholding, and a baseline U-Net architecture. The NLM (Buades, Coll, and Morel, 2005; Coupe et al., 2009) and binary thresholding algorithms were implemented in MATLAB on an Intel Xeon E 5645 CPU with a clock speed of 2.40 GHz. NLM served as a baseline image smoothing algorithm. Most hyperparameters were set to their default values (i.e., the 'SearchWindoSize' hyperparameter was set to 21, the 'ComparisonWindowSize' hyperparameter was set to 5), with the exception of the 'DegreeOfSmoothing' hyperparameter, which was set to 0.1.

Binary thresholding followed by morphological filtering (abbreviated as BT) was implemented as described in (Gomez et al., 2009; Luo et al., 2017; Noble and Boukerroui, 2006) to compare the DNN segmentations. To summarize our BT implementation, the mean of the normalized DAS B-mode image (I_n) was calculated, and the binarization decision threshold value was set as 0.70 times the mean pixel value. Pixels above and below the threshold were labeled as tissue and cyst, respectively. Connected components labeled as cyst tissue smaller than 50 pixels (i.e., an area of approximately 3 mm^2) were removed to eliminate false positives. Morphological closing (i.e., a dilation followed by an erosion) was then performed with a disk element of radius 1 pixel to fill in gaps in the segmentations. Morphological dilation dilation was then performed using a disk element of radius 2 pixels to expand the

cyst segmentations (considering that previously implemented steps tend to underestimate cyst size). Hyperparameter tuning was performed to choose the baseline hyperparameters.

DAS beamforming followed by NLM then BT (i.e., DAS+NLM+BT) was implemented to produce sequential segmentation and speckle reduced images for comparison to the parallel outputs produced by the DNN from raw IQ channel data. Finally, to compare results to the current state of the art for ultrasound image segmentation, a baseline U-Net (Ronneberger, Fischer, and Brox, 2015) network with a single encoder and a single decoder was implemented. This network was trained to predict a segmentation mask, $S_p(I_n;\theta)$, from input I_n, using S_t as the ground truth. We employed the same FCNN described in Section 5.2.2 with the exception of a modified input layer, a single decoder module, and updated trainable parameters θ to learn the optimal mapping of $I_n \rightarrow S_p$. Specifically, the first layer of the architecture shown in Fig. 5.2 was modified to accept one feature channel rather than two feature channels due to the input being the normalized DAS B-mode image, I_n. In addition, as only the DNN segmentation is being produced, only one decoder module is needed. This modified network was trained using the DSCLoss described by Eq. 5.3, after replacing $S_{p,i}(I_d;\theta)$ with $S_{p,i}(I_n;\theta)$. The same baseline settings and computer reported in Section 5.2.4 were used during training.

5.2.10 Evaluation Metrics

1. **Dice Similarity Coefficient (DSC):** DSC quantifies overlap between two segmentation masks (Zou et al., 2004). The DSC between the predicted

DNN segmentation, denoted by S_p and the true segmentation, denoted by S_t, is defined as:

$$\text{DSC}(S_p, S_t) = 2\frac{|S_p \cap S_t|}{|S_p| + |S_t|} \tag{5.5}$$

A perfect DNN segmentation produces a DSC of 1. Prior to display and evaluation, the predicted segmentation mask was binarized using a threshold of 0.5, considering that a predicted pixel value > 0.5 indicates that the network is more confident that the pixel is cyst than tissue (and vice versa for pixel values < 0.5).

2. **Contrast:** Contrast is fundamentally a measure to quantify differences between the minimum and maximum values in an image, particularly for regions inside and outside an anechoic cyst, respectively. This metric is defined as:

$$\text{Contrast} = 20\log_{10}\left(\frac{S_i}{S_o}\right) \tag{5.6}$$

where S_i and S_o represent the mean of individual uncompressed signal amplitudes, s_i and s_o, in selected regions of interest (ROIs) inside and outside the cyst, respectively, taken from the normalized image, I_n (see Eq. 5.1). The ROI inside the cyst was automated as a 2 mm-radius circular region centered at the cyst center for the simulated and phantom examples, and a 1.5 mm radius circular region for the more irregularly shaped *in vivo* examples. The choice to automatically use a small circular region about the cyst center was made to avoid manual ROI selection across the thousands of simulation and phantom test sets, yet still ensure

that the results would be a meaningful assessment of the difference in signal amplitude inside and outside the detected cyst region. This automated ROI selection additionally is intended to prevent the inclusion of misclassifications (e.g., cyst pixels at the cyst boundary detectected as tissue and vice versa), which are instead evaluated with the gCNR metric (Rodriguez-Molares et al., 2019). The ROI outside of the cyst was the same size as the inside ROI and was located at the same depth as the cyst. These ROIs were used to calculate the contrast of DNN, DAS beamformed, and enhanced beamformed images.

Because the desired DNN output image was log-compressed with a chosen dynamic range of 60 dB, an uncompressed signal, s was first calculated as:

$$s = 10^{s_{dB}/20} \tag{5.7}$$

where s refers to s_i or s_o (i.e., the subscripts were removed for simplicity), and s_{dB} is the log-compressed equivalent of s. The values of s were then used to calculate S_i and S_o in Eq. 5.6. Note that the maximum dynamic range of our network is 60 dB, which translates to a maximum possible contrast of 60 dB in the DAS beamformed and enhanced beamformed images.

3. **Signal-to-Noise Ratio (SNR):** Tissue SNR quantifies the smoothness of the background region surrounding the cyst, defined as:

$$\text{SNR} = \frac{S_o}{\sigma_o} \tag{5.8}$$

where σ_o represents the standard deviation of individual uncompressed signal amplitudes, s_o, in the selected ROI outside the cyst (i.e., the same ROI used to calculate contrast in Eq. 5.6). The enhanced beamformed image contains the same tissue background as the DAS beamformed image and therefore has identical SNR to the DAS beamformed image.

4. **Generalized Contrast-to-Noise Ratio (gCNR):** The gCNR was recently introduced as a more accurate measure of lesion detectability in comparison to CNR (Rodriguez-Molares et al., 2019), and it calculated as:

$$\text{gCNR} = 1 - \sum_{x=0}^{1} \min_{x} \{p_i(x), p_o(x)\} \tag{5.9}$$

where $p_i(x)$ and $p_o(x)$ are the probability mass functions of s_i and s_o, respectively. Considering that gCNR is intended to measure cyst detection probability, choosing the ROIs defined for contrast would bias gCNR toward better results by only providing a subset of pixels within the cyst region. Therefore, s_i for the gCNR metric was updated to be the ground truth cyst segmentation within S_t, and s_o was updated to be the same size and located at the same depth as s_i.

5. **Peak Signal-to-Noise Ratio (PSNR):** PSNR quantifies the similarity of the generated DNN image to the reference enhanced beamformed image, considering both the pixel values inside the cyst as well as the values outside the cyst to give a single value defining a global quality estimate,

defined as:

$$\text{PSNR}(D,E) = 10\log_{10}\left(\frac{MAX_E^2}{MSE}\right) \tag{5.10}$$

$$= 10\log_{10}\left(\frac{1}{\frac{||D-E||_2^2}{N}}\right) \tag{5.11}$$

where $||\cdot||_2$ is the ℓ_2 norm, D and E are the vectorized DNN image and the reference enhanced beamformed image respectively, N is the number of pixels in the images, and MSE is the mean square error between D and E. Because $E \in [0,1]$, MAX_E (i.e., the maximum absolute pixel value of image E) is equal to 1.

6. **Coefficient of Variation (CV):** To study the effect of minimal (e.g., due to hand tremors) to no perturbations in the phantom data across a given acquisition sequence, the coefficient of variation (CV) of the contrast, SNR, and gCNR metrics was calculated as:

$$\text{CV} = \frac{\sigma}{\mu} \times 100\% \tag{5.12}$$

where μ is the mean metric value across multiple acquisitions, and σ is the standard deviation of the metric across the same acquisitions. CV was calculated for both DNN and beamformed images.

7. **Processing Times:** Processing times for DAS beamforming, DNN performance, and NLM, BT, and U-Net comparisons were calculated. The processing time to perform DAS beamforming with a single plane wave was approximated from the GPU beamformer processing times for 25

plane waves reported in Hyun et al., 2019b. We included the times to perform the delay and sum operations (i.e., FocusSynAp and ChannelSum, respectively), and divided the summation of the reported processing times for these operations by 25 to achieve a processing time estimate for a single plane wave. The reported processing times were implemented on an NVIDIA Titan V GPU.

The processing times for NLM and BT were calculated after applying these algorithms to the entire test set of 4,554 simulated B-mode images. The total processing time was then divided by the total number of images processed to provide an estimate of the time to produce a single image. This time was added to the time per image reported for DAS beamforming to estimate the times for DAS+NLM, DAS+BT, and DAS+NLM+BT.

To calculate the processing times for U-Net segmentation, a mini batch of 512 tensors of simulated I_n were input 100 times into the trained network, and the total processing time was divided by the total number of images processed (i.e., 51,200 images). This time was added to the time per image reported for DAS beamforming to estimate the times for DAS+U-Net.

To calculate the processing time per image during DNN testing, a mini batch of 512 tensors of simulated I_d or I_{fds} were input 100 times into the DNN trained with unfocused or focused, data, respectively. The total processing time for each DNN was then divided by the total number of images processed (i.e., 51,200 images) to provide an estimate of the time

it would take to process a single image for each DNN.

Calculated processing times were then inverted to provide expected frame display rates. Although these reports combine CPU and GPU performance, we only perform direct comparisons of CPU-to-CPU and GPU-to-GPU processing times implemented on the same computer.

5.2.11 Exclusion Criteria

As demonstrated in our previous work (Nair et al., 2018a), higher DSCs are achieved with larger cysts compared to smaller cysts. In addition, small cysts have greater potential to be missed, which is quantified as a DSC of approximately zero. Based on this knowledge, we prioritize a fair comparison of the multiple network parameters, which we define as a minimum DSC ≥ 0.05. This criterion was required for the network trained with the baseline settings reported in Section 5.2.4, and test cases that did not meet this basic criterion with this baseline test set were excluded from the results reported in this work. Note that our exclusion criteria was only applied to one of several test sets, and the excluded images from this test set analysis were then excluded in subsequent test sets (i.e., the exclusion criteria was not repeated for each test set).

The resulting detection rate is listed for each cyst radius in Table 5.3. Overall, no experimental phantom or *in vivo* data met our exclusion criteria, and the network successfully detected the simulated cysts in $4,274$ out of $4,554$ test examples. Table 5.3 also indicates that segmentation failure primarily occurs with 2 mm-radius cysts. The remaining cyst examples were successfully

Table 5.3: Detection rate of simulated test set after training with the baseline parameters listed in Section 5.2.4 and implementing the exclusion criteria listed in Section 5.2.11

Cyst Radius	Total # of Images	# of Images Included	Detection Rate
2 mm	904	624	69%
3 mm	880	880	100%
4 mm	972	972	100%
6 mm	902	902	100%
8 mm	896	896	100%

detected, and we prefer to limit our methodology feasibility assessments to these cases. Therefore, the results in Section 5.3.1 are reported for this subset of the simulated test set. This information can additionally be used to avoid applications of our approach to cysts smaller than 2 mm radii, which are challenging for the DNN to detect, likely due to the presence of acoustic clutter in the single plane wave image.

5.3 Results

5.3.1 Simulation Results

Fig. 5.4 shows an example simulated test case from the DNN architecture shown in Fig. 5.2, using the baseline settings noted in Section 5.2.4. From top left to bottom right, this example shows simulated raw IQ channel data, the corresponding DAS beamformed ultrasound and DNN image, the known segmentation of the cyst from surrounding tissue, the DNN segmentation predicted by our network, and the DNN segmentation overlaid on the true segmentation. This example produces a DSC of 0.98, a contrast of -42.11 dB,

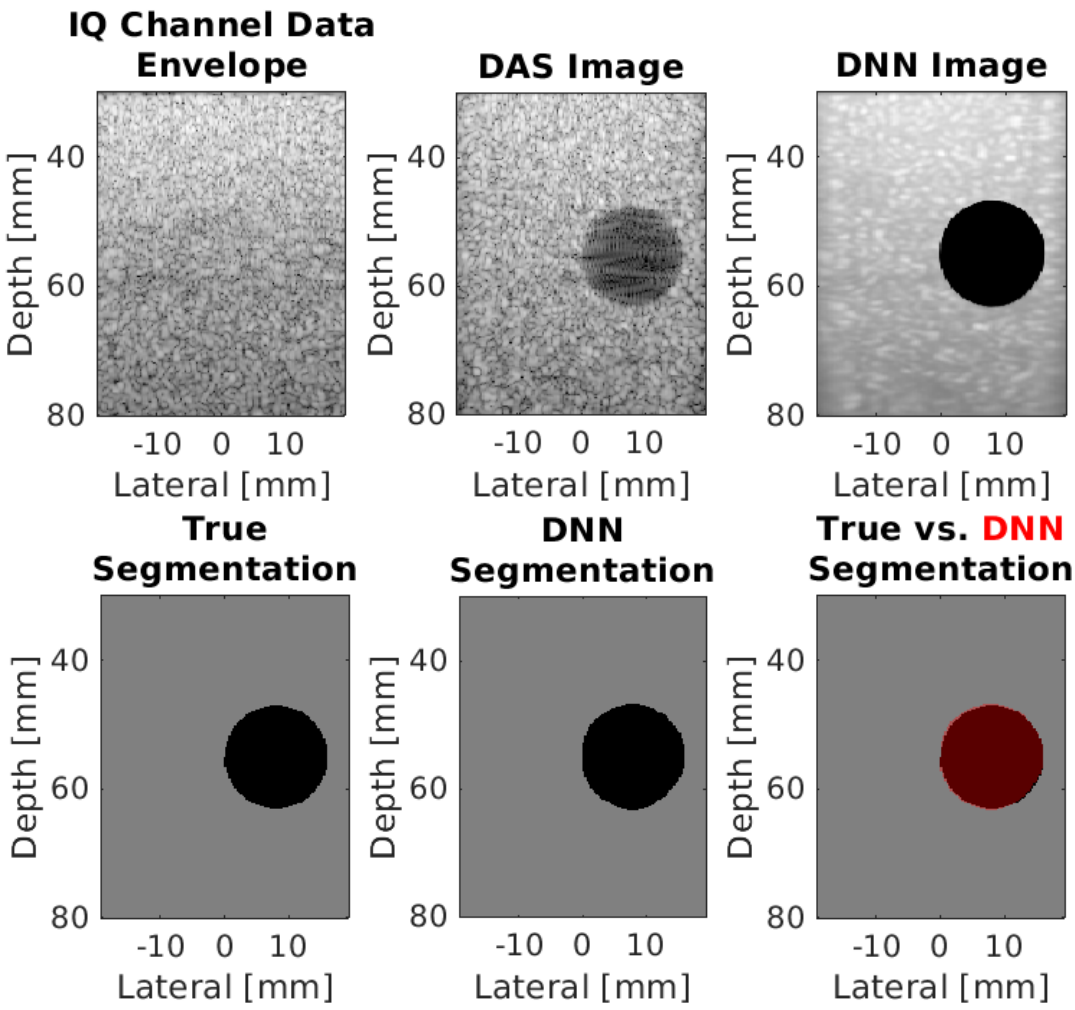


Figure 5.4: Simulation result showing, from top left to bottom right, raw IQ channel data (displayed with 60 dB dynamic range after after applying envelope detection and log compression), a DAS beamformed ultrasound image, a DNN image produced by our network, the known segmentation of the cyst from surrounding tissue, the DNN segmentation predicted by our network, and an image with a red transparent overlay of the DNN segmentation over the true segmentation.

an SNR of 3.06, a gCNR of 0.99, and a PSNR of 20.32 dB. The test set (excluding the cases noted in Section 5.2.11) produced a mean $\pm$ one standard deviation DSC of 0.92 ± 0.13, contrast of -40.07 ± 11.06 dB, SNR of 4.29 ± 1.26, gCNR of 0.95 ± 0.08 and PSNR of 20.19 ± 0.40 dB.

