

DELFT UNIVERSITY OF TECHNOLOGY

MASTER THESIS

Semi-Controllable Compression Schemes for Ultrasound Imaging

Author:
Xuyang LI

Supervisor:
Prof. Geert LEUS

*A thesis submitted in fulfillment of the requirements
for the degree of Master of Science*

in the

Circuits and System
Electrical Engineering, Mathematics and Computer Science

November 27, 2017

Abstract

The application of *Compressive Sampling (CS)* in medical ultrasound has been widely studied in recent years with the growing requirement of reconstructing high quality images with smaller data size. Most of the current studies with successful CS reconstruction are mainly focusing on the mathematical applications of CS theory in ultrasound imaging. However, the randomized mechanisms in these studies are hard to be fully fulfilled in hardware. In addition, some studies try to discover the sparse representation of signals by ignoring a part of information rather than compressing all data. We propose a new compression scheme for fast image acquisition in ultrasound imaging using a method, which is similar in style to CS. Our scheme is based on the formulation of an *inverse scattering problem (ISP)*, where the Born approximation has been used during its derivation. In our system, the ultrasound image is represented by a collection of hypothetical points, what can be called pixels. These points are identified by their unique spatial impulse responses relative to the elements in the transducer. The randomized linear combinations of the spatial impulse responses received by the transducer elements can maintain the uniquenesses of these points, similar to coding techniques in data compression. Hence, our compression scheme is more controllable than conventional CS, which can achieve a real-time compression of data during the acquisition stage in hardware. We employ L2-regularization to solve the ill-posed ISP rather than the L1-regularization used in CS since there is no any assumption of signal sparsity. In our work, we finally achieve acceptable reconstructions by compressing the raw data to 12.5% of its original size. The results are better than 12.5% with multiplexing the received signals from 12.5% elements in the array and almost as good as 25% with multiplexing the received signals from 25% elements in the array.

Acknowledgements

It is my pleasure to have an opportunity to express my thanks to everyone who supported me during the period of the thesis.

Foremost, I would like to thank my daily supervisor, Pim van de Meulen for the continuous support of my thesis project, for his patience, motivation, and knowledge. His guidance helped me in all the time of research and writing of this thesis. Without his help, so many problems which I have faced in the research cannot be solved easily.

Beside my daily supervisor, I would like to thank my professor, Geert Leus. His office was always open whenever I ran into a trouble spot or had a question about my research or writing. I have learned a lot from both his lectures and thesis related discussions.

I also want to thank my thesis committee members: Michiel Pertijs and Pieter Kruizinga, for their insightful comments and constructive suggestions.

Finally, I must express my very profound gratitude to my parents and friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Xuyang Li
Delft, The Netherlands
14/11.2017

Contents

Abstract	i
Acknowledgements	iii
1 Introduction	1
1.1 Motivation	1
1.2 Research Objectives	2
1.3 Relative Research and Novelty of our Research	2
1.4 Thesis Organization	3
2 Compression and Imaging Methodology	5
2.1 Introduction	5
2.2 Linear Measurement Model	5
2.3 Reconstruction Methods	7
2.4 The Principle of the Compression Scheme	8
3 Case 1: Entire Array Compression for a 3MHz Ultrasound System	11
3.1 Initial Parameters	11
3.2 Non-Compressed Ultrasound Imaging	12
3.3 Compressed Ultrasound Imaging	14
3.4 The Potential Improvement Methods	17
3.4.1 Smaller Pixel Size	18
3.4.2 Larger Delay Range	18
3.5 Performance Analysis	20
3.6 Discussion of Three Preprocessing Methods	25
3.6.1 The Defective of Phase Shift	25
3.6.2 The Advantages and Disadvantages of Sample Shift	26
3.7 Summary	27
4 Case 2: An Improved Design for a 18.5MHz Ultrasound System	29
4.1 Introduction	29
4.2 New Compression Architecture	30
4.3 Updated Preprocessing Methods	32
4.4 Performance Analysis	34
4.5 Discussion of the Advantages and Disadvantages of preprocessing methods and three architectures	38
4.5.1 The Impact of Architectures	38
4.5.2 The Impact of Preprocessing Methods	39
4.6 Summary	40
5 Conclusion	41
5.1 Summary	41
5.2 Future Work	41

A Reconstruction Results with New Architectures and Updated Preprocessing Methods(Pixel Size: 0.06mm)	43
B Reconstruction Results with New Architectures and Updated Preprocessing Methods(Pixel Size: 0.02mm)	47

List of Figures

2.1	Schematic Diagram of the Imaging Algorithm with Linear Element Array	6
2.2	Schematic Diagram of the 40 elements array, the distortion function includes the Sample Shift (τ is the number of samples, $\mathbf{e}_i = \mathbf{e}_{i,Sample\ Shift}$), the Phase Shift (τ is the change of signal's phase, $\mathbf{e}_i = \mathbf{e}_{i,Phase\ Shift}$) and the Amplitude Weight (τ is the gain, $\mathbf{e}_i = \mathbf{e}_{i,Amplitude\ Weight}$). \mathbf{y} is also equal to a column of \mathbf{A} when only one pixel is measured	9
3.1	Excitation of Transducer	12
3.2	Location of Point Objects	12
3.3	Non-Compression Ultrasound Imaging(dB mode). Raw 40= Raw data from all 40 elements	13
3.4	Regularized Least Squares (Off-axis Point Objects/Noisy/dB mode)	15
3.5	Matched Filter (Off-axis Point Objects/Noisy/dB mode)	15
3.6	Point Spread Functions of All Point Objects (Sample Shift (dB mode))	16
3.7	Point Spread Functions of All Point Objects (Phase Shift (dB mode))	16
3.8	Point Spread Functions of All Point Objects (Amplitude Wight (dB mode))	17
3.9	Raw data from 10 elements(Off-axis Point Objects/Noisy/dB mode)	17
3.10	Regularized Least Squares (Sample Shift/Off-axis Point Objects/Smaller Pixel Size)	18
3.11	Regularized Least Squares(Off-axis Point Objects/Noisy/Larger Maximum Delay)	19
3.12	Matched Filter(Off-axis Point Objects/Noisy/Larger Maximum Delay)	20
3.13	Condition Number variation with the Maximum Delay (Sample Shift/Linear Scale)	21
3.14	Condition Number Variation with Number of Measurements (Three preprocessing methods/dB Mode)	22
3.15	Condition Number Variation with Number of Sensors (dB Mode)	23
3.16	Effect of Phase Shift	25
3.17	Sample Shift Scheme of Two Symmetric Points	26
4.1	Schematic Diagram of Architecture 1. The distortion functions include the Sample Shift (τ is the number of delayed samples), the Amplitude Weight (τ is a sequence of gains for all samples) and the Random Subsampling (τ is the multiple of sampling frequency). Observation vector \mathbf{y}_i equals a column of \mathbf{A} when only one pixel is measured	31
4.2	Schematic Diagram of Architecture 2. The distortion functions include the Sample Shift (τ is the number of delayed samples), the Amplitude Weight (τ is a sequence of gains for all samples) and the Random Subsampling (τ is the multiple of sampling frequency). Observation vector \mathbf{y}_i equals a column of \mathbf{A} when only one pixel is measured	31

4.3	Schematic Diagram of Architecture 3, the distortion functions include the Sample Shift (τ is the number of delayed samples), the Amplitude Weight (τ is a sequence of gains for all samples) and the Random Subsampling (τ is the multiple of sampling frequency). Observation y_i equals a column of \mathbf{A} when only one pixel is measured	32
4.4	Random Subsampling Diagram	33
4.5	True Location of the Point Objects	34
4.6	Axial and Lateral image profile of the reconstructed images with Regularized Least Squares	38
A.1	Regularized Least Squares(Amplitude Weight/Off-axis Point Objects/Noisy/dB mode)	43
A.2	Regularized Least Squares(Sample Shift up to 20 samples/Off-axis Point Objects/Noisy/dB mode)	43
A.3	Regularized Least Squares(Random Subsampling/Off-axis Point Objects/Noisy/dB mode)	44
A.4	Regularized Least Squares(Random Grouping Sample by Sample+Amplitude Weight+ Sample Shift/Offgrid/Noisy/dB mode)	44
A.5	Regularized Least Squares(Naive Compression/Off-axis Point Objects/Noisy/dB mode)	44
A.6	Regularized Least Squares(Off-axis Point Objects/Noisy/dB mode). Raw data from sensors without compression.	45
A.7	Matched Filtering(Amplitude Weight/Off-axis Point Objects/Noisy/dB mode)	45
A.8	Matched Filtering(Sample Shift up to 20 samples/Off-axis Point Objects/Noisy/dB mode)	45
A.9	Matched Filtering(Random Subsampling/Off-axis Point Objects/Noisy/dB mode)	45
A.10	Matched Filtering(Random Grouping Sample by Sample+Amplitude Weight+Sample Shift/Off-axis Point Objects/Noisy/dB mode)	46
A.11	Matched Filtering(Naive Compression/Off-axis Point Objects/Noisy/dB mode)	46
A.12	Matched Filtering(Off-axis Point Objects/Noisy/dB mode). Raw data from sensors without compression.	46
B.1	Regularized Least Squares(Amplitude Weight/Off-axis Point Objects/Noisy/dB mode)	47
B.2	Regularized Least Squares(Sample Shift up to 20 samples/Off-axis Point Objects/Noisy/dB mode)	47
B.3	Regularized Least Squares(Random Subsampling/Off-axis Point Objects/Noisy/dB mode)	48
B.4	Regularized Least Squares(Random Grouping Sample by Sample+Amplitude Weight+ Sample Shift/Offgrid/Noisy/dB mode)	48
B.5	Regularized Least Squares(Naive Compression/Off-axis Point Objects/Noisy/dB mode)	48
B.6	Regularized Least Squares(Off-axis Point Objects/Noisy/dB mode). Raw data from sensors without compression.	49
B.7	Matched Filtering(Amplitude Weight/Off-axis Point Objects/Noisy/dB mode)	49
B.8	Matched Filtering(Sample Shift up to 20 samples/Off-axis Point Objects/Noisy/dB mode)	49

B.9 Matched Filtering(Random Subsampling/Off-axis Point Objects/Noisy/dB mode)	49
B.10 Matched Filtering(Random Grouping Sample by Sample+Amplitude Weight+Sample Shift/Off-axis Point Objects/Noisy/dB mode)	50
B.11 Matched Filtering(Naive Compression/Off-axis Point Objects/Noisy/dB mode)	50
B.12 Matched Filtering(Off-axis Point Objects/Noisy/dB mode). Raw data from sensors without compression.	50

List of Tables

3.1	Average and Standard Deviation of Condition Number (Sample Shift/Linear scale)	21
3.2	Average and Standard Deviation (Sample Shift/Linear scale)	22
3.3	Average and Standard Deviation (Phase Shift/Linear scale)	22
3.4	Average and Standard Deviation (Amplitude Weight/Linear scale)	23
3.5	Average and Standard Deviation (Sample Shift/Linear scale)	24
3.6	Average and Standard Deviation (Phase Shift/Linear scale)	24
3.7	Average and Standard Deviation (Amplitude Weight/Linear scale)	24
4.1	Average of Cross-correlation between Pixels (Linear scale). AW=Amplitude Weight, SS=Sample Shift, RandgrpSbyS=Random Grouping Sample by Sample, '+'=Combination of methods	35
4.2	Standard Deviation of Cross-correlation between Pixels (Linear scale). AW=Amplitude Weight, SS=Sample Shift, RandgrpSbyS=Random Grouping Sample by Sample, '+'=Combination of methods	35
4.3	Mean Squared Error between x and estimated x (Regularized Least Squares/Linear scale). AW=Amplitude, Weight SS=Sample Shift, RandgrpSbyS=Random Grouping Sample by Sample, '+'=Combination of methods	36
4.4	Mean Squared Error between x and estimated x (Matched Filtering/Linear scale). AW=Amplitude, Weight SS=Sample Shift, RandgrpSbyS=Random Grouping Sample by Sample, '+'=Combination of methods	37

Chapter 1

Introduction

1.1 Motivation

Ultrasound Imaging, also known as medical ultrasound, is a mature research area where multiple imaging modes have been researched. Nowadays, *Brightness-mode (B mode)* is used extensively to inspect internal body structure in clinical diagnose. In B-mode imaging, *conventional delay and sum beamforming (CBF)* is used. For recent decades, adaptive beamforming, which has been widely applied in SONAR and Rader signal processing, was designed for medical ultrasound. In general, the image quality can be improved by using more elements for any imaging method. With the emerging requirement to achieve a higher image quality, the increase of element numbers will result in extremely large rates of data processed by the preprocessing unit. Moreover, the fabrication complexity is high if electronic devices are dense in a limited area. Traditionally, this issue is solved by digitally down-sampling the data at the front-end. It is explained by the fact that the spectrum of a modulated signal equals only a portion of its baseband bandwidth [2]. Furthermore, the Shannon-Nyquist theorem [6] shows that the sampling frequency should be larger than twice the bandwidth to avoid aliasing of signals. All these factors constrain the application of high-resolution imaging in Ultrasound Imaging.

As one of the fundamental theorems in information theory, the Shannon-Nyquist sampling frequency is usually the minimal frequency for sampling and reconstructing an analog band-limited signal perfectly [1]. However, this theorem is questioned by a more recent signal processing technique called compressive sensing (CS) when signals are sparse. The theory of CS shows that signals can be efficiently acquired and reconstructed, which can be used to find a solution of the under-determined linear inverse system. The framework of CS [7] states that sparse signals can be reconstructed perfectly with a small amount of raw data. The research in [8][9] has shown that additional structure of compressive sensing can reduce the number of measurements without loss of accuracy. These papers have also provided an interesting feature that a randomized sensing mechanism can achieve a perfect result with high probability. However, there are two problems for a CS implementation in our imaging algorithm. Firstly, we did not assume that our object has the sparse representation in a particular domain like the other studies. Secondly, since our goal is to achieve the compression scheme in hardware design, the random sensing mechanism should be easily implemented. Hence, we prefer to find an easier design over translating conventional CS into hardware design directly.

When we consider the image reconstruction problem as a linear inverse problem $\mathbf{y} = \mathbf{Ax}$, the principle of it is equivalent to the regression problem to find out the coefficient vector \mathbf{x} such that \mathbf{Ax} is the closest point in the column space of \mathbf{A} to the observation vector \mathbf{b} . The linear problem can be solved by Regularized Least Squares. In [10], a sketching scheme has been applied to solve the problem of heavy

computation complexity when the sizes of \mathbf{y} and \mathbf{A} are massive. In this thesis, a compression scheme is designed by the same strategy for any type of linear inverse ultrasound imaging algorithm aimed at reducing the output data size to speed up the acquisition rate. We compress all data from the entire array spatially, both the output channels and data size are compressed simultaneously which causes that less Analog-to-Digital Converters are required in hardware design.

1.2 Research Objectives

The thesis will focus on designing a compressive sampling method during the acquisition stage of an ultrasound imaging system, which can be realized in hardware design easily. The primary objectives of the work are to:

- Verify the feasibility of the chosen ultrasound imaging algorithm in the presence of noise and modeling errors due to discretization of space pixelization.
- Evaluate the performance of the imaging system in the same environment after our compression scheme has been added
- Find out how the design parameters may influence the reconstruction quality
- Figure out the potential troubles of the design and find improvements or alternative methods for the initial designs
- Evaluate the performance of enhanced designs by comparing them with the imaging with the uncompressed signals and analyze the feasibility of hardware implementation with these new designs

1.3 Relative Research and Novelty of our Research

When researching the compression of data during the acquisition stage, the application of CS is inevitable in existing studies. The foundation of CS is based on two properties: sparsity and incoherence. Sparsity reflects the ability of a signal to be compressed. A signal that has a sparse representation in a given basis will have only a few non-zero values in this representation. Hence, the signal can be compressed and well reconstructed from relatively few samples by suppressing the other zero values in the signals. Incoherence in CS expresses the idea that signals that are sparse in a given basis cannot be sampled in this basis but in another basis where the signal is dense. This property guarantees that the samples acquired contain the same amount of information.

The major concern in CS is that the data should have a sparse expansion such that the number of non-zero coefficients of the image or signal in this representation basis is as small as possible. Therefore, the existing studies of CS in ultrasound imaging can be divided into three categories: the sparse distribution of scatterers, the sparse raw RF signals and the sparse RF images in the Frequency domain. In [15][16], the reconstruction is solving the direct inverse scattering problem. They plan to produce an image with only one single plane wave transmission. The first difference between their design and ours is that they apply the CS technique under the constraint that the scatterer distribution is sparse. But in our design, we didn't use the property of sparsity as in conventional CS. The other difference is their data are in the frequency domain, where our design only processes the time-domain signal. The main problem of their design is that extra memory is required to store the

data before compression, and thus the price is too high in hardware implementation. In [17][18], the authors assume the raw RF data have sparse expansion in some basis. Their objective is to reduce the quantity of pre-beamformed data by selecting the best representation basis where the number of non-zero coefficients of the sparse representation is small. At first, they only under-sample the RF data by hardware and acquire the sparse representation of compressed data by l_1 -minimization. Then, the compressed data can be computed with its sparse representation and an approximate basis. The entire compression process cannot be done in hardware individually and the compression is done temporally rather than spatially which is different from our design. In [19][20], the authors propose a reconstruction of post-beamforming images via the CS technique. The assumption is that the 2D Fourier transform of ultrasound images is sparse. In ultrasound imaging, the acquisition consists of taking samples of the image. This sampling scheme is similar to a basis of Dirac. It is equivalent to applying a sampling mask, which is a uniform random pattern in the two spatial directions. Based on the definition of CS, the Fourier basis is chosen because it is incoherent with the Dirac basis and ultrasound images are sparse in the Fourier domain. A similar strategy has been studied in the optical-imaging research field. The paper [31] assumes that the optical image is sparse in a known basis. A mask is applied in front of the transducer to construct a transmission matrix, which is incoherent with the known basis. Then, the complex optical field at discrete positions can be efficiently multiplexed to less sensors. Once the transmission matrix is known, the reconstruction can be done with less sensors than before. The spatial compression of all data into less output channels is similar to our scheme, but we construct the transmission matrix or compression matrix without using the sparsity of signals.