Fig. 5.5 shows the aggregated mean DSC, contrast, SNR, gCNR and PSNR

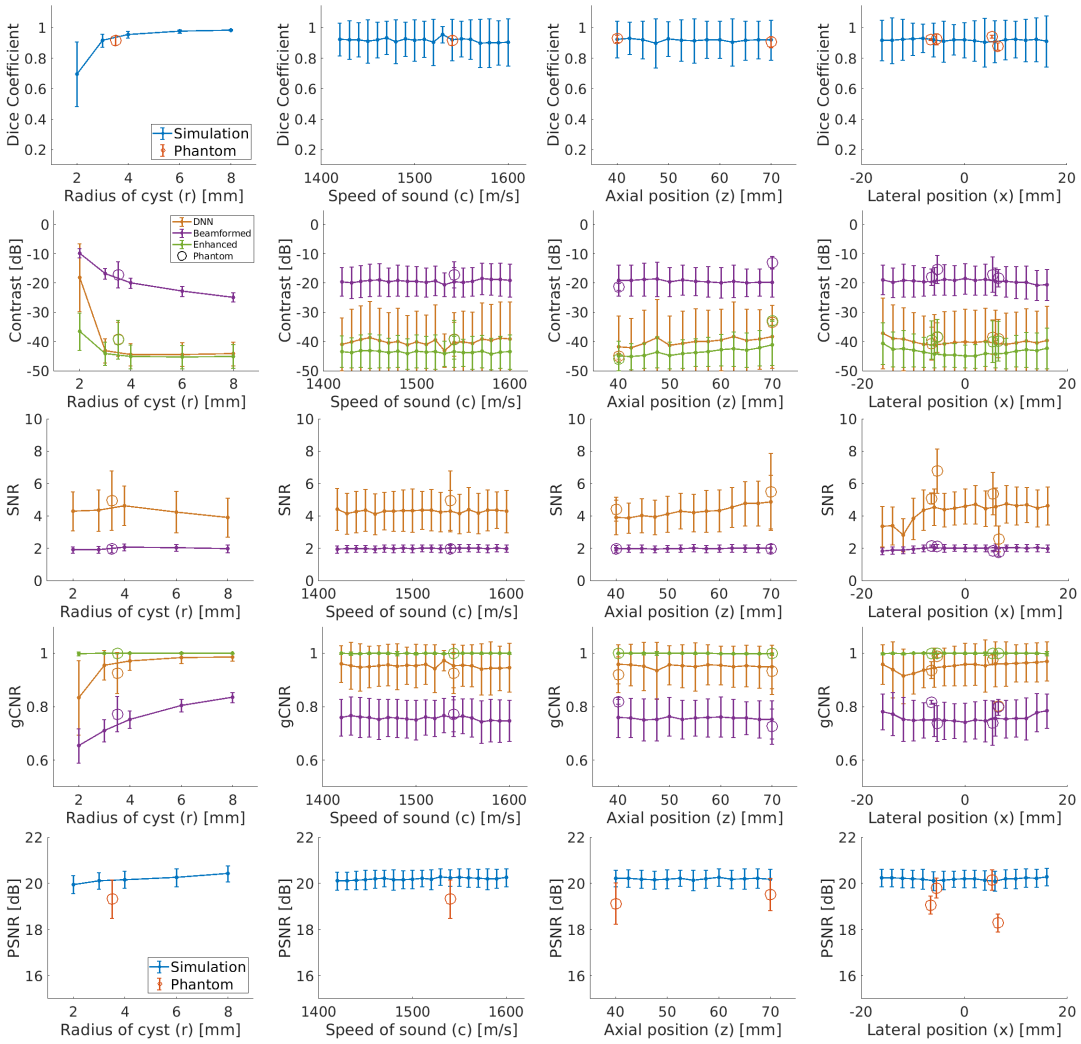


Figure 5.5: Aggregated mean (from top to bottom) DSC, contrast, SNR, gCNR and PSNR $\pm$ one standard deviation as a function of (from left to right) variation in r, c, z, and x for simulated results, and phantom results. Phantom results are displayed using unfilled circle markers. "Enhanced" indicates the performance of the enhanced B-mode images that were used for DNN training, as described in Section 5.2.3, and they represent the limits to an ideal DNN performance.

$\pm$ one standard deviation as a function of (from left to right) variation in r, c, z, and x for simulated results and phantom results. The simulation results in Fig. 5.5 reveal that the smaller, 2-mm radii cysts yield the worst DNN segmentations with a mean DSC of 0.70. The DSC rises to 0.99 for 8 mm

cysts. Similarly, as r increases, contrast improves from -18.12 dB to -44.20 dB, gCNR improves from 0.83 to 0.97, and PSNR improves from 19.95 dB to 20.42 dB. Unlike DSC, contrast, gCNR, and PSNR, SNR does not change as r increases. The DSC, contrast, SNR, and gCNR results are otherwise relatively constant as functions of the remaining parameters (i.e., c, z, and x).

Focusing on the contrast results in Fig. 5.5, the contrast of the DNN images approaches that of the enhanced beamformed image as r increases and is consistently superior to the contrast of the traditional DAS beamformed images, with a mean contrast improvement measuring 20.71 dB. In addition, Fig. 5.4 demonstrates that the tissue texture is smoother in the DNN images when compared to the DAS beamformed images. The quantitative SNR results in Fig. 5.5 support this observation, and the mean SNR improvement is 2.30. These two improvements combine to produce a mean gCNR improvement of 0.19 when DNN images are compared to DAS beamformed images.

5.3.2 Phantom Results

Fig. 5.6 shows an example test case from the phantom dataset. From top left to bottom right, this example shows raw phantom IQ channel data, a DAS beamformed ultrasound image and corresponding DNN image, the known segmentation of the cyst from surrounding tissue, the DNN segmentation predicted by our network, and the DNN segmentation overlaid on the true segmentation. This example produces a DSC of 0.92, a contrast of -40.69 dB, an SNR of 4.96, a gCNR of 0.93, and a PSNR of 18.97 dB. The entire test set produced a mean $\pm$ one standard deviation DSC of 0.92 ± 0.03, contrast

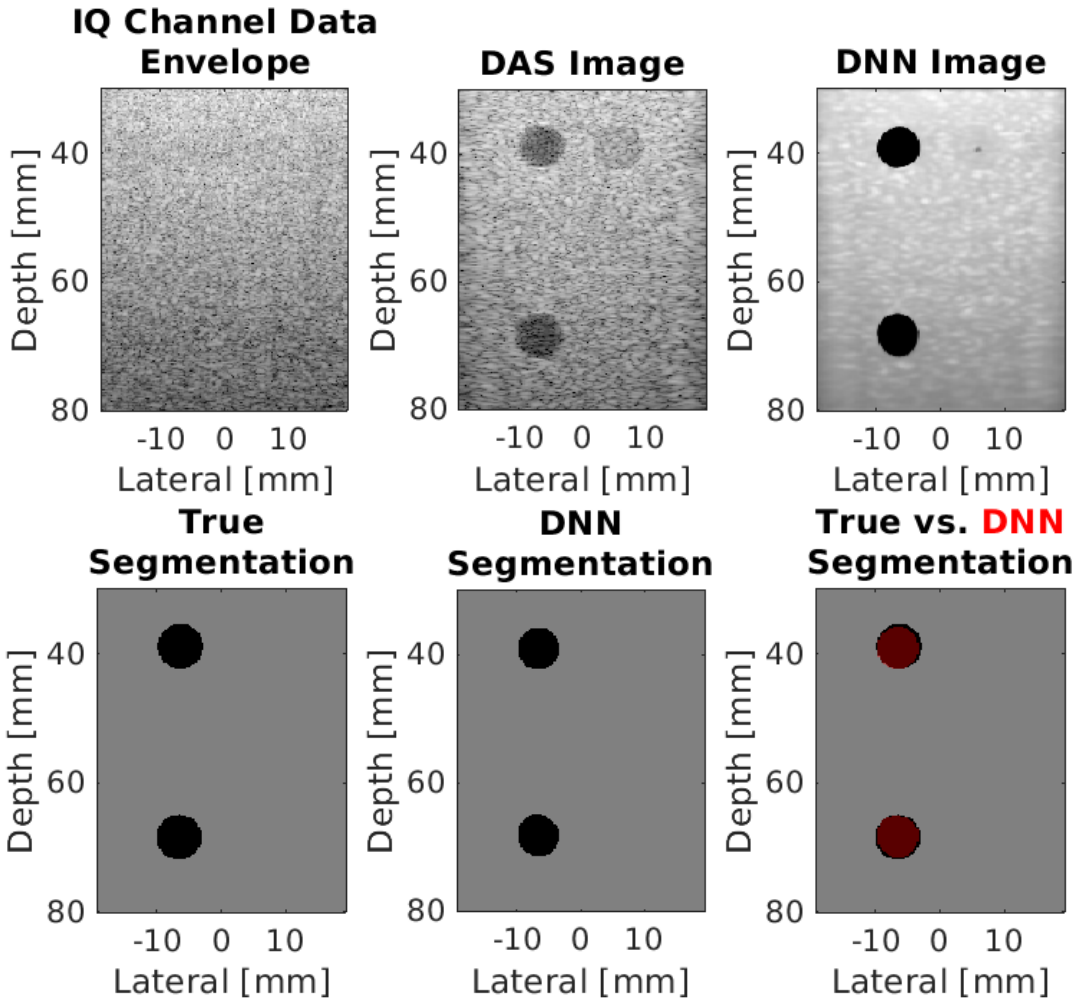


Figure 5.6: Phantom result showing, from top left to bottom right, raw IQ channel data (displayed with 60 dB dynamic range after after applying envelope detection and log compression), a DAS beamformed ultrasound image, a DNN image produced by our network, the known segmentation of the cyst from surrounding tissue, the DNN segmentation predicted by our network, and an image with a red transparent overlay of the DNN segmentation over the true segmentation.

of -39.13 ± 5.86 dB, SNR of 4.96 ± 1.84, gCNR of 0.93 ± 0.08, and PSNR of 19.33 ± 0.83 dB.

The aggregated results of this entire dataset as functions of r, c, z, and x are shown in Fig. 5.5 as unfilled circles overlaid on the previously discussed simulation results. The color of each circle corresponds to the color-coded

data type listed in the legend. Fig. 5.5 shows that the mean DSC, contrast, and gCNR measurements for the phantom results are generally within the range of the standard deviations of these same measurements for the simulation results. However, the SNR and PSNR of the phantom results are outliers when compared to those of the simulation results, because of the differences in tissue texture achieved with the DNN image.

Note that the phantom test dataset consists of 160 total plane wave insonifications. Half of these acquisitions contain the two anechoic cysts on the left side of the image, and the other half (acquired with the probe physically flipped) contain the same anechoic cysts on the right side of each image. The raw data from each acquisition was then flipped, yielding a dataset with a total of 320 plane waves and a total of eight individual "cyst templates." CV was calculated for each individual cyst template, and the mean of these eight CVs was 0.12%, 2.38%, and 0.36% for DNN image contrast, SNR, and gCNR measurements, respectively. These results are comparable to those of the DAS beamformed images (i.e., contrast, SNR, and gCNR CVs of 1.19%, 0.63%, and 0.82%, respectively). This result indicates that there were minimal variations in the acquired phantom results which were purposely acquired with minimal to no perturbations to the acquisition setup. The implication of this result is discussed in more detail in Section 5.4.

5.3.3 Incorporating Attenuation

Fig. 5.7 (top) shows example test cases from the three networks trained with, without, and both with and without attenuation combined. From left to

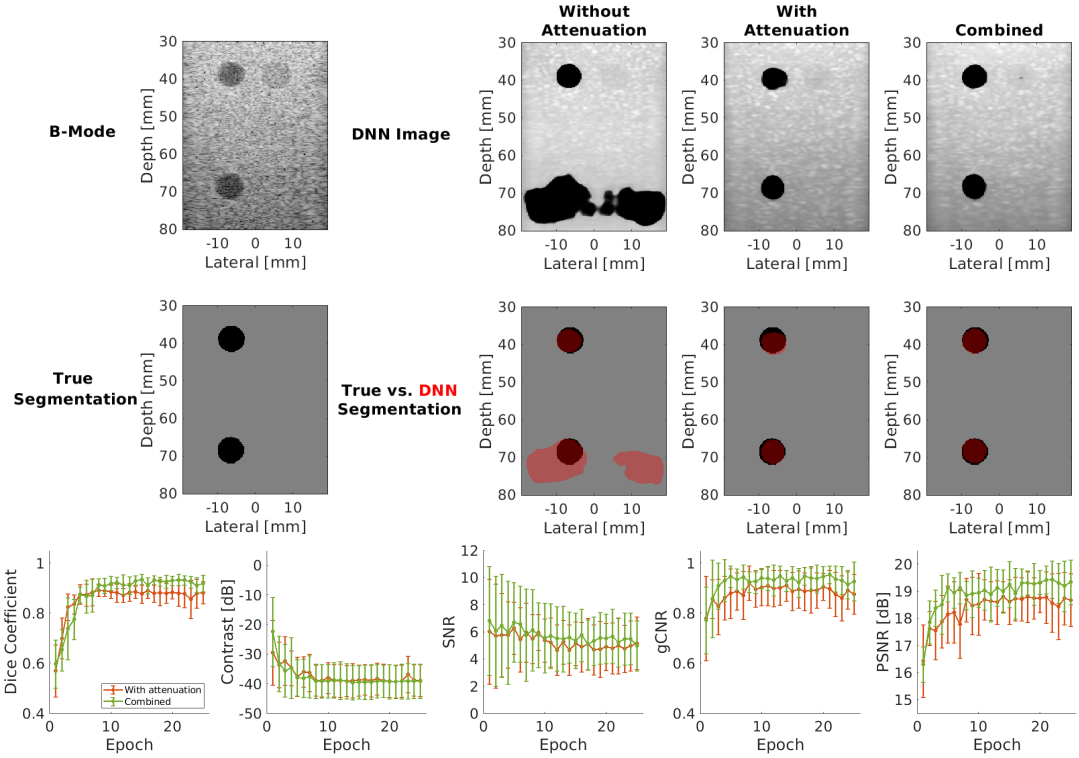


Figure 5.7: (top) Attenuation results showing, from left to right, the DAS beamformed image and ground truth segmentation reference pair, the corresponding outputs of the network trained with non-attenuated data, attenuated data, and the combined dataset of both attenuated and non-attenuated data. (bottom) Aggregated attenuation results, showing mean DSC, contrast, SNR, gCNR and PSNR $\pm$ one standard deviation as a function of epoch.

right, the first column of images displays the DAS beamformed image along with the true segmentation, the second column displays the output of the network trained without attenuation, the third column displays the output of the network trained with attenuated data, and the fourth column displays the output of the network trained with the combined dataset of both attenuated and non-attenuated data. The example output from the network trained with non-attenuated data produced DSC, contrast, SNR, gCNR, and PSNR of 0.66, -41.64 dB, 3.08, 0.64, and 14.16 dB, respectively. The network trained with attenuated data produced DSC, contrast, SNR, gCNR, and PSNR of

0.86, −40.27 dB, 4.79, 0.85, and 18.16 dB, respectively, representing improved DSC, SNR, gCNR, and PSNR with similar contrast. Similar improvements were achieved when training with both attenuated and non-attenuated data, producing DSC, contrast, SNR, gCNR, and PSNR of of 0.92, −40.69 dB, 4.96, 0.93, and 18.97 dB, respectively.

Fig. 5.7 (bottom) shows the aggregated mean DSC, contrast, SNR, gCNR, and PSNR ± one standard deviation as a function of the number of epochs for the networks trained with attenuated data and with the combined dataset of both attenuated and non-attenuated data. When trained with the combined dataset, it is remarkable that the addition of non-attenuated data does not significantly impact the performance of the network in spite of the test phantom dataset having tissue attenuation. Instead, the inclusion of non-attenuated data seems to be responsible for a subtle boost in performance. For example, when the measured DSC is averaged over epochs 11 through 25, this average improves from 0.88 when the network is trained with the attenuated dataset to 0.92 when the network is trained with the combined dataset. Similarly, when each metric result is averaged over all epochs, SNR improves from 5.26 to 5.73, gCNR improves from 0.88 to 0.92, and PSNR improves from 18.38 dB to 18.98 dB. Contrast results are similar between the two networks.

5.3.4 Comparisons Between Focused and Unfocused Input Data

Fig. 5.8 shows phantom images comparing unfocused input data, I_d, to focused input data, I_{fds}. The contrast, SNR, and gCNR of the image created with the focused input is −36.22 dB, 1.63, and 0.94, respectively. The corresponding

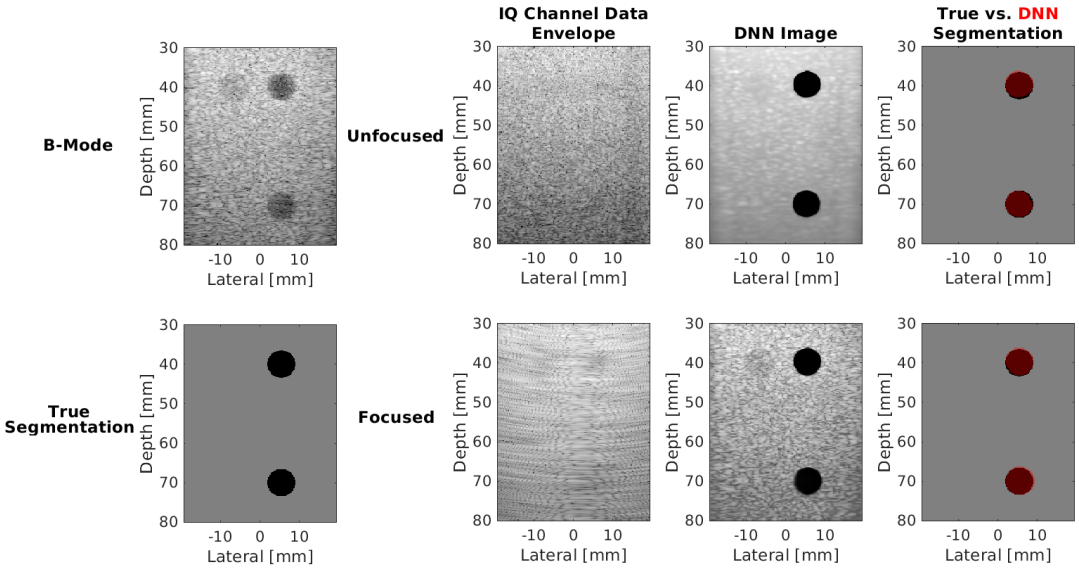


Figure 5.8: Comparison of I_d and I_{fds} input phantom data showing, from left to right, the DAS beamformed image and ground truth segmentation reference pair, the unfocused and focused IQ channel data envelopes of the input data I_d and I_{fds}, respectively, and corresponding outputs of the two DNNs. For the focused IQ channel data envelope image, a subaperture near the center of the probe is displayed as a representation of the input to one channel of the DNN.

values for the image created with unfocused data are -38.41 dB, 5.61, and 0.98, respectively. Therefore, these metrics are improved with unfocused data in this particular example. However, the PSNR and DSC are 20.14 dB and 0.94, respectively, with the unfocused input, compared to 22.63 dB and 0.94, respectively, with the focused input. While the higher PSNR with the focused input is due to tissue SNR that more closely resembles that of the DAS B-mode images, the similar DSC results demonstrate that the similar segmentation performance can be achieved with DNNs regardless of the inclusion of focusing. Table 5.4 summarizes these metrics for the acquired phantom images, and this table also compares the time required to create each image.

Table 5.4 further demonstrates that similar image quality to the reference

Table 5.4: Performance comparisons of DAS beamforming, non-local means (NLM) speckle reduction, binary thresholding segmentation followed by morphological filtering (abbreviated as BT), U-Net segmentation, and DNN results with focused and unfocused input data. Processing times for NLM and BT were calculated on a CPU with remaining processing times calculated on GPUs.

		Traditional Sequential Approaches				**Proposed DNN Approaches**	
	DAS	**DAS+NLM**	**DAS+BT**	**DAS+NLM+BT**	**DAS+U-Net**	**Unfocused DNN Input, I_d**	**Focused DNN Input, I_{fds}**
Processing Time	0.25 ms	13.29 ms	2.20 ms	15.41 ms	1.72 ms	2.53 ms	3.48 ms
Frame Rate	4,000 Hz	75 Hz	455 Hz	64 Hz	583 Hz	395 Hz	287 Hz
Phantom							
DSC	N/A	N/A	0.68 ± 0.09	0.77 ± 0.08	0.92 ± 0.02	0.92 ± 0.03	$\mathbf{0.93 \pm 0.01}$
Contrast (dB)	-17.14 ± 4.51	-16.08 ± 4.51	-17.14 ± 4.51	-16.08 ± 4.51	-17.14 ± 4.51	$\mathbf{-39.13 \pm 5.86}$	-37.30 ± 6.86
SNR	1.97 ± 0.22	5.76 ± 2.03	1.97 ± 0.22	$\mathbf{5.76 \pm 2.03}$	1.97 ± 0.22	4.96 ± 1.84	1.82 ± 0.37
gCNR	0.77 ± 0.07	0.94 ± 0.03	0.77 ± 0.07	0.94 ± 0.03	0.77 ± 0.07	0.93 ± 0.08	$\mathbf{0.95 \pm 0.03}$
PSNR (dB)	N/A	17.22 ± 0.99	N/A	17.22 ± 0.99	N/A	19.33 ± 0.83	$\mathbf{23.07 \pm 0.86}$
***In Vivo* Cyst #1**							
DSC	N/A	N/A	0.68 ± 0.00	0.76 ± 0.00	$\mathbf{0.83 \pm 0.01}$	0.77 ± 0.07	0.82 ± 0.03
Contrast (dB)	-13.61 ± 2.36	-11.43 ± 2.48	-13.61 ± 2.36	-11.43 ± 2.48	-13.61 ± 2.36	$\mathbf{-25.72 \pm 9.25}$	-25.30 ± 3.69
SNR	1.27 ± 0.07	1.76 ± 0.15	1.27 ± 0.07	1.76 ± 0.15	1.27 ± 0.07	$\mathbf{3.94 \pm 0.59}$	1.12 ± 0.21
gCNR	0.56 ± 0.03	0.76 ± 0.03	0.56 ± 0.03	0.76 ± 0.03	0.56 ± 0.03	0.75 ± 0.14	$\mathbf{0.89 \pm 0.04}$
PSNR (dB)	N/A	16.47 ± 0.01	N/A	16.47 ± 0.01	N/A	15.05 ± 0.86	$\mathbf{18.86 \pm 0.31}$
***In Vivo* Cyst #2**							
DSC	N/A	N/A	0.78 ± 0.01	$\mathbf{0.81 \pm 0.00}$	0.72 ± 0.07	-	0.79 ± 0.02
Contrast (dB)	-18.27 ± 2.59	-16.40 ± 2.55	-18.27 ± 2.59	-16.40 ± 2.55	-18.27 ± 2.59	-	$\mathbf{-31.62 \pm 2.56}$
SNR	1.29 ± 0.13	3.08 ± 0.38	1.29 ± 0.13	$\mathbf{3.08 \pm 0.38}$	1.29 ± 0.13	-	1.39 ± 0.12
gCNR	0.75 ± 0.09	0.94 ± 0.02	0.75 ± 0.09	0.94 ± 0.02	0.75 ± 0.09	-	$\mathbf{0.96 \pm 0.01}$
PSNR (dB)	N/A	19.45 ± 0.02	N/A	19.45 ± 0.02	N/A	-	$\mathbf{19.58 \pm 0.16}$

B-mode image is achieved when the input data is focused to include receive time delays. However, this focusing approach requires an updated network input layer with 30 additional input channels (to accept the increased input data size), as well as the additional step of subaperture beamforming, which both reduce the overall frame rates. Note that the additional step associated with subaperture beamforming is not included in the processing time results reported in Table 5.4, as subaperture beamforming could be implemented in hardware.

Fig. 5.9 shows *in vivo* images of Cyst #1 comparing unfocused input data, I_d, to focused input data, I_{fds}. The DSC, contrast, SNR, gCNR, and PSNR of the outputs created with the unfocused input are 0.83, -34.89 dB, 4.57, 0.90, and 15.85 dB, respectively. Although the DSC and gCNR results are lower

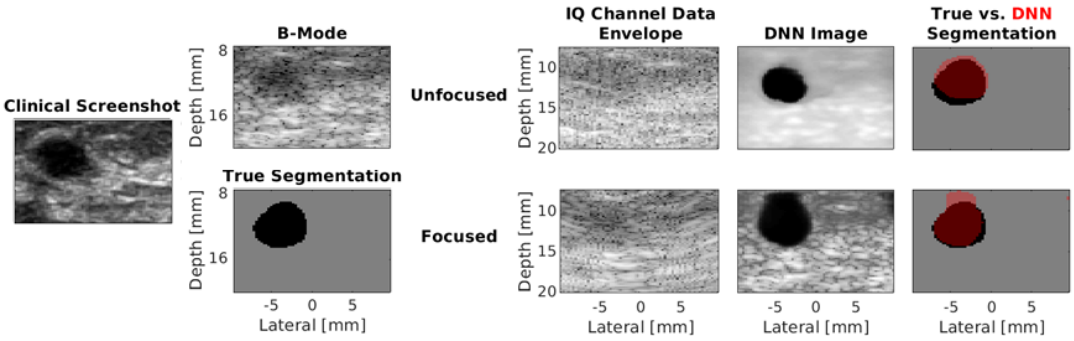


Figure 5.9: Comparison of I_d and I_{fds} input *in vivo* data from Cyst #1 showing, from left to right, the clinical image obtained from the scanner with an 8 MHz transmit frequency focused at a depth of 20 mm, the DAS beamformed image of Cyst #1 obtained using a single 0° incidence plane wave transmitted at 4 MHz and the corresponding ground truth segmentation reference pair, the unfocused and focused IQ channel data envelopes (with the latter showing the envelope of a single subaperture) of the input data I_d and I_{fds}, respectively, and corresponding outputs of the two DNNs.

than the majority of examples previously shown, it is important to note that the size of Cyst #1 is approximately 3 mm in radius, and the DSC and gCNR results of this cyst are within the range of the means $\pm$ one standard deviation obtained for the 2-4 mm radii results reported in Fig. 5.5 (i.e., 0.70 ± 0.21 to 0.96 ± 0.2 and 0.83 ± 0.14 to 0.97 ± 0.03, respectively). In addition, SNR starts at a lower value than the phantom and simulated DAS results reported in Fig. 5.5, therefore the final value obtained with the DNN is also lower than those shown in Fig. 5.5. Nonetheless, there is still an SNR increase and contrast is improved in the DNN image compared to the DAS image.

The DSC, contrast, SNR, gCNR, and PSNR of the outputs created with the focused input, I_{fds}, are 0.85, -21.89 dB, 0.93, 0.85, and 19.01 dB, respectively, for the example shown in Fig. 5.9. However, the DNN overestimates the proximal cyst boundary in this example, likely due to large amplitude differences at that boundary, which were not included during training. The

mean $\pm$ standard deviation of the evaluation metrics for the entire 160 frames in the test dataset for Cyst #1 are reported in Table 5.4.

Figs. 5.8 and 5.9 reveal that more similar SNR results were obtained with phantom and *in vivo* data when I_{fds} was the input, as summarized in Table 5.4. In particular, with I_{fds} as the input, the SNRs of the phantom and *in vivo* data more closely match the SNR results reported for the corresponding DAS B-mode images. The higher tissue SNR of DNN images obtained with I_d as the input, when compared to corresponding DAS images, occurs because of the smoother tissue texture in these DNN images, despite both DNNs being trained with data that fundamentally contains speckle, which is caused by constructive and destructive interference from sub-resolution scatterers (Wagner, 1983; Burckhardt, 1978).

These SNR results demonstrate that the DNN with I_d as input is unable to learn the finer details associated with the transformation from unfocused tissue texture to traditional B-mode image speckle (which is included in the transformation $I_d \rightarrow D$), and therefore $\hat{y}$ is not a faithful representation of y from this perspective. In contrast, considering that the same network architecture was implemented after receive focusing delays were applied to the input data (and after the input layer was modified to accept this larger input data), the transformation $I_{fds} \rightarrow D$ appears to be a simpler task for this DNN, which can be explained by the transformation from focused tissue texture to speckle being a more direct image-to-image transformation.

While the smoothing and higher SNRs observed in the output DNN images created from the unfocused input data, I_d, may be viewed as a failure

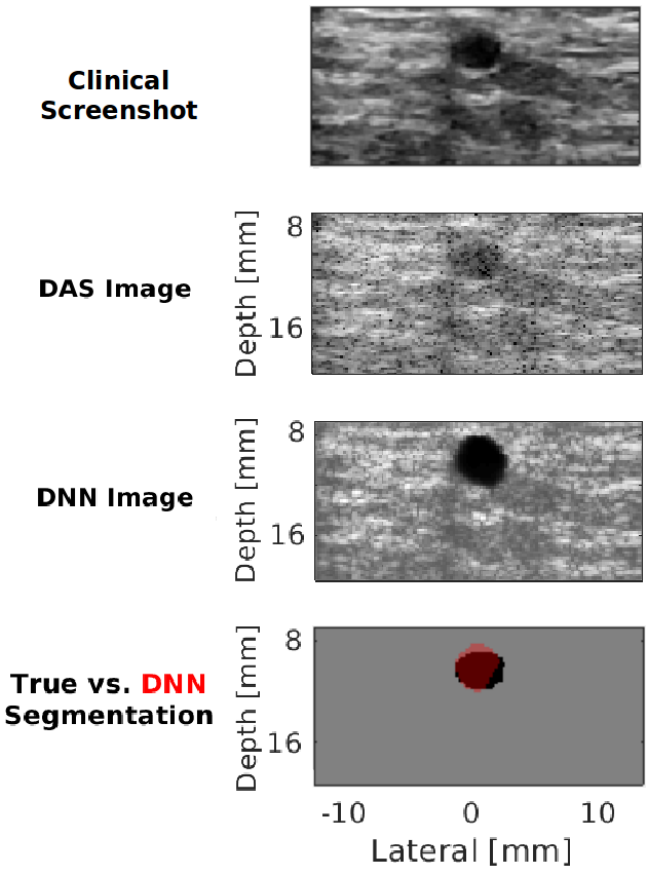


Figure 5.10: *In vivo* clinical image of Cyst #2 obtained from the scanner with a 12 MHz transmit frequency focused at a depth of 10 mm, DAS beamformed image of Cyst #2, the corresponding DNN image, and the corresponding DNN segmentation overlaid on the true segmentation.

of the network from the perspective of faithful image reconstruction, from the perspective of the proposed task and the DNN goals, the higher tissue SNR and smoother tissue texture is viewed as a benefit. These achievements are aligned with the goals of maximizing achievable frame rates, deemphasizing unimportant structures, and emphasizing structures of interest for the proposed task.

Fig. 5.10 shows an additional example of this expected trade-off between

preserving fidelity and achieving task-specific image reconstruction goals with Cyst #2. This example was obtained from focused transmissions and with a higher transmit frequency than that used during training, thus highlighting the versatility of the DNN with I_{fds} as input. This network produces DNN images that have a closer match to the DAS beamformed image, but the DNN image contains tissue structure and speckle that can potentially confuse an observer who is not skilled with reading ultrasound images (in addition to requiring more time to produce this image in comparison to the image that would be produced with an unfocused data input). The DSC, contrast, SNR, gCNR, and PSNR for this result are 0.82, -34.18 dB, 1.50, 0.97, and 19.45 dB, respectively. The mean $\pm$ standard deviation of these metrics for the entire 20 frames in the test dataset for Cyst #2 are reported in Table 5.4.

When comparing the presented DNN performance to more standard methods, Table 5.4 demonstrates that although B-mode alone produces the fastest frame rates (i.e., 4,000 Hz on a GPU), frame rates are expected to be reduced after image formation followed by either speckle reduction (i.e., DAS+NLM results in 75 Hz on a GPU+CPU), segmentation (i.e., DAS+BT results in 455 Hz on a GPU+CPU), or both speckle reduction and segmentation (i.e., DAS+NLM+BT results in 64 Hz on a GPU+CPU). The DNN that accepts unfocused data has faster frame rates (i.e., 395 Hz) when compared to the DNN that accepts focused data (i.e., 287 Hz). Although implementation on two different GPU configurations confounds direct processing time comparisons, the sequential DAS+U-Net approach was faster than the parallel DNN approaches. There is also room for improvement of the parallel DNN approaches

to achieve even faster frame rates than currently reported (Bianco et al., 2018), particularly when considering that Table 5.4 reports initial proof-of-principle results and network optimization typically follows after demonstrations of feasibility.

Table 5.4 also demonstrates that the DNN that accepts unfocused data achieves consistently higher DSC and contrast when compared to DAS+NLM+BT. The DNN that accepts focused data consistently achieves similar or better DSC results when compared to the state of the art (i.e., DAS+U-Net) and consistently improves image quality (i.e., contrast, gCNR, and PSNR) when compared to DAS+NLM, DAS+BT, DAS+NLM+BT, and DAS+U-Net. These improvements were achieved in parallel rather than sequentially, due to our task-specific training on enhanced B-mode images for simultaneous detection, visualization, and segmentation of anechoic cysts.

5.4 Discussion

The results presented in this work describe our initial successes and challenges with using deep learning to provide useful information directly from a single plane wave insonification. Overall, the proposed task-specific DNN approach is feasible. It is remarkable that acceptable images were achieved prior to the application of receive time delays to compensate for time of arrival differences. In particular, the contrast and gCNR of anechoic regions were improved with DNN images over DAS B-mode images created with a single plane wave, tissue SNR was either improved or similar depending on the inclusion of receive delays with subaperture beamforming, and DSC

values were similar, regardless of the presence of receive delays. Therefore, the benefits of this approach are that we can train exclusively on simulations of single plane wave transmissions, successfully transfer the trained networks to experimental single plane wave ultrasound data, and produce B-mode images of anechoic targets with superior contrast and gCNR (i.e., two metrics representing improved image quality) and either similar or smoother tissue texture compared to DAS beamforming. An additional benefit is that these image quality improvements were achieved while concurrently extracting segmentation information directly from the raw ultrasound channel data, resulting in similar or better segmentation performance with focused input data when compared to the current state of the art (see Table 5.4).

Typically, image formation is followed by segmentation, and this sequential process for singular plane wave transmissions generally has the limitations of reduced throughput, as well as poor image quality (which generally produces poor image segmentations). Increasing the number of plane wave transmissions further reduces throughput, yet improves image quality at the expense of frame rates. In addition to parallelizing image formation and segmentation, the proposed DNNs offer real-time feasibility (with frame rates of 287-395 Hz based on our hardware and network parameters) as well as improved image quality with a single plane wave transmission. There is additional room for improvement by optimizing the proposed implementation to increase real-time frame rates (Bianco et al., 2018) and to increase *in vivo* segmentation accuracy by including more features during training, which will be the focus of future work.

There are four key observations and insights based on the presented results of applying DNNs to the challenging task of reconstructing sufficient quality images from single plane wave channel data acquisitions. First, we successfully achieved one of the primary goals of our network training, which was to only display structures of interest and otherwise ignore (or de-emphasize) surrounding structures. For example, the higher SNR and smoother tissue texture with the unfocused input data align with our goal of de-emphasizing unimportant structures for robotic automation. It is additionally advantageous that this network produced images with smoother tissue texture without relying on computationally expensive methods, such as NLM (Coupé et al., 2009) or anisotropic diffusion (Yu and Acton, 2002), to generate training data. If speckle is truly desired, we previously demonstrated that a GAN, rather than the FCNN employed in this work, has the potential to produce speckle and provide simultaneous DNN images and segmentation maps from a single input of unfocused plane wave channel data (Nair et al., 2019).

Similar to the FCNN deemphasis of speckle, the -6 dB cyst in Fig. 5.6 is poorly visualized in the DNN image. Although the network was trained with anechoic cysts and was not trained to detect hypoechoic cysts, this result suggests that the decoder for the DNN image is somewhat sensitive to echogenecity. However, the hypoechoic cyst in Fig. 5.6 does not appear in the DNN segmentation output, which suggests that the decoder for the segmentation is selective to the detection of anechoic regions in the input data. Similar task-specific DNN approaches may be devised and implemented to emphasize (as demonstrated with anechoic regions) or de-emphasize (as

demonstrated with speckle and the low-contrast cyst) other structures of interest for ultrasound-based interventions (e.g., needle tips).