One of the main features of the existing studies we have displayed above is the type of signals to be reconstructed and the choice of the representation where the ultrasound data are assumed to be sparse. However, our compression scheme does not consider and exploit the property that there is existence of sparsity in any domain. The compression algorithm in our design is fully done by hardware in time domain during the data acquisition stage, where most conventional CS studies require the assistance of software. Furthermore, the reconstruction is also not done by l_1 -minimization, but with much simpler methods like Regularized Least Squares and Matched Filter. But, in our design, we have kept a feature of CS. The sampling strategy is an pseudo-random linear combination of signals, where each measurement contains the information of the raw signals. Since the total randomization of signals is impossible if the compression is real-time in the hardware without the assistance of temporary memory, only the parameters in processing methods are randomly generated in our design. One main objective of our compression scheme is to approximate the pseudo-random combination to the totally random combination. The second main objective is to ensure the low hardware-complexity in our design such that all procedures in the compression can be implemented with an easy structure.

1.4 Thesis Organization

In the thesis, we have researched two different compression schemes. In the first design, we compressed the data from the entire array fully into 1 output channel each time. Based on the performance and results of this initial scheme, the alternative compression schemes with less compression in the spatial dimension will be

studied as the second case. The thesis is organized as the following chapters:

Chapter 2 contains a brief introduction of the methodology of ultrasound imaging, the basic theories of reconstruction methods and the principle of our compression model.

Chapter 3 is about the first case study. The declaration of the initial parameter setting of our simulation environment is given firstly. Then, it shows reconstruction results in two cases where the raw data are uncompressed or compressed. The performance analysis section examines the influences of different design parameters on the reconstruction quality. Finally, a discussion of our compression scheme is given. The advantages and disadvantages of different preprocessing methods are analyzed based on their principles. Potential solutions of improvement or alternative solutions are proposed based on the analysis.

Chapter 4 describes the details of our second case study and examines the performances of these alternative designs. The evaluation of them can be divided into two parts: 1. The differences between various new designs of compression are tested and stated; 2. The results of the best designs and the uncompressed case are compared to determine the feasibility of the new compression schemes. The difficulties of hardware implementation are also discussed in the end.

Chapter 5 concludes the contributions and progress of the work, and provides some suggestions for the future direction of this research.

Chapter 2

Compression and Imaging Methodology

2.1 Introduction

Our ultrasound imaging is based on the formulation of an *inverse scattering problem (ISP)*. For its derivation, we utilize the Born approximation [23] and assume the emission of a single broadband plane sound wave. The formula is

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}; \quad (2.1)$$

where \mathbf{x} is the object image, \mathbf{y} is a single pulse-echo measurement, the matrix \mathbf{A} describes the pulse-echo signals of all hypothetical points and \mathbf{n} is measurement noise. Each element in \mathbf{x} is the brightness of a pixel in the image region. The detail of this imaging algorithm will be discussed in later sections.

In conventional CS, this linear model is usually solved by ℓ_1 -minimization. But in this thesis, we only concentrate on two more common methods, Regularized Least Squares and Matched Filter rather than optimization. The performances of them in different environments will be evaluated with their reconstruction results in later chapters.

The organization of this chapter is as follows. The details of the imaging algorithm will be presented in section 2.2. Then the methodology we will use to solve our linear model will be introduced in section 2.3. In section 2.4, the principles and details of the compression scheme will be introduced.

2.2 Linear Measurement Model

The diagram of the imaging algorithm is shown in Figure 2.1. The Image region is subdivided into a grid formed by a group of hypothetical point scatterers. Each hypothetical point scatterer is a pixel in the ultrasound image. There are $N_{pix} = N_x N_z$ hypothetical point scatterers in the image region. At the beginning, all elements send out plane waves forward simultaneously in transmitter mode, and turn into receiver mode after transmission. Then, an ultrasound field is created by these emitted plane waves. We suppose that our image region is a homogeneous region. During the transmission, the amplitude or energy of the plane waves will decrease gradually. For a specific point in space, the ultrasound field there is a function of time which is the impulse response. Since the impulse response will vary as a function of position relative to the transducer, it is a spatial-temporal impulse response. The pressure at a point of interest j can be modeled as the convolution of the transmit spatial impulse response $\mathbf{h}_{tx,j}[n]$, the transmit aperture's electrical impulse response $\mathbf{h}_t[n]$ and

the excitation $\mathbf{u}[n]$

$$\mathbf{p}_j[n] = \mathbf{u}[n] * \mathbf{h}_{tx,j}[n] * \mathbf{h}_t[n] \quad (2.2)$$

where $*$ denotes temporal convolution. The spatial impulse response of the point j received by the element i in the array is the convolution of the pressure $\mathbf{p}_j[n]$, the receive spatial impulse response $\mathbf{h}_{rx,i}[n]$ and the backward electrical impulse response $\mathbf{h}_r[n]$,

$$\mathbf{f}_{i,j}[n] = \mathbf{p}_j[n] * \mathbf{h}_{rx,i}[n] * \mathbf{h}_r[n] \quad (2.3)$$

$$= \mathbf{u}[n] * \mathbf{h}_{tx,j}[n] * \mathbf{h}_t[n] * \mathbf{h}_{rx,i}[n] * \mathbf{h}_r[n] \quad i \in \{1, 2, \dots, N\} \quad (2.4)$$

The spatial impulse response can be regarded as the received response emitted by a point which is equivalent to a signal source [24][25]. Finally, our system model can be simplified as in Figure 2.1. We have used the toolbox Field II [21][22] for the later simulations.

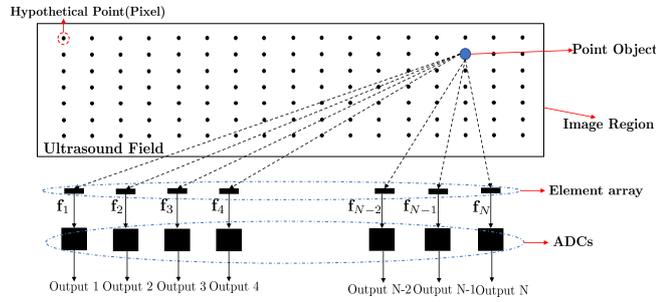


FIGURE 2.1: Schematic Diagram of the Imaging Algorithm with Linear Element Array

For each element the system of linear equations

$$\mathbf{y}_i = \mathbf{A}_i \mathbf{x} \quad (2.5)$$

can be established. For N elements in the array, $i \in \{1, 2, \dots, N\}$, the equation 2.5 can be augmented to

$$\mathbf{y}_F = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_N \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{f}_{1,1}, \mathbf{f}_{1,2}, \dots, \mathbf{f}_{1,N_{pix}} \\ \mathbf{f}_{2,1}, \mathbf{f}_{2,2}, \dots, \mathbf{f}_{2,N_{pix}} \\ \vdots \\ \mathbf{f}_{N,1}, \mathbf{f}_{N,2}, \dots, \mathbf{f}_{N,N_{pix}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N_{pix}} \end{bmatrix} = \mathbf{A}_F \mathbf{x} \quad (2.6)$$

For the element i , the matrix \mathbf{A}_i is built column by column. The column $\mathbf{f}_{i,j}$ reflects the received signals from 1 hypothetical point j received by element i . The process should be reiterated until all pixels in the image have been measured. Finally, matrix \mathbf{A}_i contains the spatial impulse responses of all pixels in the image region:

$$\mathbf{A}_i = [\mathbf{f}_{i,1}, \mathbf{f}_{i,2}, \dots, \mathbf{f}_{i,N_{pix}}] \quad (2.7)$$

$$\mathbf{f}_j = \begin{bmatrix} \mathbf{f}_{1,j} \\ \mathbf{f}_{2,j} \\ \vdots \\ \mathbf{f}_{N,j} \end{bmatrix} \quad (2.8)$$

The quality of \mathbf{A}_F will highly influence the reconstruction because it is the only prior knowledge of the image region before practical implementation.

The signal \mathbf{y} , which is a real ultrasound measurement, is constructed by using the same procedure as one column of matrix \mathbf{A} . The difference is that \mathbf{y} is a single snapshot of the entire image region when there are multiple point objects instead of 1. It means that echoes from several point objects will be received by the elements of the array. In the ideal case, the single snapshot \mathbf{y} should equal the summation of corresponding columns in matrix \mathbf{A} of all selected pixels. It should be noted that the length of \mathbf{y}_i should be aligning with the row number of \mathbf{A}_i .

The similar imaging algorithms have been used in ultrasound imaging in paper [5][13][14][15][16]. However, in all these papers, all data received by the entire array are stored in the acquisition stage. The large amount of Analog-to-Digital Converters(ADC) and data sizes are two problems which restrict the hardware implementation of this algorithm. Therefore, a compression scheme should be designed for it.

2.3 Reconstruction Methods

The reconstruction of ultrasound signals is done by solving the linear problem $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$. The first method we will use is Regularized Least Squares. Least Squares is an approach to approximate the solution of an overdetermined system of equations by minimizing the sum of squares of the residual.

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \quad (2.9)$$

However, the matrix \mathbf{A} is always ill-conditioned in practical applications. The ill-posed matrix \mathbf{A} is one of the main problems that cause the complexity of signal reconstruction and an unstable solution. The system will be very sensitive to changes at the input side or noise/modeling errors, which can cause a noisy and unstable solution. *Tikhonov Regularization*[26] is the most popular regularization that has been used to solve this problem. It transforms the problem into minimizing the quantity

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda^2 \|\mathbf{L}\mathbf{x}\|_2^2 \quad (2.10)$$

where λ is the regularization parameter to balance the weight between the minimization of the side constraint $\|\mathbf{L}\mathbf{x}\|_2^2$ and the minimization of the residual $\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2$. Throughout this paper, $\|\cdot\|$ denotes the Euclidean vector norm or the associated induced matrix norm. Therefore, the outcome of Tikhonov regularization can be considered as a trade-off between least squares and matched filtering. In our experiment, $\mathbf{L} = \mathbf{I}$. The desired value of λ can be computed inexpensively by using L-curves [27] when the *singular value decomposition*(SVD) of \mathbf{A} is known. However, when the size of \mathbf{A} is extremely large, the price of SVD is too high. Then *LSQR* [28] is a promising alternative regularization method. This is an algorithm similar in style to the famous conjugate gradient(CG) algorithm. It's based on the bi-diagonalization

procedure of Golub and Kahan [29]. It generates a sequence of estimates \mathbf{x}_i where the residual norm $\|\mathbf{y} - \mathbf{A}\mathbf{x}_i\|_2^2$ decreases monotonously. However, LSQR exhibits semi-convergence at some iteration. The estimation becomes a better approximation to the true solution with more iterations, and the noise will dominate the estimation after a specific iteration. It means LSQR with limited iterations has the regularization effect [30], where the iteration number k plays the role of the regularization parameter.

The second method is *Matched Filter*, which is commonly used in the case when a known signal is sent out, and the reflected signal is examined for common elements of the outgoing signal. A matched filter is equivalent to correlating the measurement vector \mathbf{y} with the sensing matrix \mathbf{A} to detect the known signal that is embedded in the noise. The vector \mathbf{y} can be regarded as a kind of combination of corresponding columns from \mathbf{A} , then the multiplication of the conjugate transpose of \mathbf{A} and \mathbf{y} is equal to finding the correlations:

$$\hat{\mathbf{x}} = \mathbf{A}^H \mathbf{y} \quad (2.11)$$

Since vector \mathbf{y} can be regarded as a kind of combination of corresponding columns of selected point objects in the matrix \mathbf{A} , the structure of this equation shows that the performance of the matched filter will be highly related to the Energy Distribution of the matrix \mathbf{A} . The Energy Distribution is defined as the root of the 2-norm of every column in \mathbf{A} , which shows the signal strength of every pixel received by the element array. The strength will be transformed into the brightness of points, which mean that points with higher energy will also be brighter in the image.

2.4 The Principle of the Compression Scheme

Both, the Least Squares $\mathbf{A}^T \mathbf{A} \hat{\mathbf{x}} = \mathbf{A}^T \mathbf{y}$ and the Matched Filter $\hat{\mathbf{x}} = \mathbf{A}^H \mathbf{y}$ indicate that the quality of reconstruction is related to the differences between columns (pixels) in \mathbf{A} . Unlike the traditional compressive sensing theory, the initial idea of our design is to compress the signal in a much simpler way by summing up received signals from all sensors.

$$\mathbf{y}_C = \sum_i \mathbf{e}_i \quad i \in \{1, 2, \dots, N\} \quad (2.12)$$

$$\mathbf{a}_j = \sum_i \mathbf{e}_{i,j} \quad i \in \{1, 2, \dots, N\} \quad (2.13)$$

where \mathbf{y} is the input vector in our linear model and \mathbf{a}_j is one column of matrix \mathbf{A} if only one point object is measured, N is the number of elements in the array and \mathbf{e}_i is the received signal \mathbf{f}_i from element i after preprocessing. Then signals from all channels can be integrated into 1 output channel by summation. Once the summation of the preprocessed signals is linear, the compression system can be described as:

$$\mathbf{y}_C = \mathbf{C}\mathbf{y}_F = \mathbf{C}\mathbf{A}_F \mathbf{x} = \mathbf{A}_C \mathbf{x} \quad (2.14)$$

where \mathbf{y}_C and $\mathbf{A}_C = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{N_{pix}}]$, $j \in \{1, 2, \dots, N_{pix}\}$ are compressed data, \mathbf{y}_F and \mathbf{A}_F are the raw data from all elements in the array, and \mathbf{C} is the compression matrix. For simplicity sake, we use \mathbf{A} to represent the compressed matrix \mathbf{A}_C if no further statement has been made.

In Figure 2.2, a diagram of the simplified equivalence model is shown. The raw data from N elements have been compressed into 1 output. The raw signals received by all elements are preprocessed before summation. If we did not apply any

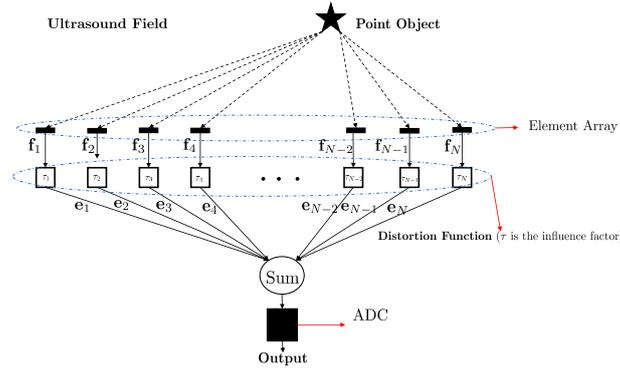


FIGURE 2.2: Schematic Diagram of the 40 elements array, the distortion function includes the Sample Shift (τ is the number of samples, $\mathbf{e}_i = \mathbf{e}_{i,Sample\ Shift}$), the Phase Shift (τ is the change of signal's phase, $\mathbf{e}_i = \mathbf{e}_{i,Phase\ Shift}$) and the Amplitude Weight (τ is the gain, $\mathbf{e}_i = \mathbf{e}_{i,Amplitude\ Weight}$). \mathbf{y} is also equal to a column of \mathbf{A} when only one pixel is measured

preprocessing method to the received signals, compression is done by linear summation. We name it *Naive Compression* for convenience of description. In Figure 2.1, for the pixels which are symmetric relative to the transducer, the received signals of all elements for them should be also symmetric. Therefore, the summations of these pairs of pixels are the same. In addition the adjacent pixels of these pairs of pixels are also highly similar with them. As a consequence, the collection of pixels, which are symmetric with respect to axial, are concentrated in a narrow area in lateral dimension. Then pixels, which have the same distance to the element array, will have compressed signals in the same time interval after the summation. The reconstruction can only provide us with the depths of the point objects that are represented by curves that contain the pixels with the same distances to the array. Hence, the preprocessing method is needed to differentiate pixels.

Three preprocessing methods are applied in the following simulation. Since the intention of preprocessing is to differentiate the received signals from all pixels, we may also call them differentiation methods in the report later. Three types of methods can be classified into two categories, delay and amplitude gain. We plan to try both phase shift and sample shift to delay our received signals. Sample shift can be easily regarded as a time delay of the discrete signal (or sampled signal). For element i , when the original received signal is $\mathbf{f}_i[n]$, the sample shift processing can be represented by

$$\mathbf{e}_{i,Sample\ Shift}[n] = \mathbf{f}_i[n - \tau_i], \quad i \in \{1, 2, \dots, N\} \quad (2.15)$$

where the delay τ_i is a uniformly distributed random number in a specific interval.