The second insight is that the results of the attenuation study (see Fig. 5.7) indicate that the DNN trained without simulated depth-dependent attenuation learns to be sensitive to the amplitude of received echoes in order to determine if a given region is cyst or tissue. However, tissue attenuation confounds this particular network and causes performance deeper into the tissue to drop, as the network confuses the decrease in echo intensity due to tissue attenuation with a decrease in echo intensity due to an anechoic cyst. Counterintuitively, we noticed that performance rises when unrealistic data in the form of the dataset without attenuation (in addition to data containing attenuation) is included in the training dataset provided to the network. This rise in performance highlights the importance of diversity in the dataset – more diverse data yields better generalization. It also showcases that the network has the potential to automatically learn what is useful (e.g., the location-dependent spatial response of the cysts) and discard what is not useful (e.g., the unrealistic lack of attenuation) with additional training data.

Third, although the DNNs were trained with circular, anechoic, cyst-like structures, there was some ability to generally distinguish tissue from cyst in the presence of irregular boundaries (see Fig. 5.9), although the boundaries themselves seemed to be estimated by the DNN as smooth and circular like the training data. The DNNs also generalized reasonably well to cyst sizes that were not included during training. The network that accepted focused data was additionally able to generalize to data acquired with focused

rather than plane wave transmissions, as shown in Fig. 5.10. There were also generalizations across transmit frequencies and other parameters that differ when comparing the Alpinion L3-8 and L8-17 ultrasound transducer parameters in Table 5.2. In addition, although the DNN that accepts focused data was trained with data containing mostly homogeneous tissue, it was able to generalize to the heterogeneities of the majority of breast tissue surrounding Cysts #1 and #2. One possible reason for poorer performance with Cyst #1 is the presence of bright reflectors in the channel data, which were not included during training. Future work will include additional modeling of heterogenous tissue. Nonetheless, the observed generalizations are promising for translation to other organs of interest for the proposed DNN (e.g., kidney calyces and ovarian follicles), as well as to other anatomical structures with similar characteristics.

The fourth observation is that the $<$2.5% mean CV values reported in Section 5.3.2 indicate stability and robustness when there is minimal to no perturbations in the input over time. This minimal CV also demonstrates that similar results were produced over the acquisition sequences. Stability and robustness are desirable properties of DNNs (Papernot et al., 2016), which are particularly necessary for biomedical imaging tasks, as imperceptibly small perturbations to the input can often significantly alter the output.

Aside from the common limitations of pilot testing (including few *in vivo* test cases and questions about generalizability to other cases), one limitation observed from the presented results is that smaller cysts presented a greater challenge than larger cysts. This observation is based on the worse DSC,

contrast, and gCNR with smaller cysts compared to larger cysts in Fig. 5.5, and the lower cyst detection ratio for smaller cysts compared to larger cysts in Table 5.3. It is known that the DSC penalizes errors obtained with smaller cysts more severely than errors obtained with larger cysts (Glocker et al., 2007). While the lower DSCs with smaller cysts are consistent with DSCs achieved with other segmentation approaches (Pons et al., 2016; Kumar et al., 2018), the degraded contrast and gCNR with decreased cyst size might be linked to the context-detail tradeoff inherent to deep learning. Prior work (Yuille and Liu, 2018) demonstrated that CNNs rely on sufficient context to make successful predictions. Linearly interpolating the data to a reduced grid size of 256 x 128 pixels provides each neuron in the CNN with greater context as each neuron sees more of the neighborhood of a particular pixel to make a prediction. However, downsampled data has reduced detail, with the same 2 mm cyst now occupying fewer input pixels in the input to a given neuron. We hypothesize that linearly downsampling to a larger grid size is one possible solution toward addressing the poorer performance with smaller cysts.

The success of the presented results has implications for providing multiple (i.e., more than two) DNN outputs from a single network input. For example, in addition to beamforming and segmentation, deep learning ultrasound image formation tasks have also been proposed for sound speed estimation (Feigin, Freedman, and Anthony, 2018), speckle reduction (Hyun et al., 2019a), reverberation noise suppression (Brickson, Hyun, and Dahl, 2018), and minimum-variance directionless response beamforming (Simson et al., 2019), as well as to create ultrasound elastography images (Wu et al.,

2018), CT-like ultrasound images (Vedula et al., 2017), B-mode images from echogenecity maps (Tom and Sheet, 2018), and ultrasound images from 3D spatial locations (Hu et al., 2017). We envisage the future use of parallel networks that output any number of these or other mappings to provide a one-step approach to obtain multimodal information, each originating from a singular input of raw ultrasound data.

One example of a specific future application possibility from this perspective, which is also supported by the results presented in this work, is high-frame rate decision support without requiring multiple different transmit sequences to obtain multiple different output images. More specifically, the parallel B-mode and segmentation information can possibly be extended to include parallel B-mode, segmentation, elastography, sound speed estimation, and CT-like ultrasound images. One could also envision periodically interspersing the more accurate focused DNN results (compared in Figs. 5.8) among the faster unfocused results to increase the confidence of system performance. These possibilities open new avenues of research to explore the benefits of producing multiple outputs from a single input for parallel clinical, automated, and semi-automated decision making.

5.5 Conclusion

This work demonstrates a possible use of DNNs to create ultrasound images and cyst segmentation results directly from raw single plane wave channel data. This approach is a promising alternative to traditional DAS beamforming followed by segmentation. A novel DNN architecture was developed and

trained with Field II simulated data containing anechoic cysts insonified by single plane waves. The feature representations learned by the DNN from simulated data were successfully transferred to real phantom and *in vivo* data. This success has future implications for task-specific ultrasound-based approaches to emphasize or de-emphasize structures of interest and for producing more than two output image types from a single input image of raw IQ channel data, opening up new possibilities for ultrasound-based clinical, interventional, automated, and semi-automated decision making.

5.6 Acknowledgment

The authors thank Alycen Wiacek and Drs. Eniola Oluyemi and Emily Ambinder for their assistance with *in vivo* data acquisition.

References

Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio (2016). *Deep learning*. Vol. 1. 2. MIT press Cambridge.

Montaldo, Gabriel, Mickaël Tanter, Jérémy Bercoff, Nicolas Benech, and Mathias Fink (2009). "Coherent plane-wave compounding for very high frame rate ultrasonography and transient elastography". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 56.3, pp. 489–506.

Nair, Arun Asokan, Kendra N Washington, Trac D Tran, Austin Reiter, and Muyinatu A Lediju Bell (2020). "Deep learning to obtain simultaneous image and segmentation outputs from a single input of raw ultrasound channel data". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 67.12, pp. 2493–2509.

Pons, Gerard, Joan Martí, Robert Martí, Sergi Ganau, and J Alison Noble (2016). "Breast-lesion segmentation combining B-mode and elastography ultrasound". In: *Ultrasonic imaging* 38.3, pp. 209–224.

Kumar, Viksit, Jeremy M Webb, Adriana Gregory, Max Denis, Duane D Meixner, Mahdi Bayat, Dana H Whaley, Mostafa Fatemi, and Azra Alizad (2018). "Automated and real-time segmentation of suspicious breast masses using convolutional neural network". In: *PloS one* 13.5, e0195816.

Mebarki, Rafik, Alexandre Krupa, and François Chaumette (2010). "2-d ultrasound probe complete guidance by visual servoing using image moments". In: *IEEE Transactions on Robotics* 26.2, pp. 296–306.

Entrekin, Robert R, Bruce A Porter, Henrik H Sillesen, Anthony D Wong, Peter L Cooperberg, and Cathy H Fix (2001). "Real-time spatial compound imaging: application to breast, vascular, and musculoskeletal ultrasound". In: *Seminars in Ultrasound, CT and MRI*. Vol. 22. 1. Elsevier, pp. 50–64.

Xian, Min, Yingtao Zhang, Heng-Da Cheng, Fei Xu, Boyu Zhang, and Jianrui Ding (2018). "Automatic breast ultrasound image segmentation: A survey". In: *Pattern Recognition* 79, pp. 340–355.

Noble, J Alison and Djamal Boukerroui (2006). "Ultrasound image segmentation: a survey". In: *IEEE Transactions on medical imaging* 25.8, pp. 987–1010.

Liu, Shengfeng, Yi Wang, Xin Yang, Baiying Lei, Li Liu, Shawn Xiang Li, Dong Ni, and Tianfu Wang (2019). "Deep learning in medical ultrasound analysis: A review". In: *Engineering*.

Perdios, D., A. Besson, F. Martinez, M. Vonlanthen, M. Arditi, and J. Thiran (2019). "On Problem Formulation, Efficient Modeling and Deep Neural Networks for High-Quality Ultrasound Imaging : Invited Presentation". In: *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–4. DOI: 10.1109/CISS.2019.8692870.

Gasse, Maxime, Fabien Millioz, Emmanuel Roux, Damien Garcia, Hervé Liebgott, and Denis Friboulet (2017). "High-quality plane wave compounding using convolutional neural networks". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 64.10, pp. 1637–1639.

Zhang, Xi, Jing Li, Qiong He, Heye Zhang, and Jianwen Luo (2018). "High-Quality Reconstruction of Plane-Wave Imaging Using Generative Adversarial Network". In: *2018 IEEE International Ultrasonics Symposium (IUS)*. IEEE, pp. 1–4.

Zhou, Zixia, Yuanyuan Wang, Jinhua Yu, Yi Guo, Wei Guo, and Yanxing Qi (2018). "High Spatial–Temporal Resolution Reconstruction of Plane-Wave Ultrasound Images With a Multichannel Multiscale Convolutional Neural Network". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 65.11, pp. 1983–1996.

Perdios, Dimitris et al. (2017). "A Deep Learning Approach to Ultrasound Image Recovery". In: *IEEE International Ultrasonics Symposium*. EPFL-CONF-230991.

Yoon, Yeo Hun, Shujaat Khan, Jaeyoung Huh, and Jong Chul Ye (2017). "Deep Learning in RF Sub-sampled B-mode Ultrasound Imaging". In: *arXiv preprint arXiv:1712.06096*.

Yoon, Yeo Hun, Shujaat Khan, Jaeyoung Huh, and Jong Chul Ye (2018). "Efficient b-mode ultrasound image reconstruction from sub-sampled rf data using deep learning". In: *IEEE transactions on medical imaging* 38.2, pp. 325–336.

Yoon, Yeo Hun and Jong Chul Ye (2018). "Deep learning for accelerated ultrasound imaging". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 6673–6676.

Khan, Shujaat, Jaeyoung Huh, and Jong Chul Ye (2019b). "Universal Deep Beamformer for Variable Rate Ultrasound Imaging". In: *arXiv preprint arXiv:1901.01706*.

Khan, Shujaat, Jaeyoung Huh, and Jong Chul Ye (2019a). "Deep Learning-based Universal Beamformer for Ultrasound Imaging". In: *arXiv preprint arXiv:1904.02843*.

Vedula, Sanketh, Ortal Senouf, Grigoriy Zurakhov, Alex Bronstein, Michael Zibulevsky, Oleg Michailovich, Dan Adam, and Diana Gaitini (2018b). "High quality ultrasonic multi-line transmission through deep learning". In: *International Workshop on Machine Learning for Medical Image Reconstruction*. Springer, pp. 147–155.

Huang, Chao-Yi, Oscal Tzyh-Chiang Chen, Guo-Zua Wu, Chih-Chi Chang, and Chang-Lin Hu (2018). "Ultrasound Imaging Improved by the Context Encoder Reconstruction Generative Adversarial Network". In: *2018 IEEE International Ultrasonics Symposium (IUS)*. IEEE, pp. 1–4.

Luchies, Adam and Brett Byram (2017). "Deep neural networks for ultrasound beamforming". In: *2017 IEEE International Ultrasonics Symposium (IUS)*. IEEE, pp. 1–4.

Luchies, Adam C and Brett C Byram (2018). "Deep neural networks for ultrasound beamforming". In: *IEEE transactions on medical imaging* 37.9, pp. 2010–2021.

Luchies, Adam C and Brett C Byram (2019). "Training improvements for ultrasound beamforming with deep neural networks". In: *Physics in medicine and biology*.

Luijten, Ben, Regev Cohen, Frederik J de Bruijn, Harold AW Schmeitz, Massimo Mischi, Yonina C Eldar, and Ruud JG van Sloun (2019). "Deep Learning for Fast Adaptive Beamforming". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 1333–1337.

Vedula, Sanketh, Ortal Senouf, Grigoriy Zurakhov, Alex Bronstein, Oleg Michailovich, and Michael Zibulevsky (2018a). "Learning beamforming in ultrasound imaging". In: *arXiv preprint arXiv:1812.08043*.

Nair, Arun Asokan, Trac D Tran, Austin Reiter, and Muyinatu A Lediju Bell (2018b). "A Deep Learning Based Alternative to Beamforming Ultrasound Images". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 3359–3363.

Nair, Arun Asokan, Mardava Rajugopal Gubbi, Trac Duy Tran, Austin Reiter, and Muyinatu A Lediju Bell (2018a). "A Fully Convolutional Neural Network for Beamforming Ultrasound Images". In: *2018 IEEE International Ultrasonics Symposium (IUS)*. IEEE, pp. 1–4.

Nair, Arun Asokan, Trac D Tran, Austin Reiter, and Muyinatu A Lediju Bell (2019). "A Generative Adversarial Neural Network for Beamforming Ultrasound Images: Invited Presentation". In: *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*. IEEE, pp. 1–6.

Simson, Walter, Magdalini Paschali, Nassir Navab, and Guillaume Zahnd (2018). "Deep Learning Beamforming for Sub-Sampled Ultrasound Data". In: *2018 IEEE International Ultrasonics Symposium (IUS)*. IEEE, pp. 1–4.

Huang, Pu, Lin Su, Shuyang Chen, Kunlin Cao, Qi Song, Peter Kazanzides, Iulian Iordachita, Muyinatu A Lediju Bell, John W Wong, Dengwang Li, et al. (2019). "2D ultrasound imaging based intra-fraction respiratory motion tracking for abdominal radiation therapy using machine learning". In: *Physics in Medicine & Biology*.

Bianco, Simone, Remi Cadene, Luigi Celona, and Paolo Napoletano (2018). "Benchmark analysis of representative deep neural network architectures". In: *IEEE Access* 6, pp. 64270–64277.

Cadeddu, Jeffrey A, Andrew Bzostek, Steve Schreiner, Aaron C Barnes, William W Roberts, James H Anderson, Russell H Taylor, and Louis R Kavoussi (1997). "A robotic system for percutaneous renal access". In: *The Journal of urology* 158.4, pp. 1589–1593.

Jackson, Valerie P (1990). "The role of US in breast imaging." In: *Radiology* 177.2, pp. 305–311.

Vargas, Hernan I, M Perla Vargas, Katherine D Gonzalez, Kamal Eldrageely, and Iraj Khalkhali (2004). "Outcomes of sonography-based management of breast cysts". In: *The American journal of surgery* 188.4, pp. 443–447.

Wikland, M, C Bergh, K Borg, T Hillensjö, CM Howles, A Knutsson, L Nilsson, and M Wood (2001). "A prospective, randomized comparison of two starting doses of recombinant FSH in combination with cetrorelix in women undergoing ovarian stimulation for IVF/ICSI". In: *Human Reproduction* 16.8, pp. 1676–1681.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 234–241.

Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*.

Ioffe, Sergey and Christian Szegedy (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167*.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Rodriguez-Molares, Alfonso, Ole Marius Hoel Rindal, Olivier Bernard, Arun Nair, Muyinatu A Lediju Bell, Hervé Liebgott, Andreas Austeng, et al. (2017). "The ultrasound toolbox". In: *2017 IEEE International Ultrasonics Symposium (IUS)*. IEEE, pp. 1–4.

Jensen, Jørgen Arendt and Niels Bruun Svendsen (1992). "Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 39.2, pp. 262–267.

Jensen, Jørgen Arendt (1996). "Field: A program for simulating ultrasound systems". In: *10TH NORDICBALTIC CONFERENCE ON BIOMEDICAL IMAGING, VOL. 4, SUPPLEMENT 1, PART 1: 351–353*. Citeseer.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Hyun, Dongwoon, Leandra L Brickson, Kevin T Looby, and Jeremy J Dahl (2019a). "Beamforming and Speckle Reduction Using Neural Networks". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 66.5, pp. 898–910.

Santos, Pedro, Geir Ultveit Haugen, Lasse Løvstakken, Eigil Samset, and Jan D'hooge (2016). "Diverging wave volumetric imaging using subaperture beamforming". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 63.12, pp. 2114–2124.

Wiacek, Alycen, Ole Marius Hoel Rindal, Eniola Falomo, Kelly Myers, Kelly Fabrega-Foster, Susan Harvey, and Muyinatu A Lediju Bell (2018). "Robust Short-Lag Spatial Coherence Imaging of Breast Ultrasound Data: Initial Clinical Results". In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 66.3, pp. 527–540.

Buades, Antoni, Bartomeu Coll, and J-M Morel (2005). "A non-local algorithm for image denoising". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 2. IEEE, pp. 60–65.

Coupe, P., P. Hellier, C. Kervrann, and C. Barillot (2009). "Nonlocal means-based speckle filtering for ultrasound images". In: *IEEE Transactions on Image Processing* 18.10, pp. 2221–2229. ISSN: 1941-0042. DOI: 10.1109/TIP.2009.2024064.