Unlike sample shift that delays the signal in the time domain directly, a phase shift should be applied in frequency domain firstly. For element i , when original received signal is $\mathbf{f}_i(n)$, the procedure of the phase shift contains several steps. Firstly, we should transform the signal into the frequency domain $\mathbf{f}_i(\xi)$. Then, the phase shift can be represented by

$$\mathbf{e}_{i,Phase\ Shift}[n] = \mathcal{F}^{-1}(\mathbf{f}_i(\xi)e^{-2j\pi\xi\tau_i}), \quad i \in \{1, 2, \dots, N\} \quad (2.16)$$

where \mathcal{F}^{-1} is the inverse Fourier transform operator and $\zeta\tau_i$ is a random number uniformly distributed from 0 to 1, which can restrict $2\pi\zeta\tau_i$ in the range from 0 to 2π .

Besides using a delay to differentiate signals from different elements, we can also let the received signal of each element be multiplied with a random amplitude weight τ_i ,

$$\mathbf{e}_{i,weighted}[n] = \tau_i \mathbf{f}_i[n], \quad i \in \{1, 2, \dots, N\} \quad (2.17)$$

where τ_i is a uniformly distributed random number in interval -1 to 1.

All three methods are able to change the initial property of the received signals from different elements. As a result, the summation of the spatial impulse responses from different pixels should be diverse. As aforementioned, the pixels at the same depth are the interference introduced by our linear summation model. However, the time-of-arrival of received signals from elements are distributed differently for the symmetric point objects in fact. All these preprocessing is able to distort the initial signals well, where the summations of preprocessed signals of them are hard to be the same. Finally, the pixels at the same depth which are highly similar to each other initially can be represented differently in the compressed data. In the following sections, all three preprocessing methods will be applied under the same condition in each experiment in order to compare their performances.

Chapter 3

Case 1: Entire Array Compression for a 3MHz Ultrasound System

In our first design, the received signals from the entire array will be compressed into 1 output signal in 1 round of compression. The data size of the compressed data will be $\frac{1}{40}$ of the raw data, and only 1 ADC is required instead of 40 ADCs for 1 output channel. The complexity and price of hardware implementation will be reduced significantly.

We should notice that our imaging algorithm highly relies on the preset grid in the image region. In reality, with the restriction of computational resources and consideration of efficiency, the number of pixels cannot be infinite to cover the entire region, which also causes other problems. It does mean that many point objects are off-axis (or off-grid) in reality. Therefore, it is a big challenge to accurately reconstruct point-like objects if they are off-axis. In order to test the worst off-axis case, all selected point objects are located at the center of the mesh when we measure vector \mathbf{y} . Additionally, the noise is inevitable for any kind of measurement. To simplify the problem, we only consider a simulation environment in Gaussian white noise and the SNR is 20dB. The combination of these two interferences is the worst environment for the compression-imaging system in our current assumption. All simulations will be done in this environment.

In this chapter, the results of our compression system with different preprocessing methods, reconstruction methods and parameters will be displayed and explained separately. The organization of this chapter is as follows. The initial parameters of the simulation are declared in section 3.1. Sections 3.2 and 3.3 display all ultrasound images with or without our compression system in an off-axis and noisy environment. The potential improvement methods are proposed in section 3.4. The performance of our compression design is analyzed in section 3.5. The advantages and disadvantages of different preprocessing methods and compression designs will be discussed in section 3.5. Section 3.6 is a brief summary of this chapter.

3.1 Initial Parameters

The setting of parameters is significant for the simulation that can reflect the performances of the system in diverse environments. The initial parameters that will be used in this chapter for all experiments are declared in this section. They will not be changed unless otherwise stated.

In the human body, the attenuation of the sound wave is increased at higher frequencies. Hence, in order to have a better penetration of deeper tissue, the center frequency of excitation is 3MHz. The excitation of our transducer is a broadband Gaussian pulse with 70 percent bandwidth, sampled at a rate of 9MHz which is

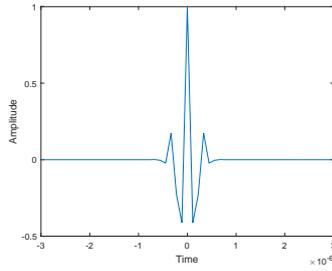


FIGURE 3.1: Excitation of Transducer

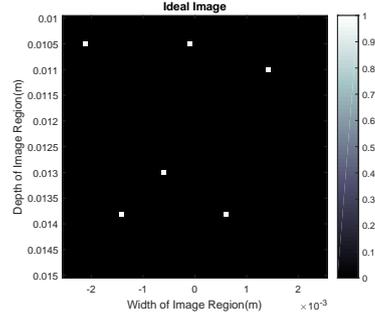


FIGURE 3.2: Location of Point Objects

above the Nyquist sampling frequency, which is shown in Figure 3.1. Then, the wavelength is equal to 0.5mm.

The width of the transducer (the element array) is 1.2cm (contains 40 elements whose width is 0.3mm). The distances between elements are very small and the Image Region is 1cm away from the transducer.

Considering the computational ability of our PC and MATLAB, the Image Region is a square of -0.25cm to 0.25cm on x -axis (width) and 1cm to 1.5cm on z -axis (depth) and the pixel size is 0.1mm (5 pixels per wavelength). It means that we will have 2601 pixels in the final image.

The positions of six chosen point objects, top-down from top left corner of the image, are Point 1 (-0.21cm, 1.05cm), Point 2 (-0.14cm, 1.38cm), Point 3 (-0.06cm, 1.3cm), Point 4 (-0.01cm, 1.05cm), Point 5 (0.06cm, 1.38cm), and Point 6 (0.14cm, 1.1cm).

The initial Sample Shift delay $\tau_{i,Sample\ Shift}$ is selected from the interval $[0, 20]$ because of hardware restrictions. The Phase Shift delay value $2\pi\omega\tau_{i,phaseshift}$ is selected from the interval $[0, 2\pi]$. The Amplitude Weight $\tau_{i,Amplitude\ Weight}$ is a value selected from the interval $[-1, 1]$.

3.2 Non-Compressed Ultrasound Imaging

Before examining the impact of our compression designs on the ultrasound imaging, it is better to consider the cases where no compression has been implemented. There are numerous traditional methods that can be applied to the reconstruction of an ultrasound image. Here, we present two of them, our imaging algorithm with full data from the elements in the array and the *Synthetic Aperture Delay and Sum imaging (SADS)*, as comparisons of our compression-imaging system. The principle of the full data case has been introduced in Chapter 2. So we only briefly explain the concept of SADS here. The core of SADS is to determine the time delays on each element. The delays are caused by relative positions of point objects and elements in the array that is the actual time of flights. The procedure of our method is to locate the positions of objects by measuring the real distance from a specific transmitter element to objects and from objects to the receiver elements. Then all time of flights can be calculated by using these distances and a known speed of the ultrasound wave. Since our received signals are discrete, all time of flights can be transformed into an equivalent number of samples. The corresponding samples of point objects can be tracked separately. The application of Synthetic Aperture technique means 40 elements in an array transmit waves sequentially while all elements are in receive mode after transmission. Then we will obtain 40 sets of data in each experiment.

After that 40 elements will reconstruct images separately. The final image will be the average of these 40 reconstructed images.

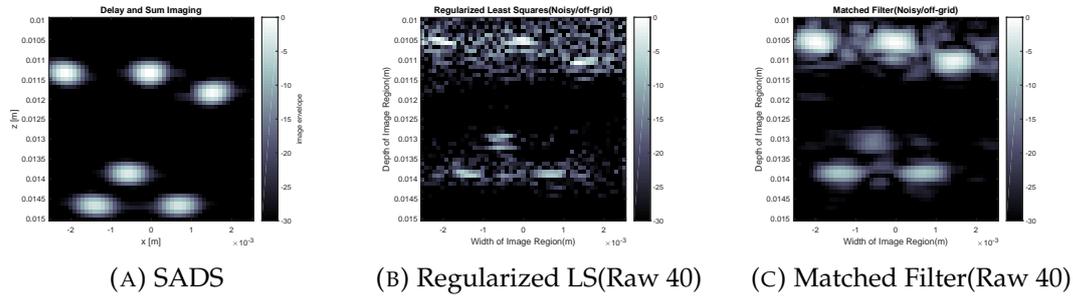


FIGURE 3.3: Non-Compression Ultrasound Imaging(dB mode). Raw 40= Raw data from all 40 elements

In Figure 3.3, the results of SADS and full raw data case with Regularized LS and Matched Filtering are displayed separately. In order to display the details of the solution x , we display all images in dB where the brightnesses of all pixels are amplified.

For the SADS case, all point objects have been reconstructed into lightness spots. At first, the fluctuations of waves are a gradual change, the small surrounding areas of points should be brighter than the background. Secondly, the final image is the mean of 40 low-quality images and the resolution of a single low-quality image is not good. So, the overlapping areas will be remained. The SADS Method does not rely on the prior knowledge of all hypothetical point scatterers (pixels), it tracks the location of point objects by its corresponding sample in the received signal which is hard to be influenced by measurement noise and off-axis of point objects. Hence, we can conclude that SADS is a stable imaging method for point objects. However, the problems of this method are also distinct. Since there is no data compression in the method, 40 channels are required to transmit all data to the post-processing unit in 40 rounds of measurements. The final data size is much larger than the full data case of our imaging algorithm. In addition, the locations of point objects are shifted because the peaks are not always at the beginning of signals. Then more samples have been considered as a part of time-of-arrivals. Then, the whole image is shifted a little bit far away from the expected location.

As for the full data case of our imaging algorithm, six off-axis point objects can be distinguished from the correct locations in both images reconstructed by Regularized Least Squares and Matched Filtering. We can find that the off-axis position of point objects has a huge impact on the quality of reconstruction. The reconstructions are bothered by the background 'noise' in the image. In the condition that point objects are located at the center of the mesh, the solutions of the least squares are no longer accurate. All four vertices of the square mesh have the same distance to the real point object, so any vertex is the probable estimate of the point object. In addition, because of the presence of measurement noise \mathbf{n} , the actual input vector is $\hat{\mathbf{y}} = \mathbf{y} + \mathbf{n}$, where \mathbf{y} denotes the noise-free input.

According to the images in Figure 3.3, we can conclude that the matched filtering method is unable to reconstruct the ultrasound images accurately compared to the Regularized Least Squares. The point objects are represented by lightness spots rather than a few pixels. We have to notice that pixels are pretty close to each other in our image region. Hence, the received signals from them are highly similar. Because the Matched Filtering method is highly dependent on the cross-correlation between

the filter \mathbf{A}^H and the observation \mathbf{y} , the similarities between columns in \mathbf{A} affect the quality of the solution hugely. Even when we use the full data from the entire array, the result of Matched filtering is always formed by lightness spots. Therefore, the performance of Matched filtering is limited in our system. In addition, we find that the brightness of selected point objects is different. It means that the variations of signal strengths in a plane wave transmission also affects the quality of reconstruction.

3.3 Compressed Ultrasound Imaging

In our simulation, only 1 pulse-echo measurement was taken for the specific image region. The reconstructions of full data case have already been discussed in last section. Usually the price of the compression is the loss of information, so the image quality should be lower in our expectation. In the initial compression model, the data can be highly compressed into $\frac{1}{40}$ of the raw data in 1 measurement. In our system, the loss is the reduction of differences between pixels because the summation will remove the temporal properties and amplitude properties of the received signals from different elements. Even though, we have introduced preprocessing methods to make up the loss, the decrease of image quality is inevitable when the raw data are highly compressed. Generally, this kind of negative influence can be improved by increasing the measurement amount. However, in our single measurement imaging algorithm, we can only apply an equivalent alternative by increasing the rounds of compression. The reasons is that whether the initial signals in multiple measurements of our specific image region are similar or not, the 10 rounds of compressions for 10 measurements and for 1 measurement are approximately equivalent when the set of variables τ in preprocessing methods is randomly generated for each round of compression. Then, we can realize the same effect as 10 measurements with 10 rounds of compression of a single measurement.

The iteration of compression can be archived by using a structure of multiple parallel compression units or adding temporary storage space. The negative impact is that the data size of the compressed data will be also enlarged with the increase of iteration amount. Hence, we decide to apply 10 rounds of compression as a trade-off of the image quality and the degree of compression. The relation between the compression rounds and the quality of results will be further analyzed in the performance analysis section.

Then, the compressed measurement vector \mathbf{y}_C and the compressed matrix \mathbf{A}_C can be described as ten sets of $\mathbf{y}_{C,1}$ $\mathbf{y}_{C,2}$... $\mathbf{y}_{C,10}$ and $\mathbf{A}_{C,1}$ $\mathbf{A}_{C,2}$... $\mathbf{A}_{C,10}$ that are stored in a vector and matrix vertically, which are

$$\mathbf{y}_C = \begin{bmatrix} \mathbf{y}_{C,1} \\ \mathbf{y}_{C,2} \\ \vdots \\ \mathbf{y}_{C,10} \end{bmatrix} \quad \mathbf{A}_C = \begin{bmatrix} \mathbf{A}_{C,1} \\ \mathbf{A}_{C,2} \\ \vdots \\ \mathbf{A}_{C,10} \end{bmatrix} \quad (3.1)$$

Let us take the first set $\mathbf{y}_{C,1}$ and $\mathbf{A}_{C,1}$ as an example to show the procedure of our data acquisition. Matrix $\mathbf{A}_{C,1}$ is constructed column by column, where each column represents a pixel in the image region. By setting a point object in different positions on a preset grid that covers the whole image region, we will repeat the process as shown in Figure 2.2. The output of the system each time is a compressed signal represent a specific pixel in the image region, which will be stored in 1 column of

$\mathbf{A}_{C,1}$. After the matrix $\mathbf{A}_{C,1}$ has been constructed, we should measure the vector $\mathbf{y}_{C,1}$ with the same process. The difference is that the process in Figure 2.2 will only be implemented once for 6 point objects rather than 1. Then, the first set $\mathbf{y}_{C,1}$ and $\mathbf{A}_{C,1}$ is obtained. If we want to implement 10 rounds of compression, it means that the same above procedure should be repeated 10 times during the building of \mathbf{A}_C and the set of preprocessing variables τ is different in each compression round. In practical observation, the acquisition of \mathbf{y}_C is done by the same process, where the row dimension of each $\mathbf{y}_{C,r}$ need to be aligned with the corresponding $\mathbf{A}_{C,r}$, $r \in \{1, 10\}$.

The ultrasound images reconstructed with Regularized Least Squares and Matched Filtering with 10 rounds of compression are shown in this section. The results of three preprocessing methods are placed side by side for comparison.

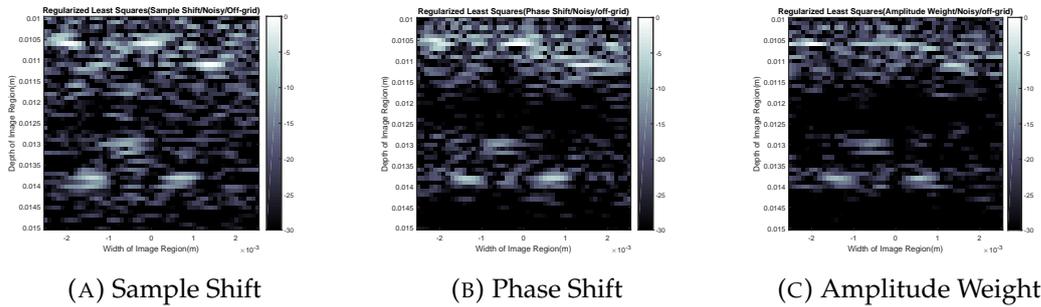


FIGURE 3.4: Regularized Least Squares (Off-axis Point Objects/Noisy/dB mode)

Figure 3.4 shows that the image quality is decreased a lot after data have been compressed. More background 'noise' has been introduced and the shape of estimated point objects are irregular. The singular value decompositions of all three matrices \mathbf{A} show that they are not full rank. It means our least squares problem turns out to be ill-posed or under-determined. Hence, the effect of regularization is significant here. We try both Tikhonov Regularization and LSQR methods, their results are pretty similar if regularization parameters have been chosen carefully. From these images, all three preprocessing methods have their own advantages and disadvantages. By contrast, the phase shift method seems better than the other two methods. In general, all results are still acceptable, because the lightness spots are still in the correct locations of the point objects.

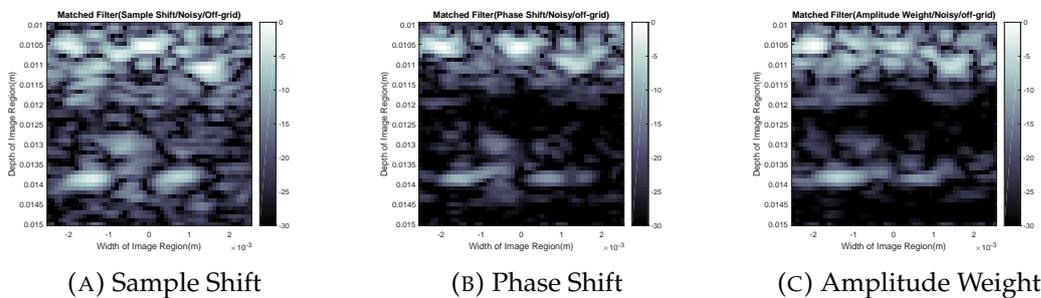


FIGURE 3.5: Matched Filter (Off-axis Point Objects/Noisy/dB mode)

Images reconstructed by the matched filter in Figure 3.5 are also worse than the reconstruction of full data from the entire array. As aforementioned, the Matched Filtering is unable to distinguish adjacent pixels, so the lightness spots are larger.