Gomez, W, L Leija, WCA Pereira, and AFC Infantosi (2009). "Morphological operators on the segmentation of breast ultrasound images". In: *2009 Pan American Health Care Exchanges*. IEEE, pp. 67–71.

Luo, Yaozhong, Longzhong Liu, Qinghua Huang, and Xuelong Li (2017). "A novel segmentation approach combining region-and edge-based information for ultrasound images". In: *BioMed research international* 2017.

Zou, Kelly H, Simon K Warfield, Aditya Bharatha, Clare MC Tempany, Michael R Kaus, Steven J Haker, William M Wells III, Ferenc A Jolesz, and Ron Kikinis (2004). "Statistical validation of image segmentation quality based on a spatial overlap index1: scientific reports". In: *Academic radiology* 11.2, pp. 178–189.

Rodriguez-Molares, Alfonso, Ole Marius Hoel Rindal, Jan D'hooge, Svein-Erik Måsøy, Andreas Austeng, Muyinatu A Lediju Bell, and Hans Torp (2019). "The generalized contrast-to-noise ratio: a formal definition for lesion detectability". In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*.

Hyun, Dongwoon, You Leo Li, Idan Steinberg, Marko Jakovljevic, Tal Klap, and Jeremy J Dahl (2019b). "An Open Source GPU-Based Beamformer for Real-Time Ultrasound Imaging and Applications". In: *2019 IEEE International Ultrasonics Symposium (IUS)*. IEEE, pp. 20–23.

Wagner, Robert F (1983). "Statistics of speckle in ultrasound B-scans". In: *IEEE Trans. Sonics & Ultrason.* 30.3, pp. 156–163.

Burckhardt, Christoph B (1978). "Speckle in ultrasound B-mode scans". In: *IEEE Transactions on Sonics and ultrasonics* 25.1, pp. 1–6.

Coupé, Pierrick, Pierre Hellier, Charles Kervrann, and Christian Barillot (2009). "Nonlocal means-based speckle filtering for ultrasound images". In: *IEEE transactions on image processing* 18.10, pp. 2221–2229.

Yu, Yongjian and Scott T Acton (2002). "Speckle reducing anisotropic diffusion". In: *IEEE Transactions on image processing* 11.11, pp. 1260–1270.

Papernot, Nicolas, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami (2016). "Distillation as a defense to adversarial perturbations against deep neural networks". In: *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, pp. 582–597.

Glocker, Ben, Nikos Komodakis, Nikos Paragios, Christian Glaser, Georgios Tziritas, and Nassir Navab (2007). "Primal/dual linear programming and statistical atlases for cartilage segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 536–543.

Yuille, Alan L and Chenxi Liu (2018). "Deep Nets: What have they ever done for Vision?" In: *arXiv preprint arXiv:1805.04025*.

Feigin, Micha, Daniel Freedman, and Brian W Anthony (2018). "A Deep learning framework for Single sided sound speed inversion in medical ultrasound". In: *arXiv preprint arXiv:1810.00322*.

Brickson, Leandra L, Dongwoon Hyun, and Jeremy J Dahl (2018). "Reverberation Noise Suppression in the Aperture Domain Using 3D Fully Convolutional Neural Networks". In: *2018 IEEE International Ultrasonics Symposium (IUS)*. IEEE, pp. 1–4.

Simson, Walter, Rüdiger Göbl, Magdalini Paschali, Markus Krönke, Klemens Scheidhauer, Wolfgang Weber, and Nassir Navab (2019). "End-to-End Learning-Based Ultrasound Reconstruction". In: *arXiv preprint arXiv:1904.04696*.

Wu, Sitong, Zhifan Gao, Zhi Liu, Jianwen Luo, Heye Zhang, and Shuo Li (2018). "Direct reconstruction of ultrasound elastography using an end-to-end deep neural network". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 374–382.

Vedula, Sanketh, Ortal Senouf, Alex M Bronstein, Oleg V Michailovich, and Michael Zibulevsky (2017). "Towards CT-quality Ultrasound Imaging using Deep Learning". In: *arXiv preprint arXiv:1710.06304*.

Tom, Francis and Debdoot Sheet (2018). "Simulating patho-realistic ultrasound images using deep generative networks with adversarial learning". In: *Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on*. IEEE, pp. 1174–1177.

Hu, Yipeng, Eli Gibson, Li-Lin Lee, Weidi Xie, Dean C Barratt, Tom Vercauteren, and J Alison Noble (2017). "Freehand ultrasound image simulation with spatially-conditioned generative adversarial networks". In: *Molecular imaging, reconstruction and analysis of moving body organs, and stroke imaging and treatment*. Springer, pp. 105–115.

Chapter 6

Radar Signal Enhancement using Deep Learning

In this chapter, we demonstrate how deep learning can be used to improve signal quality in radar, specifically a type of radar called ultra-wideband (UWB) radar, by denoising the raw received signal in each sensor element prior to beamforming. Modern UWB radar systems transmit a wide range of frequencies, spanning hundreds of MHz to a few GHz, to achieve improved penetration depth and narrower pulse width. A common challenge faced is the presence of other commercial transmission equipment operating in the same band, causing radio frequency interference (RFI). To overcome this RFI issue, radar systems have been developed to either avoid operating in bands with RFI or suppress the RFI after reception. In this work, we examine both families of operation and demonstrate that 1D convolutional neural networks based on the UNet architecture can provide powerful signal enhancement capabilities on raw UWB radar data. The model is *trained purely on simulated data* and translated to real UWB data, achieving impressive results compared to traditional sparse-recovery baseline algorithms. The work in this chapter

has been accepted for publication in Nair et al., 2021.

6.1 Introduction

Ultra-wideband (UWB) radar systems have gained significant traction due to their superior penetration capability and improved imaging resolution (Taylor, 2012). The U.S. Army, for example, has been developing UWB radar systems for detection of difficult targets in foliage penetration (Nguyen, Kapoor, and Sichina, 1997), ground penetration (Nguyen et al., 1998), and sensing-through-the-wall (Nguyen, Ressler, and Sichina, 2008). For superior penetration ability, these systems must operate in the low-frequency spectrum that spans from under 100 MHz to several GHz.

As well as requiring low frequency operation for penetration, synthetic aperture radar (SAR) obtains high resolution images by transmitting pulses with UWB — the wider the pulse bandwidth in frequency, the narrower the pulse in time, improving spatial resolution (Taylor, 2012; Carin et al., 1999; Soumekh, 1999). However, the transmission of UWB pulses is often complicated by the presence of other communication equipment sharing the same spectrum. UWB radar signals span a wide spectrum that also includes radio, TV, cellular phones, and other communication systems, each of which inject radio frequency interference (RFI) into the data.

This leaves the radar system with two approaches to solve the problem. The first is to continue to transmit in those bands and denoise the RFI-contaminated radar data after reception (Miller, Potter, and McCorkle, 1997; Nguyen and Tran, 2015). The second is to employ stepped-frequency

radars (SFRs) (Nguyen and Park, 2016; al., 2017a; al., 2017b) with frequency hopping capabilities. SRFs allow for the transmission of UWB pulses while still maintaining precise control over the transmitted spectrum, utilizing frequency synthesizers that can be configured to avoid transmitting energy in prohibited/interference frequency bands. Unfortunately, notches in the frequency domain caused by this transmission method create strong sidelobes (or ringing artifacts) in the received time domain data, which requires further signal processing to ameliorate.

One could argue that the second approach (spectral gap extrapolation) partially subsumes the first approach (RFI suppression) — as one can always suppress frequency components where there is heavy RFI by setting those Fourier coefficients to zero and then proceeding to extrapolate the resultant spectral gaps. However, this line of thinking has two major problems. First, it is assumes that the operating spectrum affected by RFI is known exactly, or else unnecessary performance degradation will be introduced. Second, there are often better performing pre-processing methods than notching RFI-affected radar data. Thus, it is better to deal with each scenario separately.

Sparsity-based signal processing methods have achieved great success in both suppressing RFI (Nguyen and Tran, 2016; Song et al., 2018) and performing spectral gap extrapolation (Cetin and Moses, 2005; Nguyen and Do, 2012; Nguyen, Tran, and Do, 2014) to combat frequency notches. However, they still struggle to distinguish neighboring and/or weak targets at fine resolution and performance drops precipitously when the RFI bands (or notches) are wider or affect more frequencies.

Deep neural networks (DNNs), and deep convolutional neural networks (CNNs) in particular, have recently become immensely popular for a wide variety of traditional signal processing tasks like image segmentation (Ronneberger, Fischer, and Brox, 2015), denoising (Zhang et al., 2017), and point source localization in the presence of noise (Allman, Reiter, and Bell, 2018), displaying extremely impressive results. In the radar domain, DNNs have been successfully used for target detection and classification (Brodeski, Bilik, and Giryes, 2019), antenna selection in cognitive radar (Elbir, Mishra, and Eldar, 2019), interference mitigation (Mun, Kim, and Lee, 2018) and vehicle detection (Major, 2019) in automotive applications, and activity recognition (Gurbuz and Amin, 2019; Jokanovic, Amin, and Ahmad, 2016; Seyfioğlu, Özbayoğlu, and Gürbüz, 2018) applications in indoor monitoring. For SAR specifically, image despeckling (Zhang et al., 2020), phase error correction (Mason, Yonel, and Yazici, 2017), change detection (Gong et al., 2015), ship detection (Deng et al., 2019) and discrimination (Schwegmann et al., 2016), and image reconstruction (Yonel, Mason, and Yazıcı, 2017; Thammakhoune and Yavuz, 2020) are just some of the problems where deep learning has helped.

Past work from our group (Tran, Tran, and Nguyen, 2018; Nguyen, Tran, and Tran, 2019) investigated the use of a specific kind of deep neural network, called a generative adversarial network (GAN) (Goodfellow et al., 2014), to perform spectral gap extrapolation and obtained promising results. In this work, we expand upon the prior work in three important ways. First, in addition to spectral gap extrapolation, we also demonstrate successful RFI suppression using a deep network. Second, we demonstrate state-of-the-art

results on real UWB SAR data. Third, we demonstrate this success via a simple 1D CNN based on the UNet (Ronneberger, Fischer, and Brox, 2015) architecture, which is easier and more stable to train than a GAN.

6.2 Method

The goal of this work is to successfully recover clean raw UWB SAR data, $\mathbf{x}$, from noisy observations, $\mathbf{y}$, observed by sensors. Specifically, we consider three kinds of noise: *(i)* RFI, where the majority of the energy of the interfering signal is located in a few frequency bands; *(ii)* random spectral gaps, where several randomly chosen narrow spectral bands are missing; and *(iii)* a block spectral gap, where a single contiguous segment of the operating spectrum is missing. We investigate the efficacy of using 1D UNet (Ronneberger, Fischer, and Brox, 2015) CNNs to remove each kind of noise – *a different 1D UNet is trained for each noise type* – and compare it with competitive baselines. The UNet is trained end-to-end (i.e., all layers are learned simultaneously).

6.2.1 Ground-truth Dataset for Network Training

To create the clean training data, a sparsity-based linear model widely employed in compressed sensing SAR (Nguyen, Tran, and Do, 2014) was used. The scene of interest was modeled as a sparse collection of independent point scatterers randomly distributed in space. As the model is linear, it is assumed that the scatterers do not interact with each other, and the final received signal is simply the sum of reflections from each of the individual scatterers.

Mathematically, the model can be expressed as

$$x(t) = \sum_i r(z_i) p(t; z_i) \tag{6.1}$$

where $x(t)$ is the received raw SAR signal, $r(z_i)$ is the reflectivity of a point scatterer located at z_i, and $p(t; z_i)$ represents the point spread function of a scatterer with unit reflectivity located at z_i.

To implement (6.1), the template pulse $p(t; 0)$ is linearly shifted to represent the response from various locations $p(t; z)$, and the shifted pulses are stored as columns of a dictionary $\mathbf{P}$. Simulating data comes down to sampling possible sparse code coefficients, $\mathbf{r}$, to combine with the dictionary to yield the received raw data

$$\mathbf{x} = \mathbf{P}\mathbf{r}. \tag{6.2}$$

An advantage of this modeling approach over Nguyen, Tran, and Tran, 2019 is that we operate on the received 1D data from each aperture element individually, i.e., the geometry of the entire aperture does not matter — a neural network trained on one geometry can generalize to another. The actual image creation (slow-time processing) is accomplished later.

The template pulse is sampled at 37.48 GHz and contains most of its energy, as measured by the -12 dB points, between 380 and 2080 MHz. We set the signal dimension (i.e., the lengths of $\mathbf{x}$ and $\mathbf{r}$) to a fixed value of 1024 samples.

A total of 1,000,000 possible sparse codes were sampled from realistic sparse code distributions mimicking coefficients obtained in side-looking SAR to construct the ground-truth training dataset. An additional 12,500 samples (of course, with no intersection with the training set) were generated and

reserved as a clean simulated test set. Lastly, 3,600 samples corresponding to two real data acquisitions of 1,800 samples each from circular-sensing SAR were reserved as a clean real test set. Each of these were then corrupted with noise to generate paired clean+noisy data, as detailed in Section 6.2.2.

6.2.2 Noise Modeling

In this work, we focus on three kinds of noise – RFI, random spectral gaps, and a block spectral gap. Below, we provide details on each and elaborate on their modeling.

6.2.2.1 Radio Frequency Interference

The scenario of RFI occurs when an interfering source transmits most of its energy in a small subset of the spectrum of the UWB SAR. Mathematically, this can be modeled as an additive noise:

$$\mathbf{y}_{int} = \mathbf{x} + \mathbf{i} \tag{6.3}$$

where $\mathbf{i}$ is the RFI signal and $\mathbf{y}_{int}$ represents the observed noisy data.

The RFI, $\mathbf{i}$, used in this work is obtained from real RFI data recorded over a long time horizon. We split the recorded RFI signal into two parts — we use samples from the first half to generate training data and samples from the second half to generate test data. For each set, we mix a randomly chosen clean signal and RFI samples at various signal-to-noise ratios (SNRs) randomly chosen from -15, -10, -5, 0, 5, and 10 dB.

6.2.2.2 Random Spectral Gaps

The scenario of random spectral gaps occurs when several narrowband sections of the radar spectrum might be restricted and off-limits to data transmission. Mathematically, this can be modeled as a masking operation in the Fourier domain:

$$FFT(\mathbf{y}_{rg}) = \mathbf{m}_{rg} \odot FFT(\mathbf{x}) \tag{6.4}$$

where $\mathbf{m}_{rg}$ is a binary mask. The total signal bandwidth is divided into 10 narrow spectral bands and depending on the missing percentage, several bands are masked to zero, while the mask affecting the remaining coefficients is one. Here, $\mathbf{y}_{rg}$ represents the observed noisy data suffering from random spectral gaps.

Noisy data corresponding to spectral missing percentages of 50%, 60%, 70%, 80%, and 90% were generated for use in training and testing by randomly choosing and eliminating the chosen percentage of spectral coefficients from the ground-truth data.

6.2.2.3 Block Spectral Gap

The scenario of a centered block spectral gap occurs as the worst-case scenario when a contiguous section of the radar spectrum centered on the middle of the transmitted template pulse's bandwidth (where most of the pulse's energy is located) is marked as restricted and not allowed for transmission. Mathematically, this too can be modeled as a masking operation in the Fourier domain:

$$FFT(\mathbf{y}_{bg}) = \mathbf{m}_{bg} \odot FFT(\mathbf{x}) \tag{6.5}$$

where $\mathbf{m}_{bg}$ is a binary mask of zeros and ones determining which spectral coefficients are transmitted and which are not available. Unlike $\mathbf{m}_{rg}$ where the zeros are chosen to lie randomly in several narrow spectral gaps, in $\mathbf{m}_{bg}$ the vanishing region is located contiguously around the center frequency of the pulse. We use $\mathbf{y}_{bg}$ to represent the observed noisy data suffering from the centered block spectral gap.

Noisy data corresponding to spectral missing percentages of 50%, 60%, 70%, 80%, and 90% were generated for use in training and testing by setting to zero the chosen percentage of spectral coefficients of the clean data.

6.2.3 Neural Network Details

The network architecture used in this study is an adaptation of the popular UNet (Ronneberger, Fischer, and Brox, 2015) architecture adapted to the 1D signal processing scenario. A visualization of its structure with the number of filters in each layer is presented in Fig. 6.1. It has a fully convolutional encoder-decoder type architecture, with a total of 20 layers – 10 layers each in the encoder and decoder. Convolutional kernel size is set to 5, with encoder layers having a stride of 2 to downsample the feature map in each layer (except for the input layer, which has a stride of 1). The decoder layers all have a stride of 2 to upsample the feature map at each layer. Skip connections are employed to connect encoder and decoder layers at the same level. Each layer uses BatchNorm (BN) (Ioffe and Szegedy, 2015) and LeakyReLU as its nonlinearity components (except for the output layer, which has neither). Sub-pixel convolutions, also known as PixelShuffle (Odena, Dumoulin, and

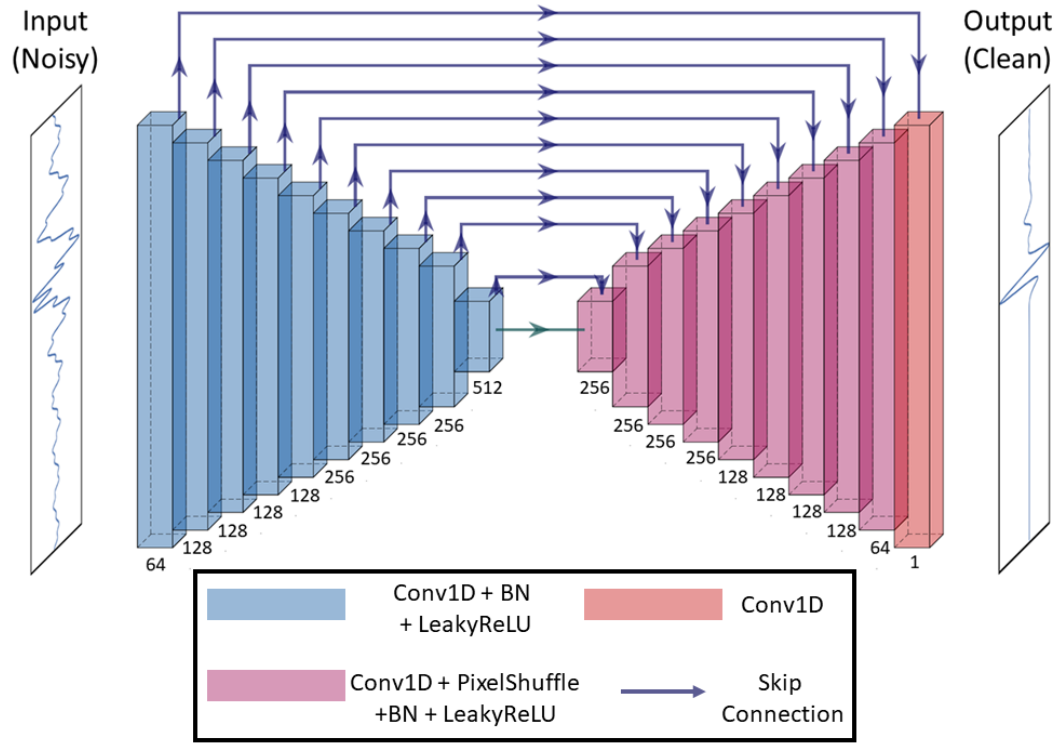


Figure 6.1: Proposed 1D UNet for UWB signal denoising. Noisy input data degraded by one of RFI, random spectral gaps, or a centered block spectral gap is denoised by the network trained on that noise type to yield an estimate of the clean target signal.

Olah, 2016; Shi et al., 2016), are used in the decoder as they seem to work better than transposed convolutions and they reduce recovery artifacts. The total number of trainable parameters in the network is 7,182,209.

A different network was trained for each noise type, but in each case, a single network was trained to denoise all noise conditions for the chosen noise type. All networks were trained with *L1Loss*, or mean absolute error, as the loss criterion, using the Adam optimizer (Kingma and Ba, 2014) with a batch size of 64 and a learning rate of 0.0001.

6.2.4 Baselines

For RFI suppression, we implement a simple but effective frequency masking baseline. As most of the energy of the bandlimited RFI signal was observed to be between 256 and 476 MHz, the Fourier coefficients of the noisy input data between those limits were set to zero to yield a baseline enhanced signal with which to compare our neural network approach.

For spectral gap extrapolation on signals with random spectral gaps, we implement a sparse coding baseline via Orthogonal Matching Pursuit (OMP) (Tropp and Gilbert, 2007) (or any of its variants such as Varadarajan, Khudanpur, and Tran, 2011), following the approach proposed in Nguyen and Do, 2012; Nguyen, Tran, and Do, 2014. Similar to the way the dictionary $\mathbf{P}$ in (6.2) was constructed, a dictionary $\tilde{\mathbf{P}}$ was constructed by linearly shifting a corrupted transmitted pulse (possessing the same random spectral gap structure as the noisy data). Every random spectral gap structure encountered required its own tailored dictionary. Sparse coding was performed on the noisy input data using this corrupted dictionary. Assuming robust sparse codes, we obtained the recovered signal from the clean dictionary $\mathbf{P}$. The number of sparse coefficients in the OMP algorithm, K, was tuned on the test data itself. While this is not possible in practice, it does yield the best possible performance for the baseline algorithm.

For spectral gap extraction on signals with a centered contiguous gap, a sparse coding baseline very similar to the one implemented for the random spectral gaps was used. Here, since the gap structure remains the same for all data at a specific missing percentage, the corrupted dictionary with

linearly shifted corrupted template pulses $\overline{\mathbf{P}}$ could be shared. Sparse coding was performed on the noisy input data using this corrupted dictionary, and the sparse code thus obtained was combined with the clean dictionary $\mathbf{P}$ to yield the baseline enhanced signal. Again, we manually tune the OMP hyperparameter, K, tuned on the test data itself to obtain the best performance for comparison.

6.2.5 Evaluation

Quantitative evaluation of denoising performance is carried out with the general purpose SNR metric reported in the dB scale. It was measured as

$$\text{SNR}(\mathbf{x}, \mathbf{z}) = 20 \log_{10} \frac{||\mathbf{x}||_2}{||\mathbf{x} - \mathbf{z}||_2} \tag{6.6}$$

where $\mathbf{x}$ is the target clean signal, $\mathbf{z}$ is the signal being compared to it, and $||.||_2$ is the ℓ_2 norm.

6.3 Experiments

6.3.1 Radio Frequency Interference

Fig. 6.2 shows quantitative comparisons between the output SNR (in dB) of the UNet-based approach and the baseline approach that sets the Fourier coefficients affected strongly by RFI to zero for various input SNR values. It is observed that the UNet approach consistently outperforms the baseline on both the simulated and real test data for all input SNR values, delivering an average SNR gain (averaged over all input SNR values) of 25.5 and 21.87 dB on simulated and real data, respectively. In contrast, the baseline only yields

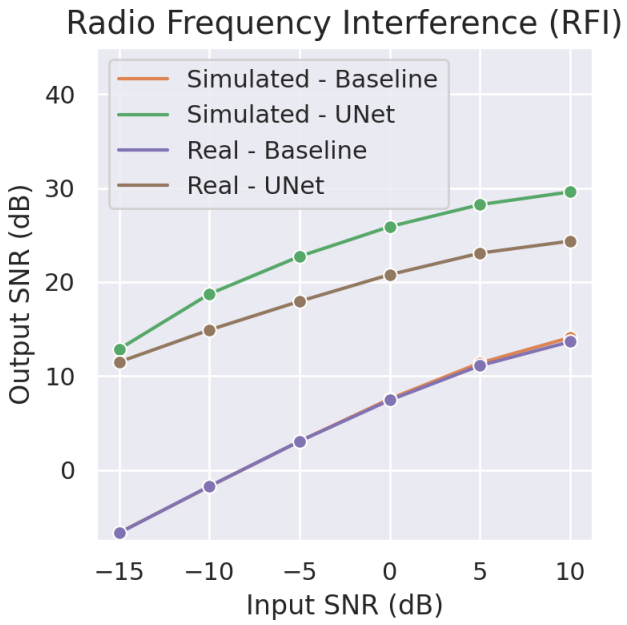


Figure 6.2: Simulated and real RFI affected data are enhanced by the baseline and the UNet. Enhancement performance at different noise levels is studied by plotting output SNR as a function of input SNR for the baseline enhanced data and the UNet enhanced data.

an average simulated and real SNR gain of 7.1 and 7.0 dB, respectively.

To visualize these results, Fig. 6.3 shows (from top-left to bottom-right) (a) clean target data, (b) noisy input data corrupted by RFI, (c) baseline enhanced output data, and (d) enhanced output data obtained from the UNet. All images are plotted with a dynamic range of 40 dB. The specific example displayed here is the real test data in the most challenging scenario when RFI is very strong (input SNR is -15dB). As a result, the target structure is barely visible in the noisy input data shown in Fig. 6.3 (b). The baseline algorithm enhances the image slightly, but the UNet does significantly better, efficiently exploiting the structure in the RFI signal and suppressing it.

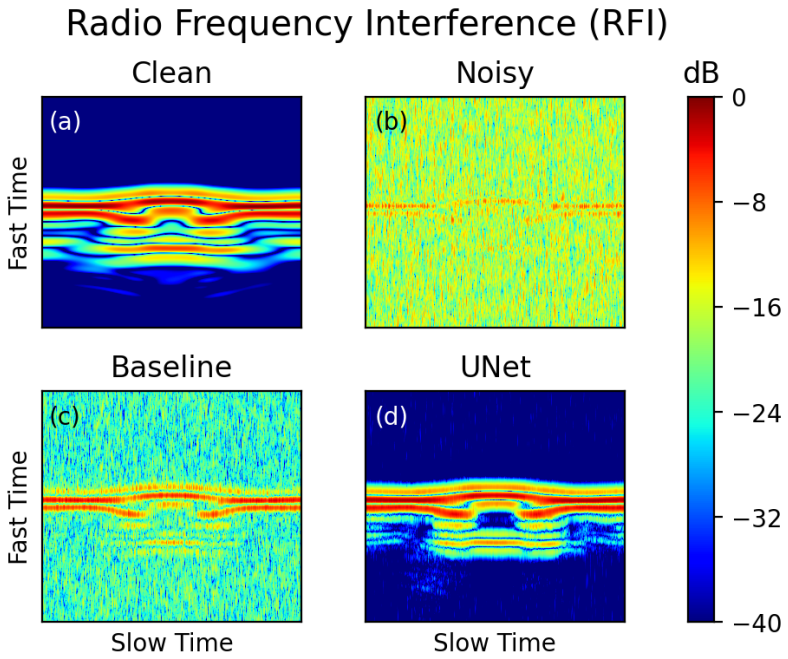


Figure 6.3: Visualization of real data denoising under challenging RFI noise conditions. (a) is the clean target data, (b) the noisy data suffering from RFI with an SNR of -15 dB, (c) the enhanced output from the baseline method, and (d) the enhanced output from the UNet trained to tackle RFI.

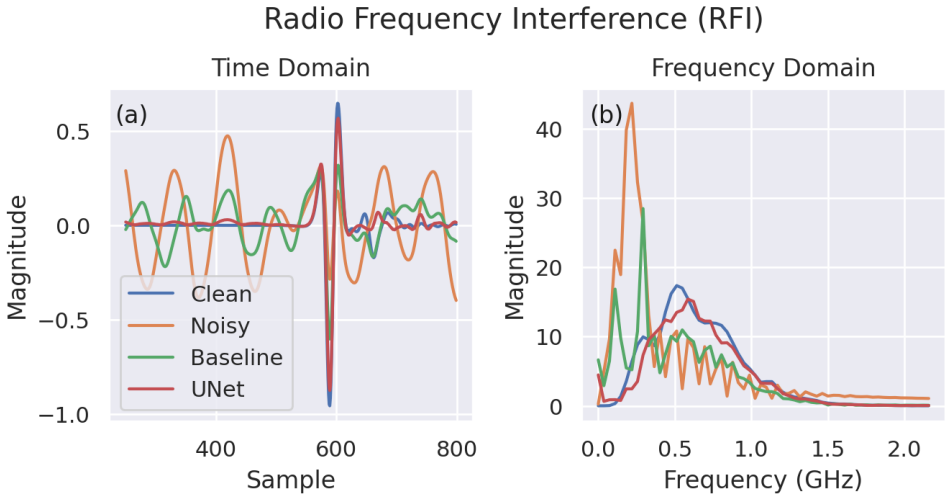


Figure 6.4: A single representative aperture element is chosen from Fig. 6.3 and the radar waveforms corresponding to clean, noisy, baseline enhanced, and UNet enhanced data are plotted for RFI noise in (a) with the corresponding magnitude spectra plotted in (b).

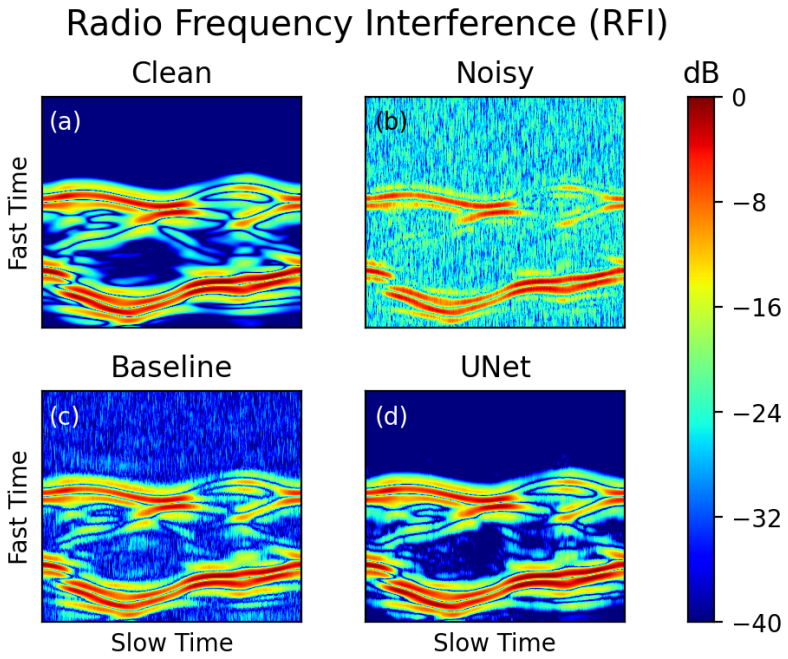


Figure 6.5: Visualization of real data denoising under milder RFI noise conditions. (a) is the clean target data, (b) the noisy data suffering from RFI with an SNR of 0 dB, (c) the enhanced output from the baseline method, and (d) the enhanced output from the UNet trained to tackle RFI.

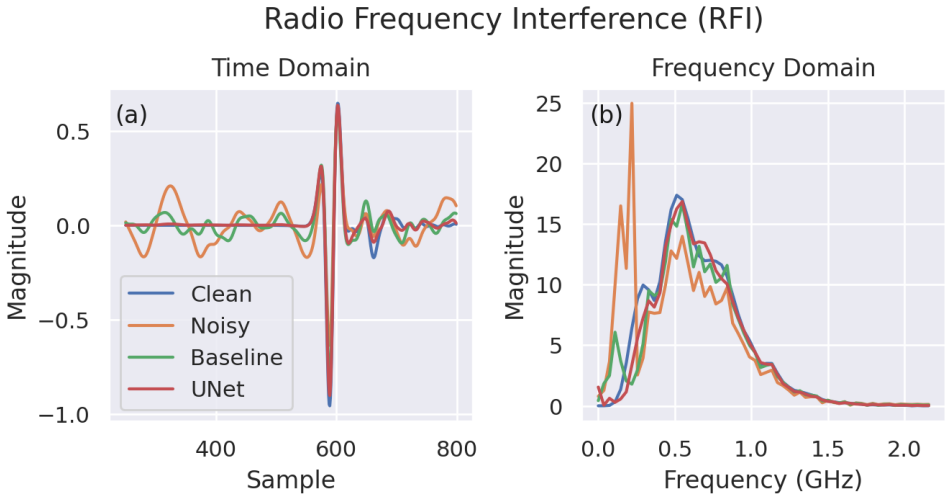


Figure 6.6: A single representative aperture element is chosen from Fig. 6.5 and the radar waveforms corresponding to clean, noisy, baseline enhanced, and UNet enhanced data are plotted for RFI noise in (a) with the corresponding magnitude spectra plotted in (b).

We study the enhancement in more detail by plotting the 1D radar waveforms received by a single representative aperture element from Fig. 6.3 in Fig. 6.4 (a). It is clear here too that the UNet does a better job suppressing the RFI and recovering the shape of the target pulse. This is confirmed again when examining the corresponding magnitude spectra in Fig. 6.4 (b).

Figs. 6.5 and 6.6 contain similar RFI denoising results for the second real dataset under milder noise.

6.3.2 Random Spectral Gaps

Fig. 6.7 plots the SNRs (in dB) versus the missing spectrum percentage for noisy input affected by random spectral gaps, baseline enhanced output, and UNet enhanced output, on both simulated and real test data. The UNet approach consistently performs as well as or better than the baseline on both simulated and real test data for all input SNR values, delivering an average SNR gain (averaged over all input SNR values) of 22.75 and 10.19 dB on simulated and real data, respectively. In contrast, the baseline only yields an average simulated and real SNR gain of 10.40 and 6.76 dB, respectively.

The network output SNR here on real data is lower than the case of RFI because we train our networks on simulated data and there is a domain shift between the training data and test data that negatively impacts network performance, which is especially impactful when the noise is signal-dependent like random spectral gaps. Thus, it is important to make our training data as representative of real test data as possible. This is the major current bottleneck to further improvements within this framework.

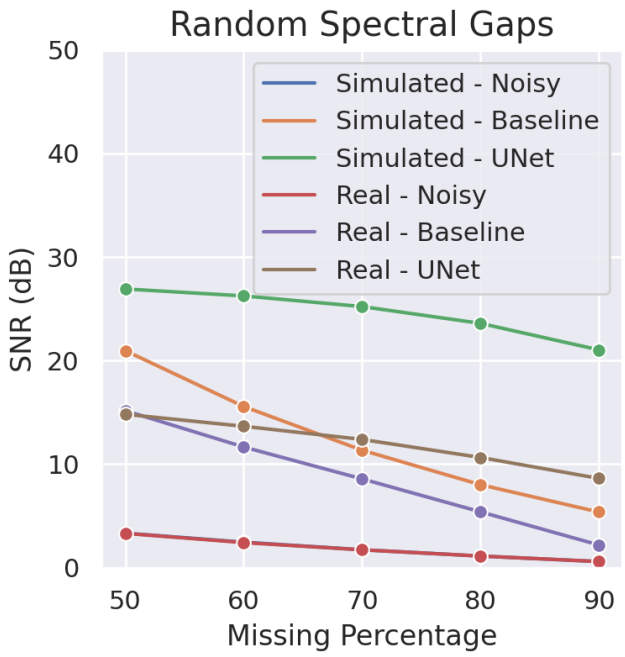


Figure 6.7: Simulated and real data suffering from random spectral gaps are enhanced by the baseline and the UNet. Enhancement performance at different noise levels is studied by plotting output SNR as a function of missing spectrum percentage for the input noisy data, the baseline enhanced data, and the UNet enhanced data.