Hence, the background noise has been zoomed in because of it. Among all three methods, the phase shift method is the closest one to the full data case.

In the Matched Filter result, we find that the point objects have been reconstructed into lightness spots. The spread of these lightness spot is called the *Point Spread Function (PSF)*. PSF has shown the fact that a mathematical point reflector in the object plane is spread out to form a finite area in an image plane, which occurs when imaging objects are point-like. The degree of spreading (blurring) of the point object is a measure for the quality of an imaging system. Said differently, PSF is the correlation between a specific pixel and the whole image region. It is the matched filter for a single point object in the ideal case, where the observation \mathbf{y} equals the corresponding column of a specific pixel in \mathbf{A} . For an on-axis point i , the PSF can be described by the formula

$$PSF(i) = \mathbf{A}(:, i)^H \mathbf{A} \quad (3.2)$$

The intention is to use it as a kind of evaluation of \mathbf{A} here.

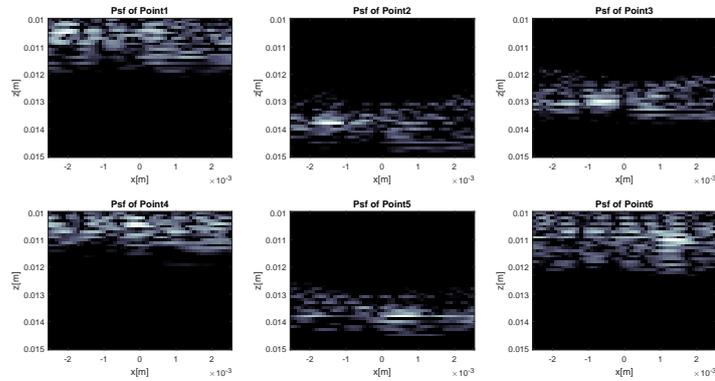


FIGURE 3.6: Point Spread Functions of All Point Objects (Sample Shift (dB mode))

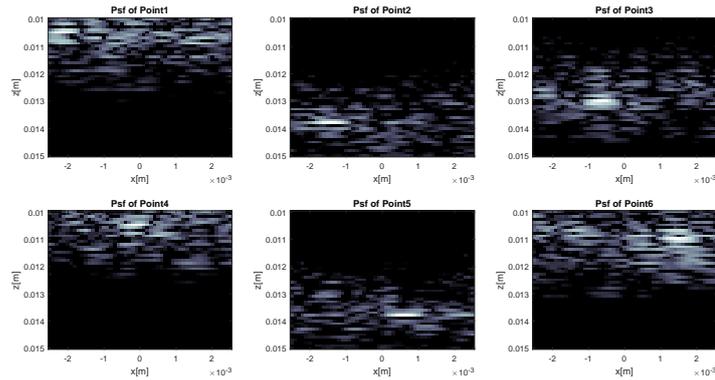


FIGURE 3.7: Point Spread Functions of All Point Objects (Phase Shift (dB mode))

In all three figures (Figures 3.6, 3.7 and 3.8), it shows that all point objects are also reconstructed into lightness spots as the matched filtering. As we have explained, all results are displaying the ideal matched filtering images of a single on-axis point that we can obtained from the compressed data. We found that all point scatterers are also surrounded by interference in adjacent depths. Depending on these figures, the performances of three preprocessing methods are too close to call. The influence

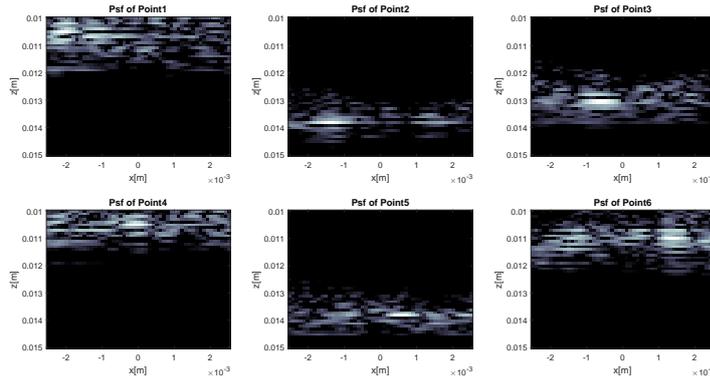
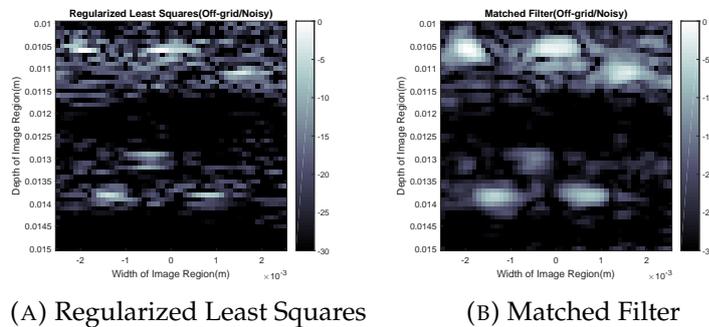


FIGURE 3.8: Point Spread Functions of All Point Objects (Amplitude Weight (dB mode))

of the inaccurate estimation is distinct. We found that all points are distinguishable because they have the highest brightness in their own image because the point object has the highest correlation with itself.

In order to examine the performance of our system, we have to find a good comparison of it. Since we do 10 rounds of compression, the data size of output is only $\frac{1}{4}$ of the raw data from 40 elements. So, we should consider a case that the reconstruction is done with the raw data from only 10 elements, which has same compression degree as our system. The 10 elements are distributed in the range of the entire array, they are element 1, 5, 10, 14, 18, 23, 27, 31, 36, 40. In Figure 3.9, the



(A) Regularized Least Squares

(B) Matched Filter

FIGURE 3.9: Raw data from 10 elements(Off-axis Point Objects/Noisy/dB mode)

Regularized Least Squares and Matched Filtering result with the raw data from 10 elements have been shown. We can find they are pretty close to or better than the results we have obtained with our compression-imaging system. It means that our compression scheme needs further improvement from the consideration of practical implementation. It is true that many factors may influence the image quality, but the role of the preprocessing methods should be the most important among all factors. Therefore, we have to investigate them more carefully in order to find out the improvement methods.

3.4 The Potential Improvement Methods

As we have shown in the last section that the idea of our compression scheme is applicable, but improvements are required if we want to implement it. There are two

possible solutions. The first one is to narrow down the grid spacing size as small as possible to weaken the impact of the off-axis error. This is a solution which can improve the image quality of any kind of method. The second one is an improvement of the Sample Shift method by enlarging the maximal delay in samples.

3.4.1 Smaller Pixel Size

Since we always consider the worst case, the off-axis point objects should be in the center of the mesh where they are farthest away from the axis. Then, the pixel size (the distance between pixels) may be a potential factor that influences the performance of reconstruction, because the off-axis point objects will be closer to the axis when pixel size is smaller. Our assumption is that the preset sensing matrix \mathbf{A} only contains the information of all on-axis pixels. The error caused by a point offset should be weaker when the point objects are closer to the axis.

Considering the computational ability of our PC, we decide to cut out a small area of the original image to illustrate the effect of pixel size reduction. This small area includes the two left-bottom points (Points 2 and 3) in the original image as shown in Figure 3.10. The image region is fixed during the simulations, while the pixel size is decreased from 0.1mm to 0.025mm . The columns of the matrix \mathbf{A} correspond to pixels of the image. Hence, the matrix \mathbf{A} will be wider with an increase of the pixel number. When the pixel size is 0.025mm , our largest pixel number is 961.

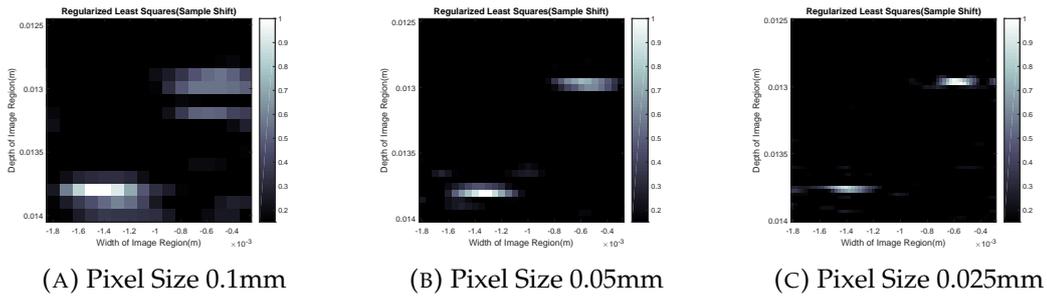


FIGURE 3.10: Regularized Least Squares (Sample Shift/Off-axis Point Objects/Smaller Pixel Size)

As shown in Figure 3.10, the improvement of the image quality is obvious. The lightness spots become more concentrated with a decrease of the pixel size. In other words, the regularized least squares estimates can be more accurate. We can conclude that a smaller pixel size is a useful method to overcome the off-axis problem. However, for a fixed image region, a smaller pixel size means more pixels in this region which requires a pretty high consumption of resources. The data size will increase rapidly with growing pixel numbers. Another disadvantage of the this method is that it cannot be used when point objects are pretty close to each other. When pixels are too close to each other in space, they will be highly similar to each other, which means they are hard to be imaged. It's why we do not display the matched filtering results because they are always large PSFs. Consequently, the 0.1mm is a wise choice for image region of $5\text{mm} * 5\text{mm}$ with consideration of performance of our PC.

3.4.2 Larger Delay Range

Narrowing down the pixel size is a strategy that we can apply to overcome the impact of off-axis point objects. However, the improvement of our compression scheme

is more critical in our research. The time delay is a factor we can consider carefully. When this maximal delay is large enough, the received signals from elements can be shifted into their unique interval in time space. The summation of these delayed signals is the same as the storage of the full data. Hence, we can enlarge the maximal time delay to observe the change of image quality. But, there should be an upper bound of the maximal delay. However, it is impossible to enlarge differences between phase shift delays as sample shifts. The received ultrasound signals are wide-band signals where the corresponding time delays of the same phase shift for varied frequencies are different. In addition, the phase shift is limited to the range $[0, 2\pi]$ because of the periodicity. As a result, the phase shift only changes the envelope of the wide-band signal rather than shift the signal in the time domain. The details will be discussed later. Therefore, we will only consider the influence of the size of the maximal sample shift. Furthermore, the pixel size is 0.1mm, which is the same as the initial setting.

In order to illustrate the influence of the Maximal Delay on the Sample Shift method, we display the reconstructions of the cases that the Maximum delays are 50, 100, 150 and 200 samples. Furthermore, the pixel size is 0.1mm, which is the same as the initial setting.

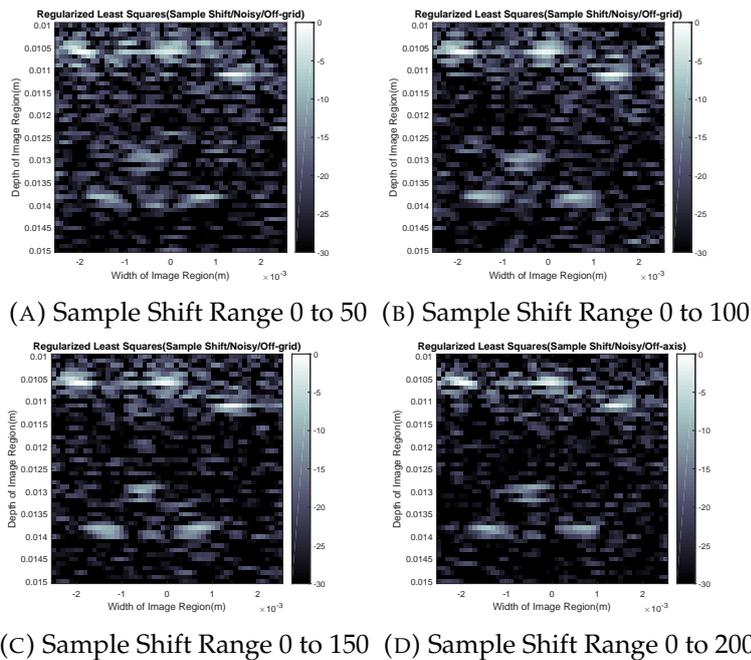


FIGURE 3.11: Regularized Least Squares (Off-axis Point Objects/Noisy/Larger Maximum Delay)

From a subjective point of view, all images in Figure 3.11 are better than when the maximal delay is 20 samples, where lightness spots of point objects are concentrated gradually and the background 'noise' is also smaller. The results are also better than the reconstruction results with the raw data from 10 elements. When the maximal delay can be 200 samples, the SVD of \mathbf{A} indicates that \mathbf{A} is full rank. Hence, we can conclude that the enlargement of the maximal delay is able to improve the quality of reconstruction. However, when the maximal delay is too large, the complexity of hardware implementation is hard. In addition, if we consider when the delay is extremely large, signals can be stored in different time spaces, which means that

equal the full data case. Therefore, a proper value of the maximal delay should be chosen. It will be discussed later in the performance analysis section.

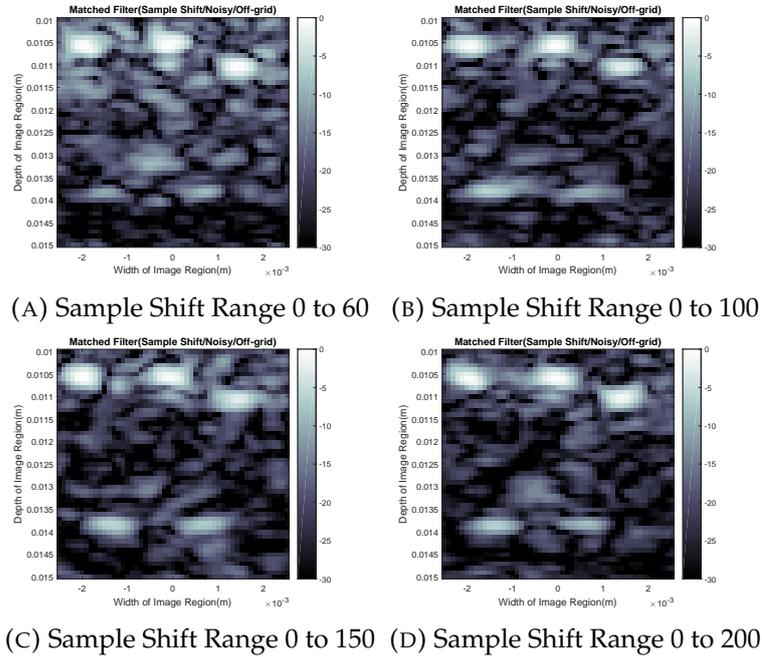


FIGURE 3.12: Matched Filter (Off-axis Point Objects/Noisy/Larger Maximum Delay)

From Figure 3.12, we found that the matched filtering results are also becoming better with the enlargement of the delay range. But the improvement is limited due to the bad performance of the matched filter. Hence, we can find that the three matched filtering results are too close to call when the maximal delay is above 100 samples.

3.5 Performance Analysis

In practical implementations, many factors may affect the performance of reconstruction, the efficiency and cost of the system. During the evaluation, more information about these factors are acquired, some of them can be helpful to the further design. In our compression imaging system, the role of the matrix \mathbf{A} is significant because it's the prior knowledge of the target image region we have before the measurement. Therefore, the condition of it is an important index when we evaluate our system.

The evaluation of our system can be done by calculating the *condition number* of \mathbf{A} , which can reflect the sensitivity of a function to the errors and changes in input. The order of the condition number magnitude indicates how much noise and errors are amplified.

The effects of three impact factors will be assessed in this section: the maximum delay, the number of measurements (or compression rounds) and the number of elements per wavelength. All other parameters are kept as the same as their initial setting in section 3.1. The parameters we have changed in the evaluations are that 15 sets of measurements (or compression rounds) are obtained to construct \mathbf{y} and \mathbf{A} . The pixel size will be 0.2mm and the size of the image region is 5mm by 5mm. Hence, less pixels are in the image region, which can reduce the time consumption.

There is only 1 variable that will be changed in each evaluation, which will be the only impact factor of the condition number variation.

The first fact we want to illustrate is the relation between the magnitude of maximum random delay and the condition of the sensing matrix \mathbf{A} . Since we have already found that image quality can be improved with the increase of the maximum delay, we can assume that there will be improvement of the condition number as we increase the maximum delay. The best case is that condition number equals 1. In this part, we will only consider the Sample Shift as we have explained before.

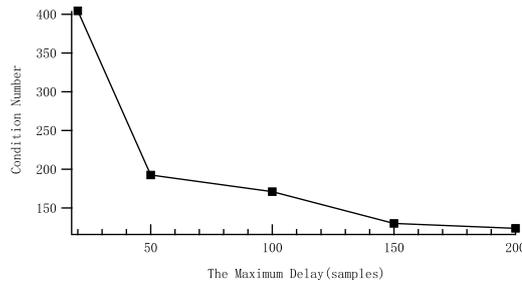


FIGURE 3.13: Condition Number variation with the Maximum Delay (Sample Shift/Linear Scale)

In Figure 3.13, as the average of 6 simulations, the condition number is monotonically decreasing with the growing of the maximal delay. However, the variation becomes slower when the maximum delay is large enough. It fits our previous observations of the image quality improvement. Then the value of the corner point will be chosen as the trade-off between quality and efficiency. The condition number is small enough here to ensure that the sensing matrix \mathbf{A} is in good condition while the data size of the compressed signals are acceptable.