Fig. 6.8 shows (from top-left to bottom-right) (a) clean target data, (b) noisy input data corrupted by random spectral gaps setting 90% of the spectrum to zero, (c) enhanced output data obtained from the OMP baseline, and (d) enhanced output data obtained from the UNet. The UNet does well in recovering the target clean data, outperforming the baseline in this severe noise condition and recovering the target structures. The radar signals recorded by a single representative aperture and its magnitude spectra can be observed in Fig. 6.9 (a) and (b), respectively.

Figs. 6.10 and 6.11 contain similar random spectral gap extrapolation results for the second real dataset under milder noise.

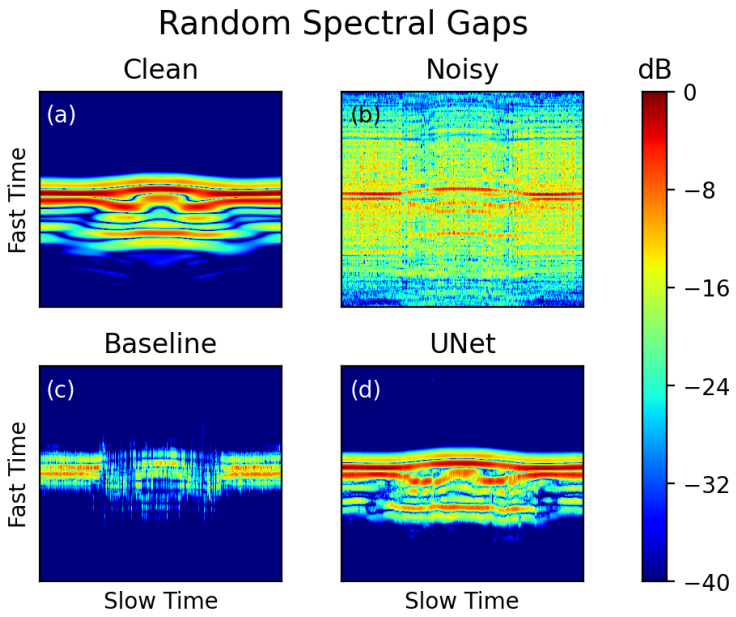


Figure 6.8: Visualization of real data denoising under challenging random spectral gaps noise conditions. (a) is the clean target data, (b) the noisy data suffering from random spectral gaps with a spectral missing percentage of 90%, (c) the enhanced output from the baseline method, and (d) the enhanced output from the UNet trained to tackle random spectral gaps.

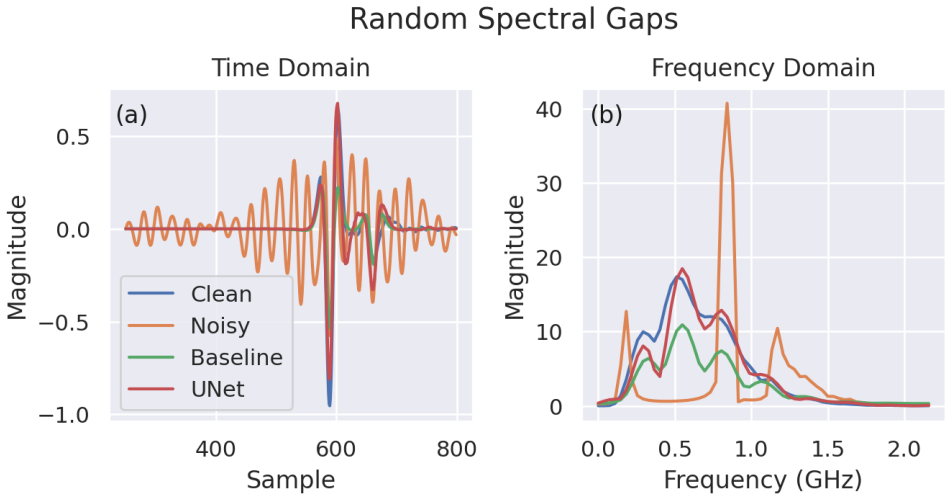


Figure 6.9: A single representative aperture element is chosen from Fig. 6.8 and the radar waveforms corresponding to clean, noisy, baseline enhanced, and UNet enhanced data are plotted for random spectral gaps noise in (a) with the corresponding magnitude spectra plotted in (b).

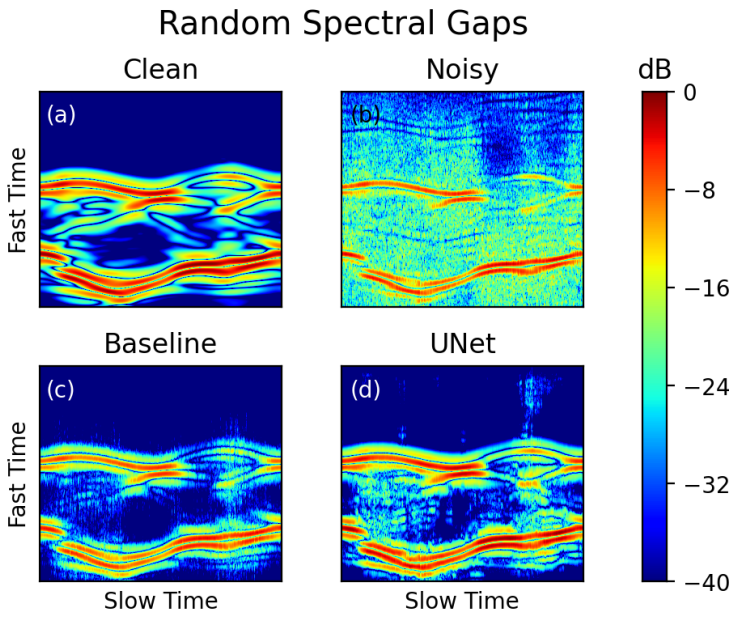


Figure 6.10: Visualization of real data denoising under milder random spectral gaps noise conditions. (a) is the clean target data, (b) the noisy data suffering from random spectral gaps with a spectral missing percentage of 50%, (c) the enhanced output from the baseline method, and (d) the enhanced output from the UNet trained to tackle random spectral gaps.

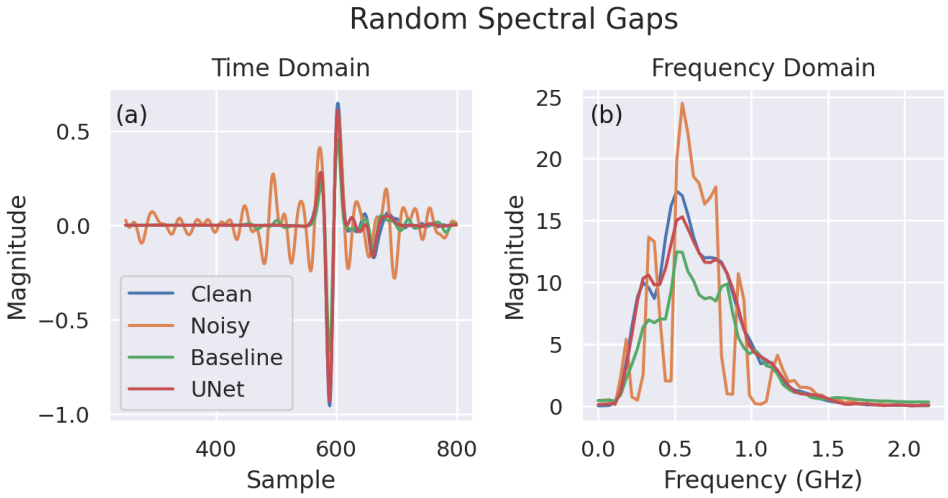


Figure 6.11: A single representative aperture element is chosen from Fig. 6.10 and the radar waveforms corresponding to clean, noisy, baseline enhanced, and UNet enhanced data are plotted for random spectral gaps noise in (a) with the corresponding magnitude spectra plotted in (b).

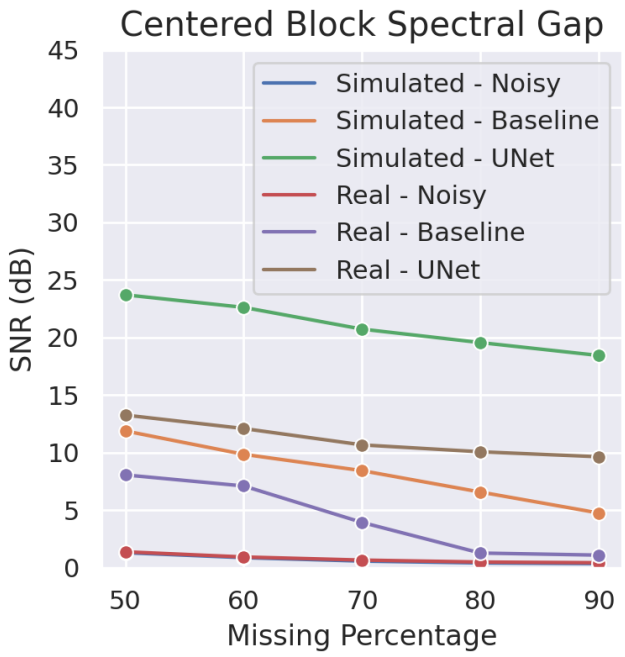


Figure 6.12: Simulated and real data suffering from a centered block spectral gap are enhanced by the baseline and the UNet. Enhancement performance at different noise levels is studied by plotting output SNR as a function of missing spectrum percentage for the input noisy data, the baseline enhanced data, and the UNet enhanced data.

6.3.3 Centered Block Spectral Gap

Fig. 6.12 plots SNR (in dB) versus missing spectrum percentage for noisy input data affected by a centered block spectral gap, baseline enhanced output, and UNet enhanced output, on both simulated and real test data. The UNet-based approach outperforms the baseline OMP approach on all missing percentages on both simulated and real data, yielding a SNR gain of 20.31 and 10.37 dB, respectively, compared to 7.60 and 3.51 dB, respectively. The network output SNR here on real data though is lower than the case of RFI due to the same data domain shift as elaborated on in Section 6.3.2.

Fig. 6.13 shows (from top-left to bottom-right) (a) clean target data, (b)

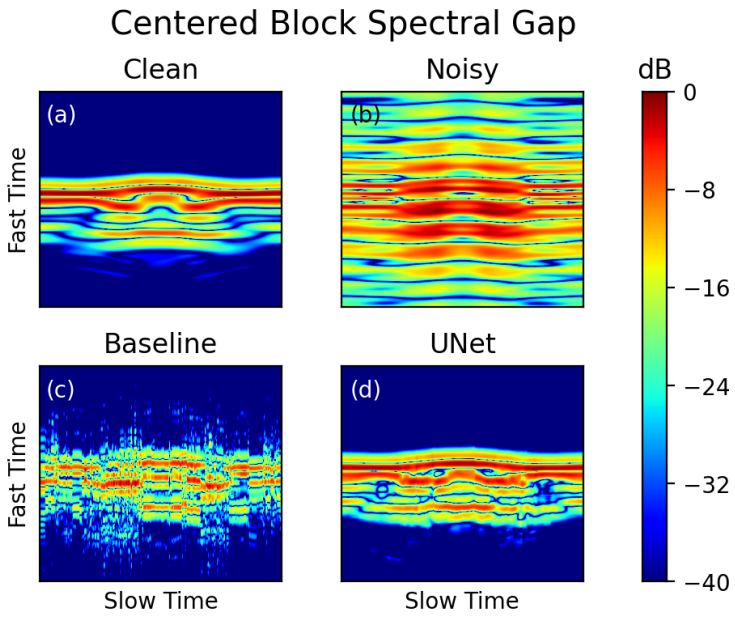


Figure 6.13: Visualization of real data denoising under challenging centered spectral gap noise conditions. (a) is the clean target data, (b) the noisy data suffering from a centered block spectral gap with a spectral missing percentage of 90%, (c) the enhanced output from the baseline method, and (d) the enhanced output from the UNet trained to tackle the centered block spectral gap.

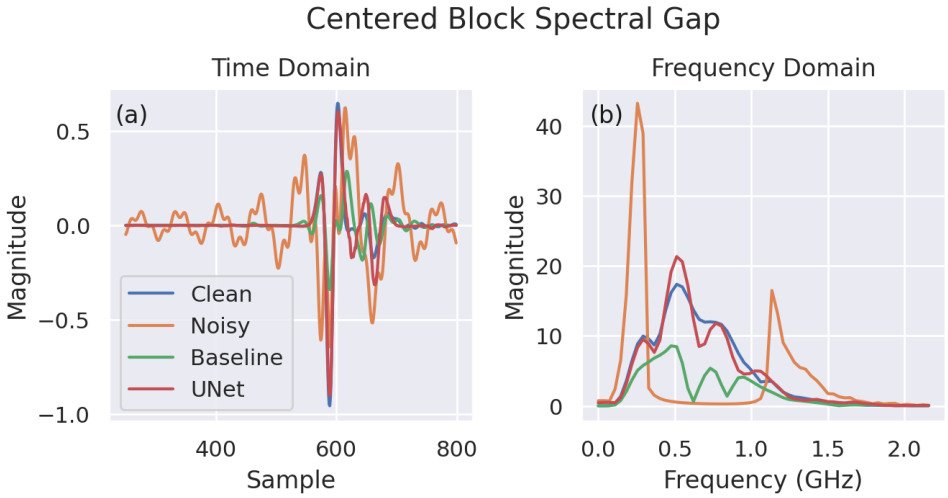


Figure 6.14: A single representative aperture element is chosen from Fig. 6.13 and the radar waveforms corresponding to clean, noisy, baseline enhanced, and UNet enhanced data are plotted for centered spectral gap noise in (a) with the corresponding magnitude spectra plotted in (b).

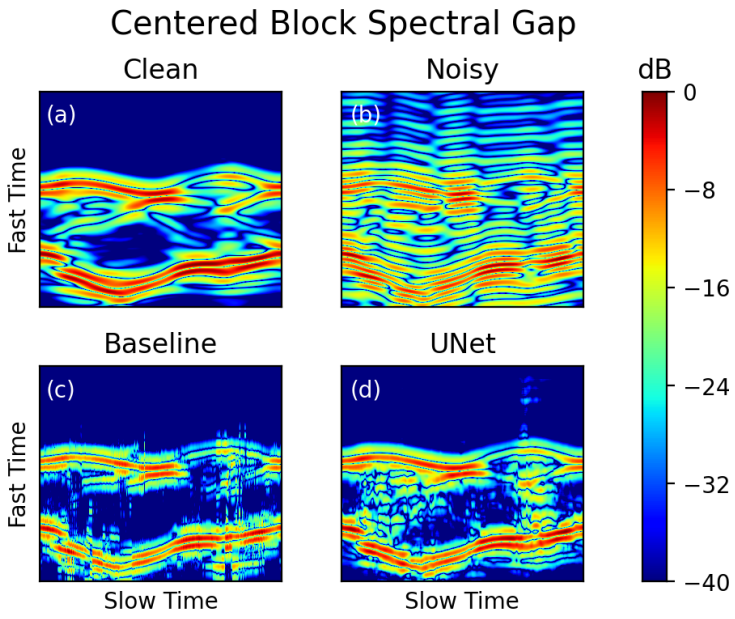


Figure 6.15: Visualization of real data denoising under milder centered spectral gap noise conditions. (a) is the clean target data, (b) the noisy data suffering from a centered block spectral gap with a spectral missing percentage of 50%, (c) the enhanced output from the baseline method, and (d) the enhanced output from the UNet trained to tackle the centered block spectral gap.

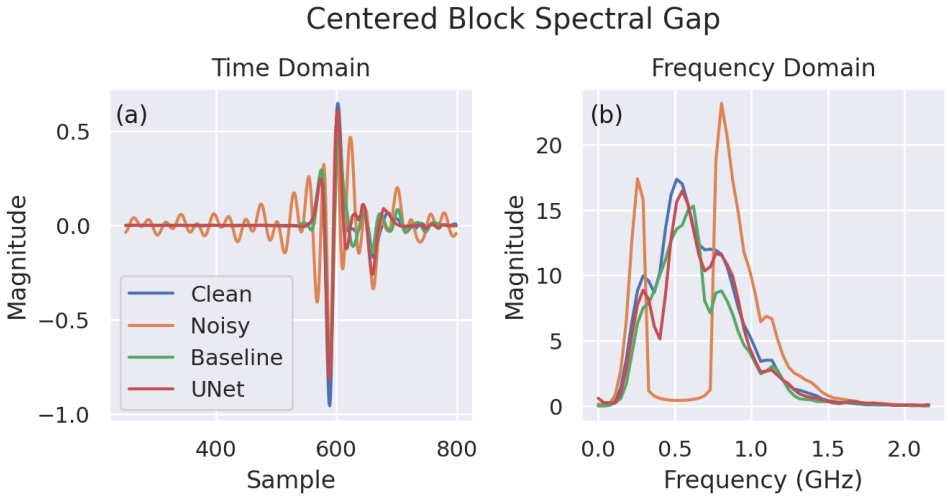


Figure 6.16: A single representative aperture element is chosen from Fig. 6.15 and the radar waveforms corresponding to clean, noisy, baseline enhanced, and UNet enhanced data are plotted for centered spectral gap noise in (a) with the corresponding magnitude spectra plotted in (b).

noisy input data corrupted by a centered block spectral gap setting 90% of the spectrum to zero, (c) enhanced output data obtained from the OMP baseline, and (d) enhanced output data obtained from the UNet. The UNet does well, largely eliminating ringing artifacts and recovering target structural information better than OMP. This observation is confirmed by studying closely the radar signals recorded by a single representative aperture and its magnitude spectra in Fig. 6.14 (a) and (b), respectively.

Figs. 6.15 and 6.16 contain similar block spectral gap extrapolation results for the second real dataset under milder noise.

6.4 Conclusion

In this work, we demonstrated the efficacy of using 1D UNet networks to address three types of noise widely encountered by a UWB SAR – bandlimited RFI, random spectral gaps, and a contiguous block spectral gap, with the networks – one trained for each noise type – achieving good results even in challenging scenarios and displaying the recovery robustness at multiple noise levels. We trained our model purely on simulated data generated by a simple sparse linear model and demonstrated the network's remarkable generalization to real test data. Since our approach operates on individual data apertures, one key benefit is that the test sensor geometry is no longer required to match the training sensor geometry. In other words, our approach is less scene-dependent. In fact, we trained our networks using synthetically generated data on a side-looking geometry and successfully tested our networks on raw SAR data collected from a circular 360°-sensing geometry.

References

Nair, Arun Asokan, Akshay Rangamani, Lam H Nguyen, Muyinatu A Lediju Bell, and Trac D Tran (2021). "Spectral Gap Extrapolation and Radio Frequency Interference Suppression Using 1D UNets". In: *2021 IEEE Radar Conference (RadarConf)*. IEEE.

Taylor, James D (2012). *Ultrawideband radar: applications and design*. CRC Press.

Nguyen, Lam H, Ravinder Kapoor, and Jeffrey Sichina (1997). "Detection algorithms for ultrawideband foliage-penetration radar". In: *Radar Sensor Technology II*. Vol. 3066. International Society for Optics and Photonics, pp. 165–176.

Nguyen, Lam H, Karl A Kappra, David C Wong, Ravinder Kapoor, and Jeffrey Sichina (1998). "Mine field detection algorithm utilizing data from an ultrawideband wide-area surveillance radar". In: *Detection and Remediation Technologies for Mines and Minelike Targets III*. Vol. 3392. International Society for Optics and Photonics, pp. 627–643.

Nguyen, Lam, Marc Ressler, and Jeffrey Sichina (2008). "Sensing through the wall imaging using the Army Research Lab ultra-wideband synchronous impulse reconstruction (UWB SIRE) radar". In: *Radar Sensor Technology XII*. Vol. 6947. International Society for Optics and Photonics, 69470B.

Carin, Lawrence, Norbert Geng, Mark McClure, Jeffrey Sichina, and Lam Nguyen (1999). "Ultra-wide-band synthetic-aperture radar for mine-field detection". In: *IEEE Antennas and Propagation Magazine* 41.1, pp. 18–33.

Soumekh, Mehrdad (1999). *Synthetic aperture radar signal processing*. Vol. 7. New York: Wiley.

Miller, Timothy, Lee Potter, and John McCorkle (1997). "RFI suppression for ultra wideband radar". In: *IEEE transactions on aerospace and electronic systems* 33.4, pp. 1142–1156.

Nguyen, Lam H and Trac D Tran (2015). "Estimation and extraction of radio-frequency interference from ultra-wideband radar signals". In: *2015 IEEE*

International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, pp. 2848–2851.

Nguyen, Cam and Joongsuk Park (2016). *Stepped-Frequency Radar Sensors: Theory, Analysis and Design*. Springer.

al., Brian R Phelan et (2017a). "Design of ultrawideband stepped-frequency radar for imaging of obscured targets". In: *IEEE Sensors Journal* 17.14, pp. 4435–4446.

al., Brian R Phelan et (2017b). "System upgrades and performance evaluation of the spectrally agile, frequency incrementing reconfigurable (SAFIRE) radar system". In: *Radar Sensor Technology XXI*. Vol. 10188. International Society for Optics and Photonics, p. 1018812.

Nguyen, Lam H and Trac D Tran (2016). "RFI-radar signal separation via simultaneous low-rank and sparse recovery". In: *2016 IEEE Radar Conference (RadarConf)*. IEEE, pp. 1–5.

Song, Yongping, Jun Hu, Yongpeng Dai, Tian Jin, and Zhimin Zhou (2018). "Estimation and mitigation of time-variant RFI in low-frequency ultra-wideband radar". In: *IEEE Geoscience and Remote Sensing Letters* 15.3, pp. 409–413.

Cetin, Mujdat and Randolph L Moses (2005). "SAR imaging from partial-aperture data with frequency-band omissions". In: *Algorithms for Synthetic Aperture Radar Imagery XII*. Vol. 5808. International Society for Optics and Photonics, pp. 32–43.

Nguyen, Lam and Thong Do (2012). "Recovery of missing spectral information in ultra-wideband synthetic aperture radar (SAR) data". In: *2012 IEEE Radar Conference*. IEEE, pp. 0253–0256.

Nguyen, Lam H, Trac Tran, and Thong Do (2014). "Sparse models and sparse recovery for ultra-wideband SAR applications". In: *IEEE Transactions on Aerospace and Electronic Systems* 50.2, pp. 940–958.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 234–241.

Zhang, Kai, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang (2017). "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising". In: *IEEE Transactions on Image Processing* 26.7, pp. 3142–3155.

Allman, Derek, Austin Reiter, and Muyinatu A Lediju Bell (2018). "Photoacoustic source detection and reflection artifact removal enabled by deep learning". In: *IEEE Transactions on Medical Imaging* 37.6, pp. 1464–1477.

Brodeski, Daniel, Igal Bilik, and Raja Giryes (2019). "Deep radar detector". In: *2019 IEEE Radar Conference (RadarConf)*. IEEE, pp. 1–6.

Elbir, Ahmet M, Kumar Vijay Mishra, and Yonina C Eldar (2019). "Cognitive radar antenna selection via deep learning". In: *IET Radar, Sonar & Navigation* 13.6, pp. 871–880.

Mun, Jiwoo, Heasung Kim, and Jungwoo Lee (2018). "A deep learning approach for automotive radar interference mitigation". In: *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. IEEE, pp. 1–5.

Major, Bence et al. (2019). "Vehicle Detection With Automotive Radar Using Deep Learning on Range-Azimuth-Doppler Tensors". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0.

Gurbuz, Sevgi Zubeyde and Moeness G Amin (2019). "Radar-based human-motion recognition with deep learning: Promising applications for indoor monitoring". In: *IEEE Signal Processing Magazine* 36.4, pp. 16–28.

Jokanovic, Branka, Moeness Amin, and Fauzia Ahmad (2016). "Radar fall motion detection using deep learning". In: *2016 IEEE Radar Conference (RadarConf)*. IEEE, pp. 1–6.

Seyfioğlu, Mehmet Saygın, Ahmet Murat Özbayoğlu, and Sevgi Zubeyde Gürbüz (2018). "Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities". In: *IEEE Transactions on Aerospace and Electronic Systems* 54.4, pp. 1709–1723.

Zhang, Gang, Zhi Li, Xuewei Li, and Yiqiao Xu (2020). "Learning synthetic aperture radar image despeckling without clean data". In: *Journal of Applied Remote Sensing* 14.2, p. 026518.

Mason, Eric, Bariscan Yonel, and Birsen Yazici (2017). "Deep learning for radar". In: *2017 IEEE Radar Conference (RadarConf)*. IEEE, pp. 1703–1708.

Gong, Maoguo, Jiaojiao Zhao, Jia Liu, Qiguang Miao, and Licheng Jiao (2015). "Change detection in synthetic aperture radar images based on deep neural networks". In: *IEEE transactions on neural networks and learning systems* 27.1, pp. 125–138.

Deng, Zhipeng, Hao Sun, Shilin Zhou, and Juanping Zhao (2019). "Learning deep ship detector in SAR images from scratch". In: *IEEE Transactions on Geoscience and Remote Sensing* 57.6, pp. 4021–4039.

Schwegmann, Colin P, Waldo Kleynhans, Brian P Salmon, Lizwe W Mdakane, and Rory GV Meyer (2016). "Very deep learning for ship discrimination in synthetic aperture radar imagery". In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, pp. 104–107.

Yonel, Bariscan, Eric Mason, and Birsen Yazıcı (2017). "Deep learning for passive synthetic aperture radar". In: *IEEE Journal of Selected Topics in Signal Processing* 12.1, pp. 90–103.

Thammakhoune, Sean and Emre Yavuz (2020). "Deep learning methods for image reconstruction from angularly sparse data for CT and SAR imaging". In: *Algorithms for Synthetic Aperture Radar Imagery XXVII*. Vol. 11393. International Society for Optics and Photonics, p. 1139306.

Tran, Dung N, Trac D Tran, and Lam Nguyen (2018). "Generative adversarial networks for recovering missing spectral information". In: *2018 IEEE Radar Conference (RadarConf18)*. IEEE, pp. 1223–1227.

Nguyen, Lam, Dung N Tran, and Trac D Tran (2019). "Spectral Gaps Extrapolation for Stepped-Frequency SAR via Generative Adversarial Networks". In: *2019 IEEE Radar Conference (RadarConf)*. IEEE, pp. 1–6.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative adversarial nets". In: *Advances in Neural Information Processing Systems*, pp. 2672–2680.

Ioffe, Sergey and Christian Szegedy (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167*.

Odena, Augustus, Vincent Dumoulin, and Chris Olah (2016). "Deconvolution and checkerboard artifacts". In: *Distill* 1.10, e3.

Shi, Wenzhe, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang (2016). "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1874–1883.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Tropp, Joel A and Anna C Gilbert (2007). "Signal recovery from random measurements via orthogonal matching pursuit". In: *IEEE Transactions on Information Theory* 53.12, pp. 4655–4666.

Varadarajan, Balakrishnan, Sanjeev Khudanpur, and Trac D. Tran (2011). "Stepwise optimal subspace pursuit for improving sparse recovery". In: *IEEE Signal Processing Letters* 18.1, pp. 27–30.

Chapter 7

Summary and Future Directions

This dissertation examines a number of problems where the exciting recent progress made by machine learning can be brought to bear to improve beamforming in the domains of audio, ultrasound, and radar. This includes progress in using machine learning to enhance data prior to beamforming, to replace the beamforming step itself, and to enhance post-beamformed data.

Starting off in Chapter 2, we implemented audiovisual zooming by drawing inspiration from linear discriminant analysis in machine learning to design a novel beamformer that extended the concept of the camera's FOV to enhance audio recording. We presented a method that estimates the sound spectral matrices which accounts for the desired sound signals within the FOV and those outside of the FOV. The estimated spectral matrices allow us to enhance sound coming within the FOV by solving a generalized eigenvalue problem. Our method requires no analysis of captured video frames. It can enhance however many sound sources within the FOV, and the captured imagery is in tandem with the resulting sound signal.

Next, in Chapter 3, we presented a novel deep learning pipeline using

cascaded DNNs in both the time and time-frequency domains to enhance speech suffering from clipping, codec distortions, and gaps, together. The cascaded pipeline developed nears the performance ceiling set by the most challenging single distortion of gaps in speech while simultaneously allowing the function of each component network to remain interpretable.

Moving to ultrasound, in Chapter 4 we re-examined the lag summation step of the short-lag spatial coherence algorithm to improve performance. While the original short-lag spatial coherence (SLSC) imaging algorithm does not consider the content of the images formed at different lags before summing them, our proposed method exploits tissue texture differences in SLSC images created with various short lag values through both weighted summation of individual coherence images (i.e., M-weighting) and the application of robust principal component analysis, demonstrating increased contrast, signal-to-noise ratio, and contrast-to-noise ratio.

Next, in Chapter 5 we demonstrated a deep neural network approach to creating ultrasound images and cyst segmentation results directly in one step from raw single plane wave channel data. This approach holds promise to replace the classical beamform-then-segment approach followed by most imaging pipelines. In addition, our network was trained only with Field II simulated data containing anechoic cysts insonified by single plane waves but generalized to real phantom and *in vivo* data.

Lastly, we move to the radar domain in Chapter 6 where we demonstrated the efficacy of using 1D UNet networks to address bandlimited radio frequency interference, random spectral gaps, and contiguous block spectral

gaps. A separate network was trained for each noise type, and the networks were performant even in challenging scenarios, displaying recovery robustness at multiple noise levels. As our approach operates on individual data apertures, the test sensor geometry is no longer required to match the training sensor geometry making our approach less scene-dependent, a fact demonstrated by training our networks using synthetically generated data on a side-looking geometry and successfully testing our networks on raw synthetic aperture radar (SAR) data collected from a circular 360°-sensing geometry.

7.1 Future Directions

1. **Extending the audiovisual zooming algorithm to reverberant environments:** As our audiovisual zooming algorithm currently stands, it performs poorly in highly reverberant environments. This is because in such environments, a sound source outside of the field of view (FOV) may emit sound waves that arrive to the microphone array from within the desired FOV through reflections. Our audiovisual zooming method is unable to distinguish between these reflected signals and the direct path signals from targets actually in the field of view that we desire to enhance. Future work will investigate addressing this limitation by one or more of dereverberating the signal (Zhang et al., 2020), relaxing the strong coupling between estimation of the spectral matrices and the camera geometry and instead using the visual content (Yu et al., 2020), or estimating the room acoustics by analyzing captured video frames to understand the environment geometry and acoustic properties (Li,

Langlois, and Zheng, 2018; Gao et al., 2020).

2. **Studying complex networks for speech enhancement:** Recent advances in speech enhancement research have enabled deep neural networks to work directly with the complex spectrograms (Hu et al., 2020; Isik et al., 2020). This allows us to no longer require operating in the time domain in order to enhance phase information. Comparing the efficacy of our proposed cascaded pipeline with a single complex network architecture is a promising future direction.

3. **Task-specific deep ultrasound beamformers:** The success achieved by our deep learning beamformer holds promise for future task-specific ultrasound-based approaches to emphasize or deemphasize other structures of interest apart from anechoic cysts. In addition, though we are currently producing two outputs – a B-mode image and a segmentation – we can generalize the architecture to produce more than two output image types (e.g., adding a third simultaneous output of a sound speed image, as estimated with deep learning in Feigin, Freedman, and Anthony, 2019) from a single input image of raw IQ channel data, opening up new possibilities for ultrasound-based clinical, interventional, automated, and semi-automated decision making.

4. **Incorporating scene geometry in deep learning for SAR:** While we achieved promising results operating on single aperture elements, better modeling the geometry of raw SAR data for arbitrary sensor geometries is key to better integrating deep learning in different parts of the SAR imaging pipeline together. While we have achieved some progress in

doing so for fixed (linear) sensor geometries, doing so for arbitrary geometries is still an open question.

References

Zhang, Wangyou, Aswin Shanmugam Subramanian, Xuankai Chang, Shinji Watanabe, and Yanmin Qian (2020). "End-to-end far-field speech recognition with unified dereverberation and beamforming". In: *arXiv preprint arXiv:2005.10479*.

Yu, Jianwei, Shi-Xiong Zhang, Bo Wu, Shansong Liu, Shoukang Hu, Mengzhe Geng, Xunying Liu, Helen Meng, and Dong Yu (2020). "Audio-visual Multi-channel Integration and Recognition of Overlapped Speech". In: *arXiv preprint arXiv:2011.07755*.

Li, Dingzeyu, Timothy R. Langlois, and Changxi Zheng (2018). "Scene-Aware Audio for 360° Videos". In: *ACM Trans. Graph.* 37.4.

Gao, Ruohan, Changan Chen, Ziad Al-Halah, Carl Schissler, and Kristen Grauman (2020). "Visualechoes: Spatial image representation learning through echolocation". In: *European Conference on Computer Vision*. Springer, pp. 658–676.

Hu, Yanxin, Yun Liu, Shubo Lv, Mengtao Xing, Shimin Zhang, Yihui Fu, Jian Wu, Bihong Zhang, and Lei Xie (2020). "Dccrn: Deep complex convolution recurrent network for phase-aware speech enhancement". In: *arXiv preprint arXiv:2008.00264*.

Isik, Umut, Ritwik Giri, Neerad Phansalkar, Jean-Marc Valin, Karim Helwani, and Arvindh Krishnaswamy (2020). "PoCoNet: Better Speech Enhancement with Frequency-Positional Embeddings, Semi-Supervised Conversational Data, and Biased Loss". In: *Proc. Interspeech 2020*, pp. 2487–2491.

Feigin, Micha, Daniel Freedman, and Brian W Anthony (2019). "A deep learning framework for single-sided sound speed inversion in medical ultrasound". In: *IEEE Transactions on Biomedical Engineering* 67.4, pp. 1142–1151.

Vita

Arun Asokan Nair was born in Mumbai, Maharashtra, India in 1992, and spent his adolescent years in Dubai, U.A.E. He received the B.Tech. and M.Tech. degrees in Electrical Engineering from the Indian Institute of Technology Madras, Chennai, India in 2015, and the M.S.E. degree in Electrical and Computer Engineering from the Johns Hopkins University, Baltimore, MD, USA, in 2017, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His research is focused on applying machine learning to the signal processing problem of beamforming. Over the course of his Ph.D., he has worked as a research intern at Snapchat Research, NYC, and Microsoft Research (Applied Sciences Group), Redmond. He received the Best Paper Award at the Association for Computing Machinery (ACM) International Conference on Multimedia, 2019, for his work on audiovisual zooming.