The table of averages and standard deviations of different condition numbers are shown below:

Maximum Delay	Average	Standard Deviation
20	404.32	33.00
50	192.60	6.45
100	171.09	4.87
150	130.15	3.17
200	123.75	3.96

TABLE 3.1: Average and Standard Deviation of Condition Number (Sample Shift/Linear scale)

According to table 3.1, the standard deviations of the condition numbers are also decreasing like the average. It shows that the condition of \mathbf{A} is not only more reliable, but also more stable with larger differences in delays between sensors. Based on these condition numbers, a delay range from 50 to 100 is a better choice than others on account of quality and cost. However, if we consider the compression degree of the signals, 50 samples would be a better choice according to the condition numbers. It is true that this maximal delay interval may be not a wise choice from a hardware's view. However, it is an optimal value for this compression according to the simulation, so it will be applied in further evaluation.

As aforementioned, the number of measurements (or compression rounds) can affect the quality of reconstruction. Pixels can be differentiated well when more measurements have been obtained. However, the number of measurements is also

related to the data size of the compressed signals. If we have 40 measurements in our simulation, the size of compressed signals should be equivalent to the full data from 40 elements, where the compression is meaningless. At the same time, the size of the data can also influence the time consumption and computational complexity. Therefore, the number of measurements should be constrained by an upper bound when a matrix \mathbf{A} is in a good condition. Six simulations have been done for precise outcomes like before. All three differentiation methods: Sample shift, Phase shift and Amplitude Weight will be applied separately to be the comparison of each other.

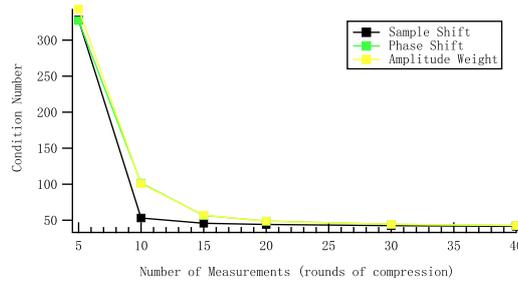


FIGURE 3.14: Condition Number Variation with Number of Measurements (Three preprocessing methods/dB Mode)

Figure 3.14 shows that the condition numbers of \mathbf{A} are varying with the number of measurements, where three differentiation methods have been applied to the received signals separately. Firstly, we can conclude that the improved sample shift case is the best among all three methods because the condition number of it converges to 1 faster than the other 2 methods. But this difference between them will be reduced with the increase of the measurements number. In general, the growing amount of measurements can effectively improve the condition number.

The table of averages and standard deviations of different condition numbers are shown below:

Number of Measurements	Average Condition Number	Standard Deviation
5	$2.6 * 10^{16}$	$3.8 * 10^{14}$
10	453.3	39.5
15	192.6	6.5
20	159.6	5.5
30	131.1	3.8
40	118.7	4.9

TABLE 3.2: Average and Standard Deviation (Sample Shift/Linear scale)

Number of Measurements	Average Condition Number	Standard Deviation
5	$2.3 * 10^{16}$	$3.5 * 10^{15}$
10	$1.2 * 10^5$	$3 * 10^4$
15	676.5	92.5
20	272.3	21.2
30	162.8	10.9
40	137.1	13.3

TABLE 3.3: Average and Standard Deviation (Phase Shift/Linear scale)

Number of Measurements	Average Condition Number	Standard Deviation
5	$1.5 * 10^{17}$	$3.4 * 10^{16}$
10	$1.1 * 10^5$	$2.8 * 10^4$
15	712.3	61.3
20	267.2	17.3
30	164.3	12.1
40	134.6	11.7

TABLE 3.4: Avere and Standard Deviation (Amplitude Weight/Linear scale)

Tables 3.2, 3.3 and 3.4 give us the relations between condition numbers and the amount of measurements with three different pre-processing methods. For all Sample Shift delay, Phase Shift delay and Amplitude Weight methods, reliable results are also more stable according to the simulation results. The conclusion is that 15 measurements are enough to construct matrix **A** in good condition in the consideration of a smaller data size.

The number of sensors in the transducer array is another influential factor similar to the number of measurements. Both of them can affect the data size of the compressed signals. If we consider our imaging algorithm with full data from the entire array, more unique details of a specific pixel can be acquired and stored if we put more elements in the array. The qualities of reconstructions should be improved by the extra information. In our compression imaging system, it means that more information can be contained in the compressed signals. We assume that the condition of **A** can be also improved as the reconstruction with full data. Since the element array size should be fixed to 1.2cm, the sensor size is inversely proportional to sensor quantity. The size of each element will be smaller if we put more elements in the array. We decide to show the variation of the condition number with the increase of the number of sensors. Finally, the scale of the x-axis is shown in the number of sensors per wavelength.

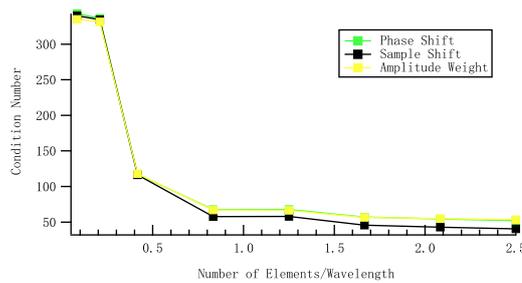


FIGURE 3.15: Condition Number Variation with Number of Sensors (dB Mode)

Since some values are much larger than the others, we transfer all values into dB mode for plotting where the differences can be easily shown in Figure 3.15. The decreasing curves of all three differentiation methods are really close. It means that the impacts of the sensors amount on the three methods is the same meaning that more sensors are helpful to improve the condition of matrix **A**.

The table of averages and standard deviations of different condition numbers are shown below:

Number of Sensors	Average Condition Number	Standard Deviation
2	$9.7 * 10^{16}$	$5.1 * 10^{15}$
5	$5.2 * 10^{16}$	$1.2 * 10^{15}$
10	$6.5 * 10^5$	$4.1 * 10^4$
20	770.7	30.6
30	784.1	30.8
40	192.6	6.5
50	138.1	7.1
60	105.7	7.0

TABLE 3.5: Average and Standard Deviation (Sample Shift/Linear scale)

Number of Sensors	Average Condition Number	Standard Deviation
2	$5.5 * 10^{16}$	$5.1 * 10^{15}$
5	$3.7 * 10^{16}$	$2.5 * 10^{15}$
10	$7.9 * 10^5$	$8.9 * 10^4$
20	$2.2 * 10^3$	214.1
30	$2.1 * 10^3$	364.3
40	676.5	92.5
50	555.1	97.2
60	477.8	66.5

TABLE 3.6: Average and Standard Deviation (Phase Shift/Linear scale)

Number of Sensors	Average Condition Number	Standard Deviation
2	$1.4 * 10^{17}$	$9.2 * 10^{15}$
5	$6.6 * 10^{16}$	$6.4 * 10^{15}$
10	$7.4 * 10^5$	$1.5 * 10^5$
20	$2.4 * 10^3$	331.3
30	$2.4 * 10^3$	238.8
40	712.3	61.3
50	523.6	76.3
60	406.1	52.9

TABLE 3.7: Average and Standard Deviation (Amplitude Weight/Linear scale)

The variations of the condition number in tables 3.5, 3.6 and 3.7 also show that condition number can be smaller while there are enough elements. However, we have to consider the manufacturing complexity of hardware. So, the elements number of a single transducer cannot be infinity. Here we would like to use 40 elements, which is enough to reconstruct acceptable ultrasound images in our system.

Combining those simulation results, we can conclude that an element array with 40 sensors, 15 measurements and delays up to 50 samples is an optimal set of parameters for our current design and simulation environment. Except for the Larger Maximum Delay in the Sample Shift case, two other factors can be used to promote the performance of the entire system implement of the preprocessing method.

3.6 Discussion of Three Preprocessing Methods

In previous simulations, we have implemented three types of preprocessing methods to differentiate the received signals. These preprocessing methods have a direct impact on the performance of our compression scheme. However, we have found that they have their own advantages and disadvantages. So, it would be helpful to research the potential cause of this if we plan to perfecting our system.

3.6.1 The Defective of Phase Shift

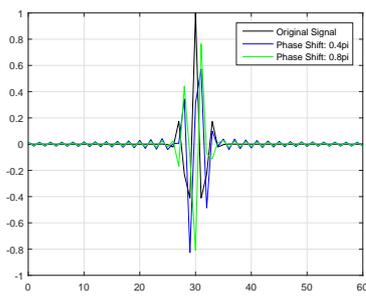
Since our ultrasound signal is a broadband signal, it is important to note that phase shift is not a reliable method to delay a broadband signal. Since phase shift should be applied in the frequency domain, then we should study this problem from the Fourier Transform function

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j2\pi\omega t} dt \quad (3.3)$$

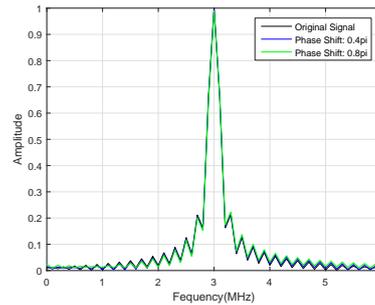
where ω is frequency and $\mathbf{f}(t)$ is denoted by $\hat{f}(\omega)$ in the frequency domain.

For element i , applying any real number delay τ to it in the time domain, $e_i(t) = f(t - \tau)$, then it turns out to be $\hat{e}_i(\omega) = e^{-j2\pi\omega\tau}\hat{f}(\omega)$ in the frequency domain. The phase angle is $\theta = 2\pi\zeta\tau$, which should be restricted in the interval $[0, 2\pi]$. It shows that signals at different frequencies will have diverse time delays for one specific phase shift. Theoretically, for a broadband signal which consists of a large group of frequency components, the waveform can be retained when its frequency component is in proper phase alignment with each other. Instead of delaying the signal in the time domain, a fixed phase shift for all frequencies will break down this kind of alignment and cause delay distortion. As a result, phase delay is also known as envelope delay, which is the rate of change of the phase versus frequency curve.

We can illustrate this conclusion by shifting our excitation signal, which is a broadband signal.



(A) Phase Shifted Excitation Signal



(B) Envelop of Phase Shifted Excitation Signal

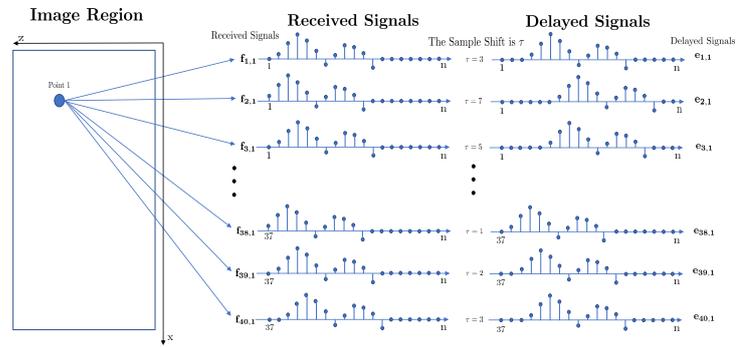
FIGURE 3.16: Effect of Phase Shift

Figure 3.16 shows that the pulse was not actually shifted in time domain. Due to different time delays of varied frequency signals, the original waveform becomes distorted, where side lobes are generated as a kind of noise. However, when we examine the envelop of the shifted signal, it shows that signals are remained in a fixed interval. It means that the Phase Shift is similar to the Amplitude Weight to some extent, which change the waveform the initial signals. Although, its performance

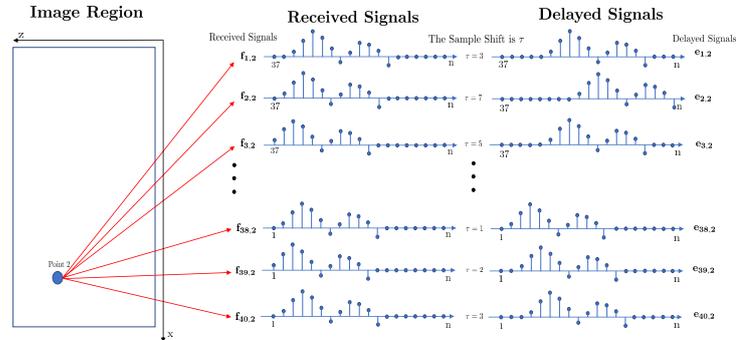
is acceptable as we have seen previously, it is still not an appropriate preprocessing method for a broadband signal since not further improvement can be achieved.

3.6.2 The Advantages and Disadvantages of Sample Shift

Compared to the Phase Shift, the Sample Shift can achieve a time delay for both narrowband and broadband signals. We have already mentioned that, when the Sample Shift is large enough where the signals from all elements can be shifted to their unique intervals, the compressed data is equivalent to storing the full data from all elements. So, it is valuable for research. Let us use 2 points in the image region, which are symmetric relative to the transducer, as the example to explain the difference between the Sample Shift and the other two methods. The details can be illustrated by Figure 3.17.



(A) Sample Shift Scheme of Point 1



(B) Sample Shift Scheme of Point 2

FIGURE 3.17: Sample Shift Scheme of Two Symmetric Points

In Figure 3.17, we assume the difference of time-of-arrivals between 2 adjacent elements is 1 sample and the pulse shapes of signals are the same for the convenience of illustration. This assumption is only partially true in reality.

We mark these two points as Point 1 and Point 2. Since the construction of matrix \mathbf{A} is done separately for each pixel, we will also investigate these 2 points 1 by 1. When the element array receives the spatial impulse responses of Point 1 ($\mathbf{f}_{i,1}$ $i \in \{1, 2, \dots, N\}$), the received signals of all elements will be delayed differently in time domain. The delayed signals ($\mathbf{e}_{i,1}$ $i \in \{1, 2, \dots, N\}$) will be summed up into 1

output. As for Point 2, the received signals ($f_{i,2} \ i \in \{1,2,\dots,N\}$) of the elements array are a mirror of Point 1 ($f_{i,1} \ i \in \{1,2,\dots,N\}$). For instance, a signal is received by the 1st element in the array when measuring Point 1 will also be obtained by the 40th element in the array when measuring Point 2, which is shown in Figure 3.17. Therefore, the Naive Compression is unable to distinguish them because these 2 points have the same compressed signal after summation. However, if we apply the same set of delays to these two measurements, the symmetric pattern between 2 points is destroyed. The summation of the delayed signals in 2 cases cannot be the same unless the delay set of the entire array is also totally symmetric. However, when we consider the Amplitude Weight and Phase Shift methods, both these methods will only distort the signals in their original time intervals. Since the distances between elements are small, the time intervals of all received signals are highly overlapping with each other. In addition, the summation of 40 highly overlapping signal pulses is hard to be controlled. Hence, Points 1 and 2 are probably represented by similar signal pulses even if the set of amplitude weights or phase shifts is not symmetric. As a consequence, the Sample Shift is a more hopeful method to distinguish a pair of mirrored points which is the main interference caused by linear summation.

However, every coin has two sides. The differences between highly similar pixels have been enlarged, but the similarities between previously uncorrelated pixels become higher simultaneously. When the maximum sample shift is not large, the probability that the summations of delayed signals share the same time interval is pretty high. In this case, the Sample Shift is worse than the other two differentiation methods because it did not distort the waveform of signals. From Figures 3.4 and 3.5, we can find more background 'noise' in the image when the maximal delay is up to 20 samples. Another problem of the Sample Shift is the trade-off between the high compression degree and high image quality. Based on the above displays and analysis, we can conclude that the Sample Shift can perform better than the other two methods when the maximal delay has been chosen appropriately.

3.7 Summary

Our compression scheme for ultrasound imaging is applicable for off-axis point objects in a noisy environment. Compared to the imaging with the received data from all elements, the image quality has decreased a lot due to the loss of information in compression. But the quality of the reconstructions cannot meet our expectation except for the Sample Shift case with a larger maximum delay because they cannot beat the case where we use raw data from 10 elements. Among the three preprocessing methods, the Phase Shift is unable to be promoted due to the restriction of periodicity if it has been used to delay broadband signals, so we will not consider it in the later designs. The advantage of the Sample Shift method is its simplicity and room for promotion. Meanwhile, unlike the Phase Shift, the Amplitude Weight does not have the critical effect, the only problem is how to use it properly. Hence, we will implement it differently in the following research.

In addition, another potential hidden trouble of our design can be found from this fact that the compressed signals of symmetric pixels or pixels at the same depth are the same if no preprocessing has been done. During the discussion, we have mentioned that all three preprocessing methods can eliminate the similarities between pixels. We did ten rounds of compression to suppress the impact of the highly compressed raw signals. The final result is the combination of them. However, if the

outcomes of ten rounds of compression share the same errors, then the errors will be amplified 10 times if the result is a combination of them.

Chapter 4

Case 2: An Improved Design for a 18.5MHz Ultrasound System

4.1 Introduction

As we have discussed in the last chapter, our initial design cannot meet our expectation. A new design is required based on a part of information we have obtained. In this chapter, we will focus on the changes in 2 factors. The first one is the architecture of the system. In Chapter 3, all received signals will be processed and summed up into 1 output signal, the size of the output signals is only $\frac{1}{40}$ of the raw data from 40 channels in each round of compression. Even if we need 10 to 20 rounds to achieve high-quality reconstruction, the final compressed data are $\frac{1}{4}$ to $\frac{1}{2}$ of the raw data from 40 channels in data size. The intention of multiple rounds of compression is to enlarge the differences between the corresponding compressed signals of pixels. The realization of multiple rounds of compression in hardware can be parallel compression units or more memories for the temporary storage of data. In hardware design, parallel compression units should be a better strategy than memory. However, the number of elements connected to each compression unit is worth studying. Therefore, a new compression architecture can be designed to reduce the connections and ensure the quality of the reconstruction result simultaneously. The new compression architecture that will be discussed in this chapter is a potential solution where the spatial property of data can be also exploited. The idea is that data from 64 channels can be compressed by groups rather than an entirety. Our design is to divide 64 elements into 8 sub-arrays, where each sub-array contains 8 different elements physically. The data from 8 elements in different sub-arrays are compressed separately with the same model as in Chapter 2. Then, the data size of the compressed signals should be $\frac{1}{8}$ of the raw data from 64 channels in 1 round of compression. If we only consider the data size of the compressed signals, the reconstruction results of the new architecture should be similar to 8 rounds of the previous compression design, where the connection between compression units and elements is only $\frac{1}{8}$ compared to before. The structure of the new architecture will be introduced in detail in the following section of this chapter, and the feasibility of the new architecture will be analyzed. Except for a new architecture, some changes will be also made in preprocessing methods. According to the previous simulation results, we believe that all three preprocessing methods are not perfect for our system. In this chapter, we will try to research improved or alternative solutions for them. We hope these improvements can help us obtain a better result.

This chapter will be organized as follows. In section 4.2 the backgrounds of the new architecture will be explained. The update of preprocessing methods is introduced in section 4.3. Section 4.4 is mainly focused on the evaluation and analysis of

our new designs. Section 4.5 contains the discussion of advantages and disadvantages of all kinds of methods. A short summary is given in section 4.6

4.2 New Compression Architecture

As aforementioned, our system will be transformed into an array with 8 sub-arrays and 64 elements in total. In the performance analysis section in chapter 2, we have already discussed that more elements in the array allows improve the condition of our system. So, the system model should provide us with better results when the element number is increased from 40 to 64. Another significant effect of this new 8 sub-array architecture design is related to the basis of our compression scheme. The essence of our compression is to represent pixels uniquely with fewer data as well as with full data from the entire array. The uniqueness is mainly created by the relative temporal information and the pulse shape of a pixel to the elements in the array. However, the linear summation model of our compression scheme damages these properties. The design of 8 sub-arrays can partially preserve the temporal information in the received signals.

A problem of this new architecture is how to group elements, which is related to the exploitation of the spatial property. There are 3 schemes we would like to propose. The easiest way is to group elements in sequence, where 8 mutual adjacent elements will be drawn in the same group. Then 8 element groups will contain elements 1 to 8, 9 to 16, ..., 57 to 64 separately. It can be regarded as if 8 larger size elements have been used in the array rather than 64 smaller elements.

Since there does not exist a compulsive rule of grouping, random grouping is a good choice when a specific scheme of grouping is hard to be determined. Hence, the second scheme is to group 64 elements randomly into 8 groups, resulting in no regular format of element arrangement in this form. The main problem of this architecture is its uncontrollability, and finding the upper and lower bound of its performance. However, it can be a good reference if we aim at studying the robustness of a system.

The last scheme is choosing an element from each group in the first architecture to form a new group. For example, the first group of elements consists of element 1, 9, 17, 25, 33, 41, 49, 57. Finally, 8 groups of elements will be identical copies of each other with different offsets on the x-axis which is a kind of interleaving model. Then, the array has been transformed into 8 sub-arrays in a linear sequence.

We call the three types of grouping as Architecture 1, 2 and 3 separately. To illuminate the details of the three new architectures of compression intuitively, we use three figures to display them. I have also simplified our system into signal sources plus receivers. The principle is same as we have discussed in chapter 2.

As shown in Figures 4.1, 4.2 and 4.3, the received signals will be processed firstly and summed by groups later. Then our observation vector \mathbf{y} or specific columns of matrix \mathbf{A} will become equation 13, where the storage of the compressed signals from sub-arrays is the same as the multiple rounds of compression in chapter 3

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_8 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_8 \end{bmatrix} \quad (4.1)$$

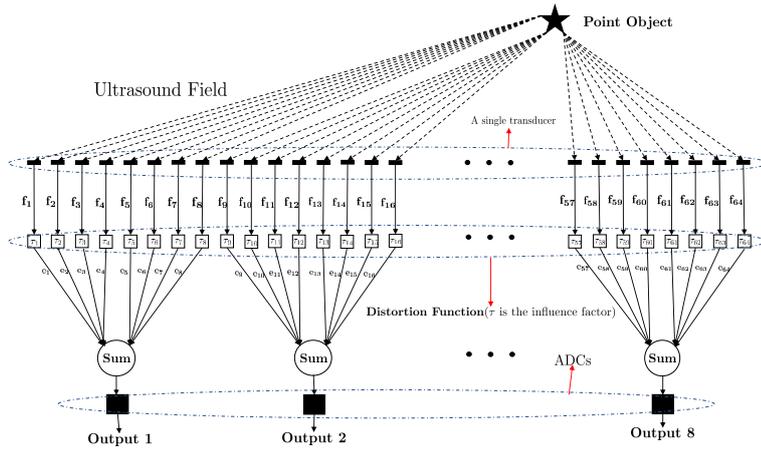


FIGURE 4.1: Schematic Diagram of Architecture 1. The distortion functions include the Sample Shift (τ is the number of delayed samples), the Amplitude Weight (τ is a sequence of gains for all samples) and the Random Subsampling (τ is the multiple of sampling frequency). Observation vector \mathbf{y}_i equals a column of \mathbf{A} when only one pixel is measured

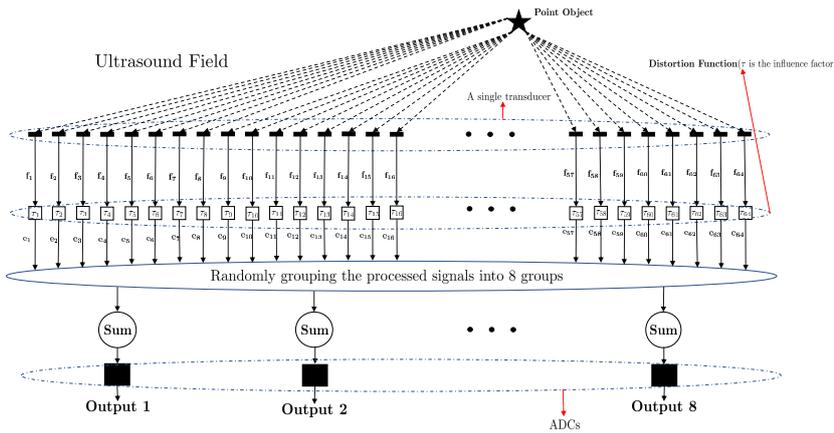


FIGURE 4.2: Schematic Diagram of Architecture 2. The distortion functions include the Sample Shift (τ is the number of delayed samples), the Amplitude Weight (τ is a sequence of gains for all samples) and the Random Subsampling (τ is the multiple of sampling frequency). Observation vector \mathbf{y}_i equals a column of \mathbf{A} when only one pixel is measured

where \mathbf{y}_i is the compressed signal of observations from sub-array i , $\mathbf{A}_i = [\mathbf{a}_{i,1}, \mathbf{a}_{i,2}, \dots, \mathbf{a}_{i,j}, \dots, \mathbf{a}_{i,2601}]$ is the measurement matrix constructed by the compressed signals from sub-array i .

Before we apply any other preprocessing methods to differentiate the received signals from different elements, we can also consider the Naive Compression with the new architecture. The received signals from 8 elements in each sub-array are summed up directly without preprocessing. In Chapter 2, when we sum up the raw data from the entire array, extra rounds are meaningless for our single measurement imaging system if we do not implement any preprocessing methods. But the compression system in the new architectures would provide us with different results

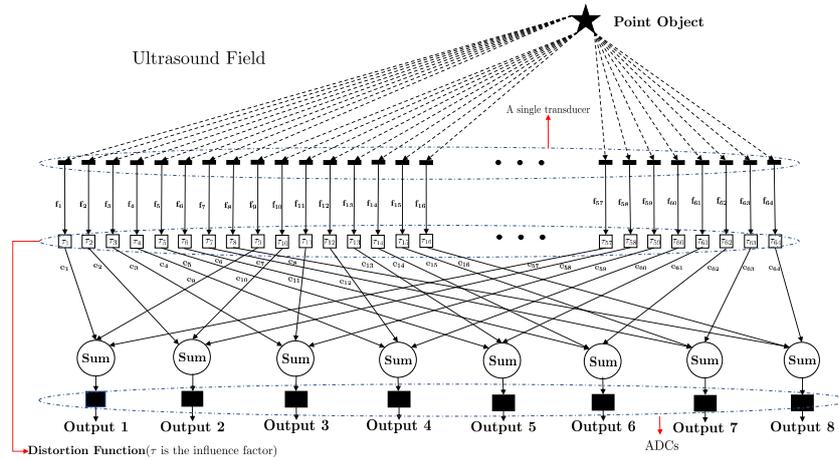


FIGURE 4.3: Schematic Diagram of Architecture 3, the distortion functions include the Sample Shift (τ is the number of delayed samples), the Amplitude Weight (τ is a sequence of gains for all samples) and the Random Subsampling (τ is the multiple of sampling frequency). Observation \mathbf{y}_i equals a column of \mathbf{A} when only one pixel is measured

from images of curves. The actual effects of the 3 architectures will be discussed in the later section with their corresponding reconstructions. This Naive Compression method is a good reference to examine the effects of preprocessing methods and the architecture of the compression system.

4.3 Updated Preprocessing Methods

Four preprocessing methods will be used to improve the performance of compression, which are the Sample Shift, the Amplitude Weight, the Random Subsampling and the Random Grouping sample by sample (RandgrpSbyS). Among them, the Sample Shift method is kept the same as before meaning that the received signals from elements will be delayed in the time domain differently. Due to the restriction of hardware implementation, the maximal delay has been constrained less than 20 samples.

In the last chapter, when we apply the Amplitude Weight, it is a fixed value τ for each signal. Even if the value is randomly generated for each element, the performance of this method is not so good. However, there is room for improvement if τ can be a sequence rather than a fixed value for each element. It is certain that the differences between randomly generated sequences are much larger than randomly generated single values. If we multiply the received signals with these sequences, both the envelopes and the waveforms will be changed. In this view, the effect of sequences is better than values. In a practical implementation, a pseudo-random sequence τ , which has the same length as the sampled signal and contains eight-level weights in a range from $[-1, 1]$, will be generated for each element separately and multiplied with the sampled signals.

$$\mathbf{e}_i[n] = \tau_i[n] \circ \mathbf{f}_i[n], \quad i \in \{1, 2, \dots, N\}, \quad (4.2)$$

Then each sample has its corresponding gain, which means that samples are amplified differently. Furthermore, the pseudo-random sequence is generated for each

element independently, so the differences between elements should be much larger.

Except for these two previously used methods, a new method has been proposed that can differentiate between the received signals by controlling the sampling process, what we call "Random Subsampling". The term "Random Subsampling" does not mean that a lower sampling frequency will be used to sample the received signals. On the contrary, the received signals should be over-sampled firstly, where the sampling frequency will be 4 times larger than the initial setting in our case. The scheme of the method is shown in Figure 4.4.

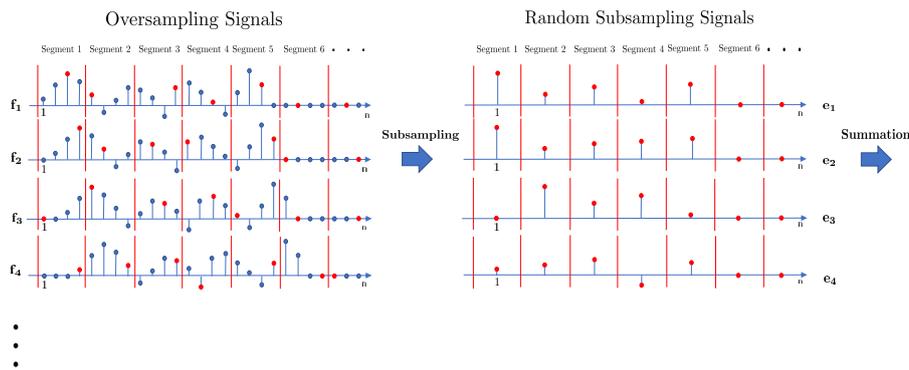


FIGURE 4.4: Random Subsampling Diagram

From the Figure 4.4, we understand that the sampled signals will be sub-sampled after oversampling. In order to display the principle of this new method clearly, we didn't use the practical pulse shapes of signals in the diagram. In reality, the magnitudes of samples are not varying so rapidly as the signals in the scheme due to the oversampling. The procedure of the method is that the sampled signal will be divided into several segments of four samples firstly. Then one sample will be sampled from each segment randomly and inserted into the new signal in sequence. Different sub-sampling formats will be generated and applied to different elements separately. So the final sub-sampled signals should be hardly identical or similar to each other in our assumption.

Another preprocessing method is derived from Architecture 2 which randomly divides 64 elements into 8 physical groups. However, from the view of signals, there is more freedom to manipulate the grouping. The idea we will use is to randomly group signals sample by sample (RandgrpSbyS). The principle is that all connections between elements and output channels in Architecture 2 are virtual or temporary connections, which will change temporally. For each time slot, the samples from 64 elements are grouped with the same scheme as Architecture 2. Then, Architecture 2 would be totally randomized in the temporal dimension. Compared to the original architecture 2, the freedom of signals in the spatial dimension has been exploited. The benefit is that the problems introduced by randomization can be alleviated with a large amount of temporal samples. Meanwhile, it can be combined with other preprocessing methods because it owes the property of grouping and preprocessing.

4.4 Performance Analysis

Due to the change of system architecture and differentiation methods, the performance of the system should be analyzed in order to find out the best combination of them. The reconstructed images by Regularized Least Squares and Matched Filtering are shown in Appendix. We only focus on the environment that all point objects are off-axis with the noisy background. The SNR is still 20dB as before.

The setting of parameters has been changed a lot in the following simulation. The center frequency rises from 3MHz to 18.5MHz, then the sampling frequency should be also increased above the Nyquist Sampling Frequency, which is 55.5MHz. Then the image region should be closer to the transducer because the high-frequency waves cannot transmit far away. Meanwhile, the pixel size should be smaller in order to match the shorter wavelength of ultrasound waves, which is 0.06mm in the simulation. Furthermore, 15 point objects will be put into the image region. They are distributed in a 5 by 3 rectangular form, 5 rows with the same depths in the lateral dimension and 3 columns with the same ranges in the axial dimension. The location of point objects is shown in Figure 4.5

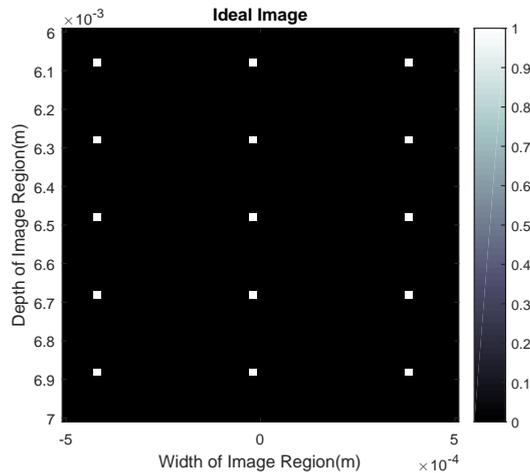


FIGURE 4.5: True Location of the Point Objects

During the analysis, we also take the raw data case into account. Our compressed data will be $\frac{1}{8}$ of the raw data from the entire array in one round of compression. Then, we will also examine the reconstruction results with raw data from 8 elements (element 1, 9, 17, 25, 33, 41, 49, 57), 16 elements (element 1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, 61) and 64 elements in the array to be the comparisons of our compression imaging system designs because their data sizes are equivalent to the 1x, 2x, and 8x of our compressed signals. At least, our system should let us obtain better results than raw data from 8 sensors. Otherwise, the compression designs are excessive.

Firstly, we prefer to evaluate the system from an objective view. Instead of evaluating the quality of the measurement matrix \mathbf{A} by condition numbers, we choose to examine the matrix \mathbf{A} by checking its cross-correlation between columns. Each column in the matrix \mathbf{A} represents a pixel in the image region. We can calculate the auto-correlation of \mathbf{A} by

$$\mathbf{R} = \mathbf{A}^T \mathbf{A} \quad (4.3)$$

where matrix \mathbf{A} is normalized column by column and \mathbf{R} is the auto-correlation matrix of \mathbf{A} . All elements in the diagonal of \mathbf{R} are autocorrelations of columns in \mathbf{A} and

the elements in the other places are cross-correlations between columns in **A**. In the best case, columns in the matrix **A** (pixels) should be orthogonal to each other, so all cross-correlation elements should be zero. It is true that we cannot reach the ideal case in reality that all cross-correlation elements are zero. However, we can deduce that they should approximate zero on average. Only in this way, the ultrasound signal can be well reconstructed. In the following, we will use tables and histograms to illustrate the distribution.

Compression Method	Architecture 1	Architecture 2	Architecture 3
Amplitude Weight	0.0014	0.0013	0.0013
Random Subsampling	0.0071	0.0058	0.0056
Sample Shift	0.0020	0.0019	0.0021
Naive Compression	0.0081	0.0075	0.0070
Raw data from 8 elements	0.0011		
Raw data from 16 elements	0.0015		
Raw data from 64 elements	0.0008		
RandgrpSbyS	0.0072		
RandgrpSbyS+AW	0.0012		
RandgrpSbyS+SS	0.0023		
RandgrpSbyS+AW+SS	0.0013		

TABLE 4.1: Average of Cross-correlation between Pixels (Linear scale).
AW=Amplitude Weight, SS=Sample Shift, RandgrpSbyS=Random
Grouping Sample by Sample, '+'=Combination of methods

From table 4.1, the averages of cross-correlations are close to zero for all kinds of designs and raw data cases. Usually the averages are smaller when the conditions of matrix **A** are better. Therefore, we can conclude that pixels have orthogonal representations on average in all cases.

Differentiation Method	Architecture 1	Architecture 2	Architecture 3
Amplitude Weight	0.0360	0.0349	0.0349
Random Sub-sampling	0.0602	0.0429	0.0494
Sample Shift	0.0428	0.0413	0.0415
Naive Compression	0.0660	0.0479	0.0573
Raw data from 8 elements	0.0406		
Raw data from 16 elements	0.0307		
Raw data from 64 elements	0.0230		
RandgrpSbyS	0.0420		
RandgrpSbyS+AW	0.0347		
RandgrpSbyS+SS	0.0343		
RandgrpSbyS+AW+SS	0.0337		

TABLE 4.2: Standard Deviation of Cross-correlation between Pixels (Linear scale). AW=Amplitude Weight, SS=Sample Shift, RandgrpSbyS=Random Grouping Sample by Sample, '+'=Combination of methods

The standard deviation is being used to quantify the amount of variation of a set of values. Compared to the average, the standard deviation is commonly used to measure confidence in statistical conclusions. A small standard deviation indicates that values tend to be close to the average of the set. From table 4.2, we can find that

the Amplitude Weight and Random grouping sample by sample plus other preprocessing methods have the standard deviations than raw data from 8 elements. It is a more accurate method than the average. It reflects the values of cross-correlation are more concentrated around average or close to zero, so the conditions of the matrix \mathbf{A} are better with these methods. As aforementioned, the matched filtering results highly rely on the incoherence between pixels. Then, we can obtain the same conclusions from the reconstructed images of matched filtering in Figures A.7 to A.12 as from above tables.

The cross-correlation of columns in matrix \mathbf{A} analyzes the system by evaluating the quality of \mathbf{A} . We should note that the Matched Filter is unreliable when the matrix \mathbf{A} is ill-posed. Then, the Regularized Least Squares is a more suitable algorithm we can count on because its performance does not totally rely on the condition of \mathbf{A} . In Chapter 3, we have already examined both LSQR and Tikhonov Regularization methods, when the regularization parameters have been appropriately chosen, their performances are close. We apply LSQR in this chapter uniformly in consideration of computational complexity. The selection of regularization parameters (or iteration numbers) has a huge impact on the final result. Even though there are many scientific ways to calculate them, but we can also determine them subjectively or objectively. The subjective way is to observe the reconstructions. As for the objective, we decide to use the *mean squared error (MSE)* to measure the quality of the reconstruction results of Regularized Least Squares.

$$MSE = E[(\hat{\mathbf{x}} - \mathbf{x})^2] \quad (4.4)$$

The initial image \mathbf{x} is a matrix where only the corresponding elements of the point objects are 1 and other elements are zero. The $\hat{\mathbf{x}}$ is the normalized matrix of the reconstructed images, which are shown in the Appendix. The simulation is done in a noisy (SNR=20 dB) environment where all point objects are off-axis. This is the worst case we will face in Point-Object Imaging. But, we have to note that the results with the lowest MSEs are not always the best because the quality of images is finally determined by our subjective views.

Differentiation Method	Architecture 1	Architecture 2	Architecture 3
Amplitude Weight	0.0142	0.0159	0.0180
Random Sub-sampling	0.0261	0.0229	0.0192
Sample Shift	0.0250	0.0190	0.0209
Naive Compression	0.0341	0.0224	0.0235
Raw data from 8 elements	0.0308		
Raw data from 16 elements	0.0151		
Raw data from 64 elements	0.0117		
RandgrpSbyS	0.0146		
RandgrpSbyS+AW	0.0131		
RandgrpSbyS+SS	0.0164		
RandgrpSbyS+AW+SS	0.0140		

TABLE 4.3: Mean Squared Error between \mathbf{x} and estimated $\hat{\mathbf{x}}$ (Regularized Least Squares/Linear scale). AW=Amplitude, Weight SS=Sample Shift, RandgrpSbyS=Random Grouping Sample by Sample, '+ '=Combination of methods

In table 4.3, a lower MSE means that the reconstructed image is more approximate to the ideal image where the points should be more concentrated and less

background ‘noise’. We can find that the Amplitude Weight and the Random Grouping Sample by Sample (or plus other methods) can achieve lower MSEs than other differentiation methods. According to the Figures A.1, A.4 and A.6, it is true that the images reconstructed with them are better than the reconstruction with the raw data from 8 sensors at least, but it is hard to judge whether they are better than the reconstruction with the raw data from 16 sensors. From Figures A.1 to A.5, we can find that the effect of the architecture is not significant when preprocessing has been applied except the Random Sampling methods where the quality of the images corresponds to their MSEs. In a subjective view, the Naive Compression and the Random Sampling methods are almost the same. The reasons will be discussed in the following section.

In order to illustrate the Matching Filtering results more intuitively, we also measure their MSEs.

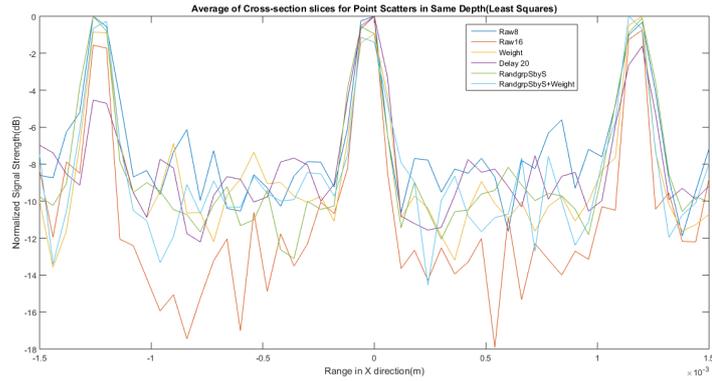
Differentiation Method	Architecture 1	Architecture 2	Architecture 3
Amplitude Weight	0.0145	0.0171	0.0148
Random Sub-sampling	0.0545	0.0292	0.0295
Sample Shift	0.0285	0.0190	0.0182
Naive Compression	0.0364	0.0261	0.0172
Raw data from 8 elements	0.0202		
Raw data from 16 elements	0.0139		
Raw data from 64 elements	0.0089		
RandgrpSbyS	0.0258		
RandgrpSbyS+AW	0.0127		
RandgrpSbyS+SS	0.0120		
RandgrpSbyS+AW+SS	0.0155		

TABLE 4.4: Mean Squared Error between x and estimated \hat{x} (Matched Filtering/Linear scale). AW=Amplitude, Weight SS=Sample Shift, RandgrpSbyS=Random Grouping Sample by Sample, ‘+’=Combination of methods

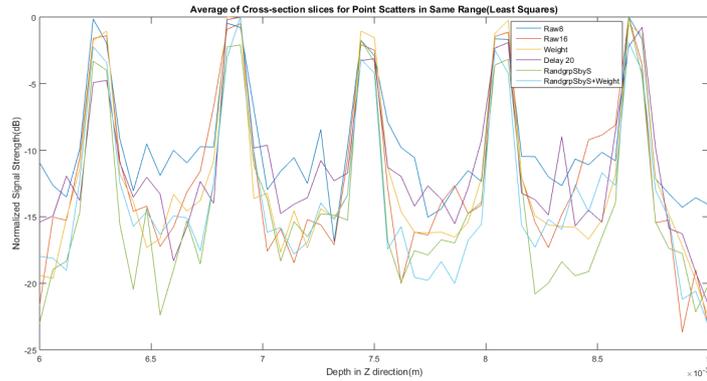
Previously, we have already evaluated the Matched Filtering results indirectly with the cross-correlations between columns in **A**. The relative differences of results in table 4.4 are not totally the same as in table 4.2. But we can still get the same conclusion that the Amplitude Weight and Random Grouping Sample by Sample plus other preprocessing methods are the best among all methods. But, we cannot totally trust the MSE. For instance, the combination of Naive Compression and Architecture 3 provide us with an incorrect result (Figure A.11(c)), but it has a pretty low MSE.

Combined with the figures in Appendix and above evaluation results, we plan to have a further study of the corresponding designs of better reconstructions. The chosen designs are the Amplitude Weight plus Architecture 1, the Sample Shift plus Architecture 2, the Random Grouping Sample by Sample (or plus the Amplitude Weight). In order to understand 2-D images intuitively, we transform the images into 2 slices in axial dimension and lateral dimension separately. Since all point objects are distributed in rows (lines in axial dimension) and columns (lines in lateral dimension), we plan to cut them out with their adjacent rows (4 rows) and columns (8 columns) because the PSFs are shorter in lateral dimension than axial dimension. Then, we take the average of them to represent one dimension of images in a single slice.

In Figure 4.6, the peaks of main-lobes in curves are the locations of point objects, the side-lobes are ‘noise’ on the background of images. For a single curve, the larger



(A) Cross-section slices of rows of point scatters



(B) Cross-section slices of rows of point scatters

FIGURE 4.6: Axial and Lateral image profile of the reconstructed images with Regularized Least Squares

relative difference between the peaks of the main-lobes and the side-lobes indicates that point objects are much brighter than the background area. If we compare all curves together, the disparities between each compression design is obvious. According to the Figure 4.6, the Amplitude Weight and Random Grouping Sample by Sample perform better than others, especially in axial dimension.

The same conclusion can be obtained from the results with smaller pixel sizes in a smaller image region. The reconstructions are shown in Appendix B (Figure B.1-B.12)

4.5 Discussion of the Advantages and Disadvantages of pre-processing methods and three architectures

4.5.1 The Impact of Architectures

If we do not apply any preprocessing methods, the compression achieved by pure summation of the received signals (Naive Compression) is impossible to reconstruct images successfully. This is what we have learned from Chapter 2. But the new design of 8 sub-arrays in the architecture totally changes this consequence. And each type of grouping scheme provides us with a unique result (Figures A.5 and A.11). Let us analyze the reasons in sequence.

In Architecture 1, the received signals from 8 adjacent elements, which are in the same sub-array, are summed up. It is the Naive Compression of a sub-array. For a single sub-array, the reconstruction is the image contains curves at the same depth of the points of interest that have the same distances to the sub-array as the point objects. In Chapter 2, we have already explained that these curves are formed by a collection of pixels, which are symmetric relative to the array, and the surrounding areas of them. For each pair of symmetric pixels and adjacent pixels, the Naive Compression is hard to distinguish them. The difference is that the final result is the combination of 8 sub-arrays rather than the duplicates from a single larger array. But the matched filtering image of Architecture 1 in Figure A.11(A) may be confusing. So, Figure A.11(A) is the superposition of 3 columns point objects, the image is formed by five 'curves' because the PSFs are too wide.

As for the Architecture 2, for a single sub-array, the received signals in it are similar to delaying signals in the corresponding sub-array in Architecture 1. The time-of-arrival differences in the sub-array are fixed initially. After, the application of the random sample shift, the time-of-arrivals of received signals of elements in the sub-array are randomized. Compared to the Architecture 1, the randomized grouping also has a similar impact on the same sub-array since the time-of-arrivals of signals are also randomized. However, this assumption is based on ignoring the influence of pulse shapes. If we consider that the pulse shapes of the received signals are not the same, then the sample shift which has already randomized the time-of-arrivals of the entire array is different from the Architecture 2 which is just a similar model in the sub-array. Therefore, the individually application of Architecture 2 is able to give us an acceptable result (Figure A.5 (B) and Figure A.11 (B)).

The reconstruction result of Architecture 3 is 'remarkable'. For each sub-array of this architecture, the pixels have the same Naive Compression results if they are symmetric relative to the sub-array. A specific point object has 8 symmetric pixels from the views of 8 sub-arrays, where these pixels also have their own symmetric pixels for 8 sub-arrays. This is the truth in all kind of architectures. But in Architecture 1, these pixels are concentrated in the surrounding area, which cause huge PSFs. In Architecture 2, the randomized pattern suppresses the impact, where these symmetric pixels are distributed randomly. The superposition of them cannot influence the images significantly. In Architecture 3, the sub-arrays are distributed in the interleaving style, where the symmetric pixels are also distributed spatially in sequence. And their locations are highly overlapping due to the regular distributions of them. As a result, point objects have many duplicates in **A**, where the matched filtering results in Figure A.11(A) have shown that columns of point objects have many duplicates in the axial dimension.

4.5.2 The Impact of Preprocessing Methods

According to the above evaluations and analysis, we found that Random Sub-sampling is the worst performer among all preprocessing methods. It's really interesting to research the potential problems of this method. We have an assumption that there is a problem of this preprocessing methods. Firstly, we know that the similarities between adjacent pixels are high from all results we have obtained which is the same in all experiments. The procedure of the Random Sub-sampling was realized by dividing the oversampled signals into segments and sampling one sample from each segment. The intention is to differentiate the received signals since this process is done randomly. However, we have to sum them up. We can imagine that

one sample will represent a small interval initially. Here we will represent this interval with four samples if we oversample the original signal. Since our sampling frequency is already above Nyquist Sampling Frequency, we can conclude that this interval should be small enough. Hence, all samples in the random sub-sampled signals of the oversampling received signals should be also in the corresponding intervals that was represented by the mid-point of this interval. This will be a problem if we sum them up. The summation result should be similar to the pure summation case, even if all samples in sub-sampled signals are located at the edges of the intervals. So, in the current stage, the Random Subsampling is not a good choice for the preprocessing method.

Before we discuss the Random Grouping Sample by Sample method, we have to consider the full data case carefully. In the full data case, we only store the signals from all elements in sequence. For any pair of two pixels, they are diverse because the time-of-arrivals and waveforms between elements are different. This kind of differences are more distinct when more measurements have been stored. In our design, the number of different sets of measurements is 64 (64 elements in the array). Previously, the three architectures preserve both properties partially. The sample shift and amplitude weight methods enhance the impact of temporal information and magnitude information separately. Hence, they have improved the results differently.

Then, let us consider the Random Grouping Sample by Sample. It can be regarded as an extension of Architecture 2 in the time domain. But, the effects of them are totally different. Due to the random grouping, every element is the potential source for one output channel. The temporal information caused by relative various time-of-arrivals has been totally eliminated. The summations in all sub-arrays are in the same time interval with high probability. For pixels in the same depth, the time interval of outputs are highly similar, even if we have randomly grouped the received signals sample by sample and summed them up in 8 groups. But in the magnitude dimension, for a single sub-array (or output), the waveforms of signals are totally randomized, which is the same as the application of amplitude weight. The Architecture 2 and Random Grouping Sample by Sample can be treated as two types of grouping, which exploit the temporal and magnitude property individually. We can name them pseudo-processing of the raw signals. Hence, we can find some similarities between their reconstruction result.

4.6 Summary

We have introduced and examined new architectures and preprocessing methods in this chapter. The simulation results show that the new designs of compression system are effective when appropriate preprocessing methods have been chosen. The improvement of the Amplitude Weight methods is significant. And we find the random grouping sample by sample is a good enhancement of the grouping scheme design. Among all preprocessing methods, the performance of Random Subsampling is much lower than our initial expectation. So we believe the combination of these two methods is the most stable method to achieve a high quality ultrasound reconstruction. We can compress the raw data to 12.5% of its data size, where the outcome is better than multiplexing the received data from 8 elements and almost as well as the received data from 16 sensors.

Chapter 5

Conclusion

5.1 Summary

It was determined from the background research that existing studies of compressive sampling are highly reliant on searching the sparse expansion of the objects. However, the hardware implementations are less concerned in these studies. The Semi-controllable Compression Scheme was started with the intent to develop a compression scheme containing the features of compressive sampling theory to reduce the output signals of an ultrasound transducer. With this objective, a compression scheme based on the pseudo-random linear combination of the received signals from transducer elements has been designed for a pulse-echo ultrasound imaging. The compression degree can be easily manipulated by determining the summation format. The preprocessing unit before linear summation increases the probability that the pixels have their unique representations, which is able to promote the quality of reconstructions. In Chapter 3, in the design of Entire Array Compression, the uniqueness of a specific pixel in temporal dimension and magnitude dimension is highly compressed. The application of preprocessing methods can improve the reconstructions to some degree. But the reconstruction is still insufficient to meet our expectation. As a consequence, alternative compression architectures and new preprocessing methods have been proposed in Chapter 4. According to the simulation results, we can conclude that our compression-imaging system is able to reconstruct the ultrasound signals well when the appropriate architectures and preprocessing methods have been chosen. Both reconstruction and evaluation results demonstrate that the improved 8-level Amplitude Weight method or its combination with the Random Grouping Sample by Sample are the most suitable preprocessing methods in current stage. Acceptable ultrasound images can be reconstructed with only $\frac{1}{8}$ of the original data and output channels. Compared to the conventional compressive sampling, our compression scheme is much simpler and flexible. Whether or not the ultrasound signals are potentially sparse in any domain, our compression scheme is applicable. Furthermore, our original intention is to apply the easiest method during the design, such as the linear summation, the reconstruction methods etc. We believe there is room for promotion to achieve a much better result and make our system more stable as described below.

5.2 Future Work

There are still a lot of works that can be done in this following research, we can divide them into 3 parts: At first, all values of variables or formats in our preprocessing methods are randomly generated with some restrictions that is why we call our system a semi-controllable compression scheme. We use the same idea as the conventional CS that the randomized mechanism can give us a good result with high

probability. But we did not apply any kind of optimization during the reconstruction. Hence, the instability of random numbers is a potential trouble in any practical implementation. A possible solution is to transform our problem into an optimization problem. The object of optimization is the set of variables in our preprocessing methods, which is directly related to the differentiation of the pixels. We hope a high-quality reconstruction can be achieved by a stable system.

Secondly, the combination of differentiation methods is another choice for the improvement of our system. However, at the current stage, the impact of a combination such as amplitude weight and sample shift is not significant. Hence, a more suitable strategy of combination can be studied.

Finally, the prototype of our compression scheme is based on linear summation of preprocessed signals. It is an easy and controllable method to compress data. However, during the discussion, we have explained that the linear summation may also introduce some problems. A possible idea is that a non-linear combination of signals may be a better choice if our goal is to differentiate the pixels with high similarity. However, an appropriate mathematical model and acceptable price of implementation may be the hardest part of this idea.

Appendix A

Reconstruction Results with New Architectures and Updated Preprocessing Methods(Pixel Size: 0.06mm)

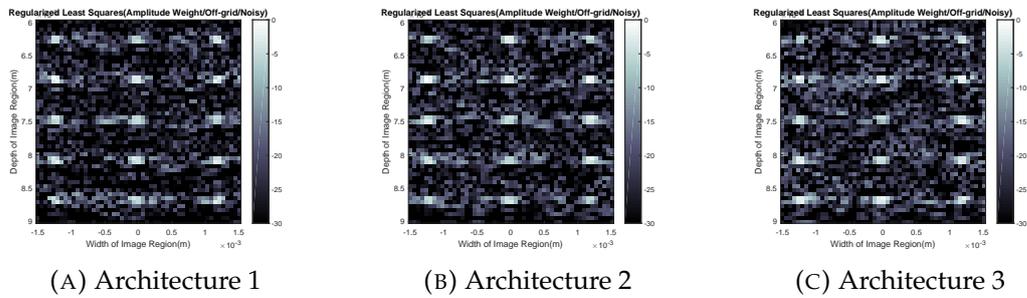


FIGURE A.1: Regularized Least Squares(Amplitude Weight/Off-axis Point Objects/Noisy/dB mode)

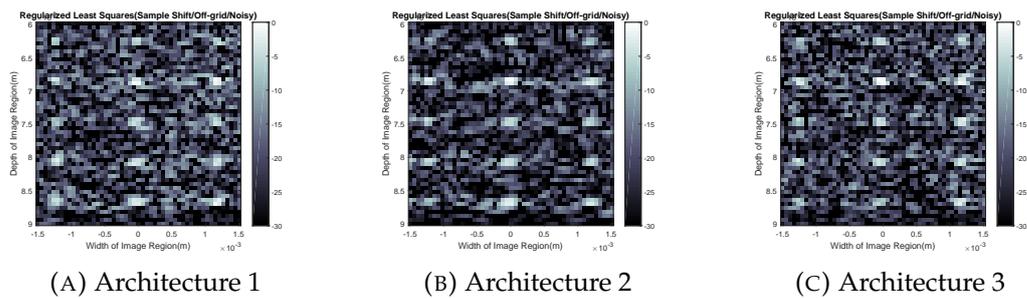


FIGURE A.2: Regularized Least Squares(Sample Shift up to 20 samples/Off-axis Point Objects/Noisy/dB mode)

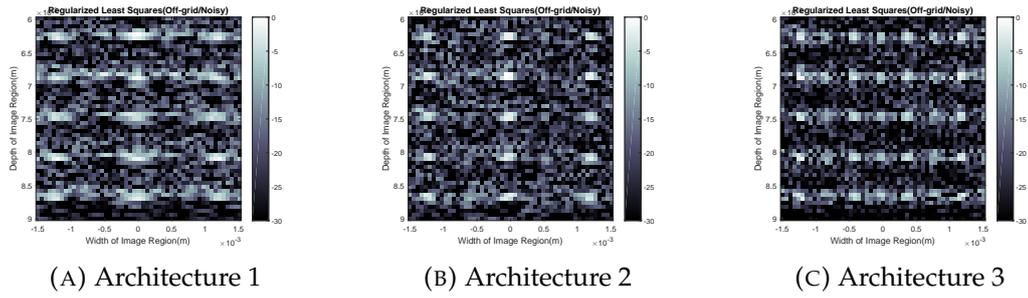


FIGURE A.3: Regularized Least Squares(Random Subsampling/Off-axis Point Objects/Noisy/dB mode)

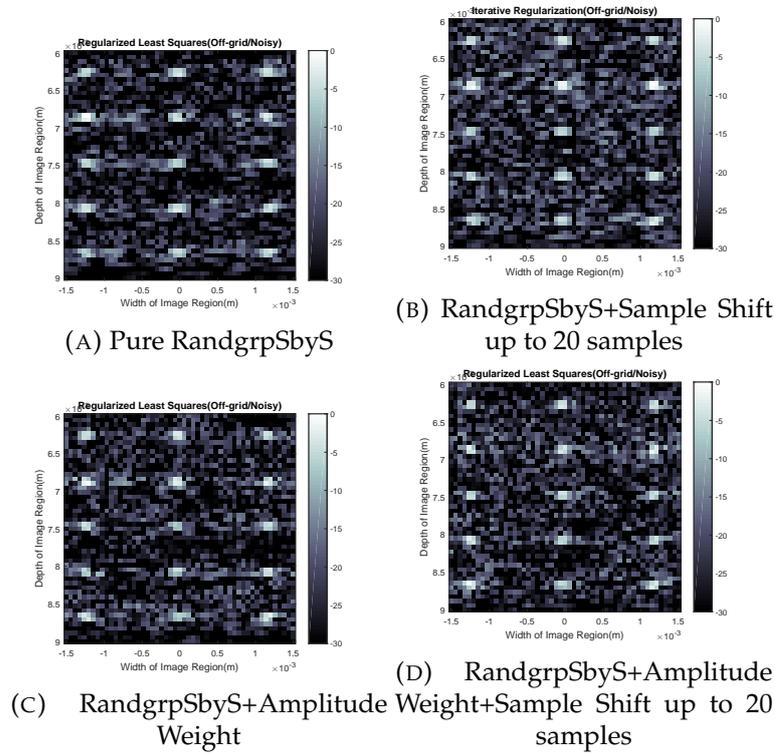


FIGURE A.4: Regularized Least Squares(Random Grouping Sample by Sample+Amplitude Weight+ Sample Shift/Offgrid/Noisy/dB mode)

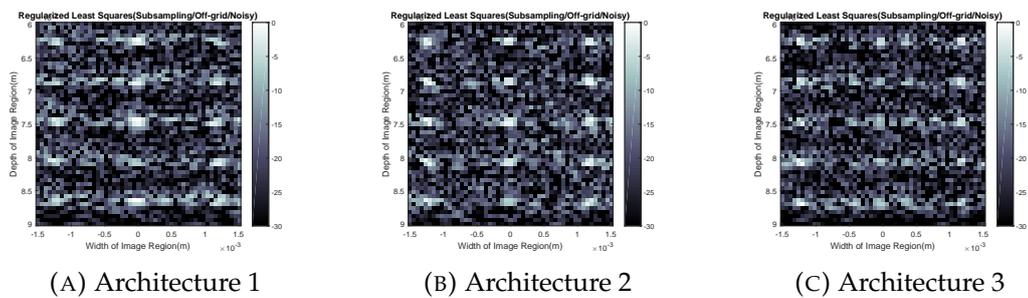


FIGURE A.5: Regularized Least Squares(Naive Compression/Off-axis Point Objects/Noisy/dB mode)

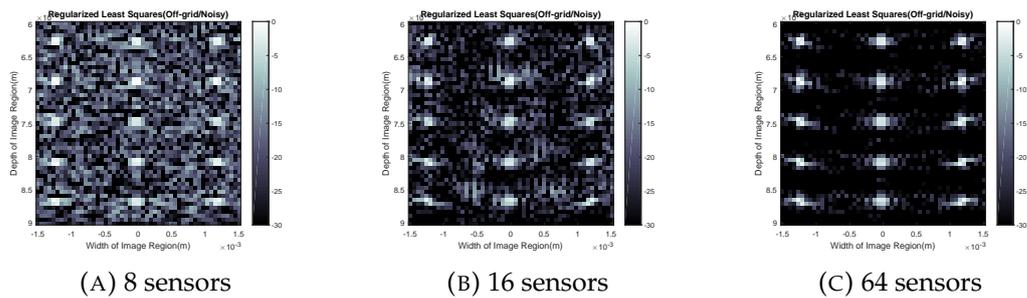


FIGURE A.6: Regularized Least Squares(Off-axis Point Objects/Noisy/dB mode). Raw data from sensors without compression.

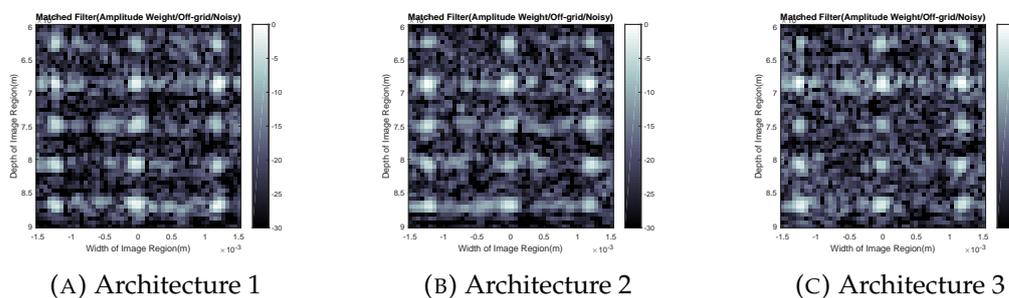


FIGURE A.7: Matched Filtering(Amplitude Weight/Off-axis Point Objects/Noisy/dB mode)

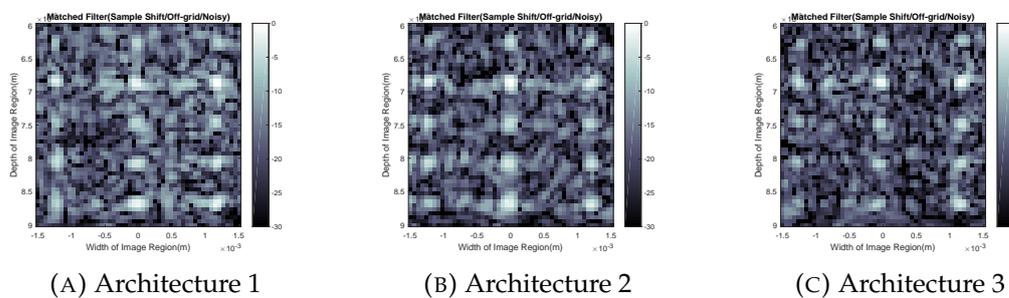


FIGURE A.8: Matched Filtering(Sample Shift up to 20 samples/Off-axis Point Objects/Noisy/dB mode)

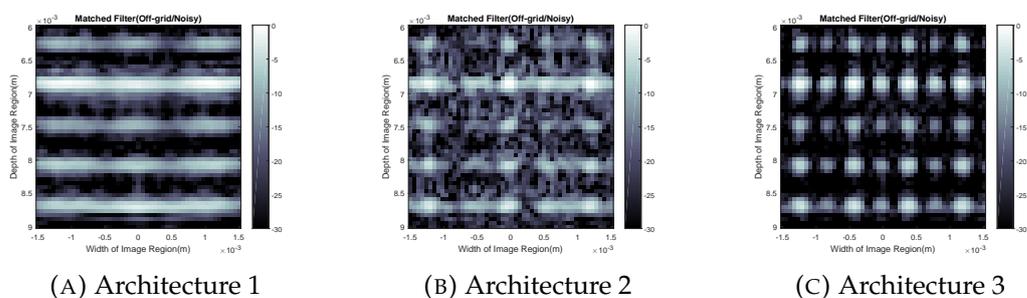


FIGURE A.9: Matched Filtering(Random Subsampling/Off-axis Point Objects/Noisy/dB mode)

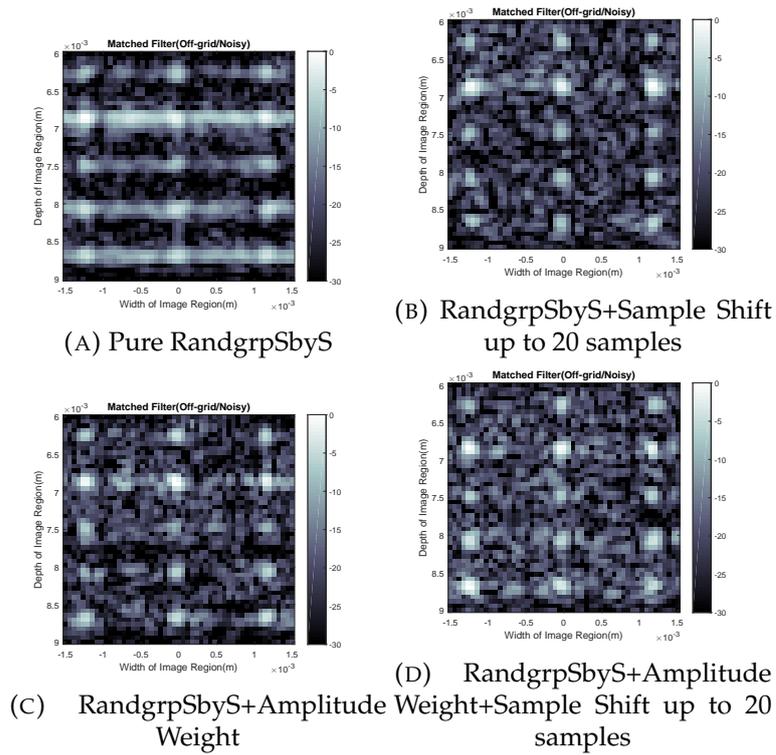


FIGURE A.10: Matched Filtering(Random Grouping Sample by Sample+Amplitude Weight+Sample Shift/Off-axis Point Objects/Noisy/dB mode)

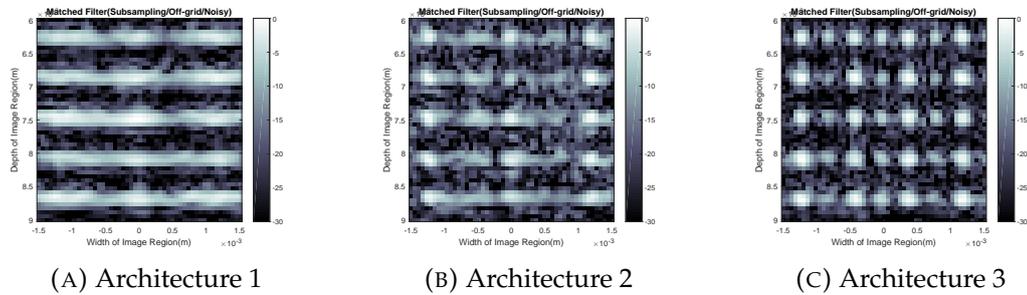


FIGURE A.11: Matched Filtering(Naive Compression/Off-axis Point Objects/Noisy/dB mode)

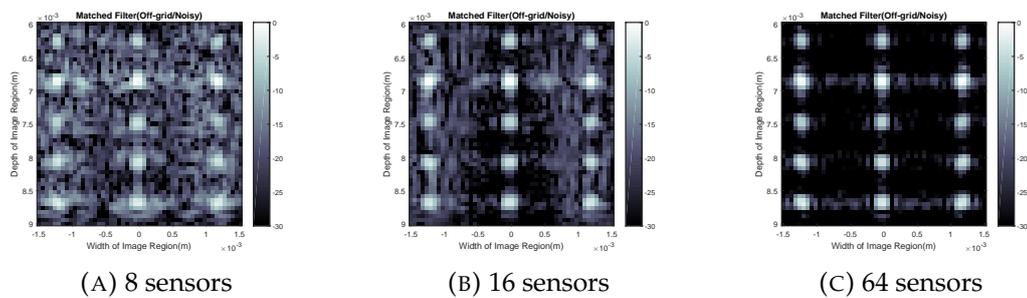


FIGURE A.12: Matched Filtering(Off-axis Point Objects/Noisy/dB mode). Raw data from sensors without compression.

Appendix B

Reconstruction Results with New Architectures and Updated Preprocessing Methods(Pixel Size: 0.02mm)

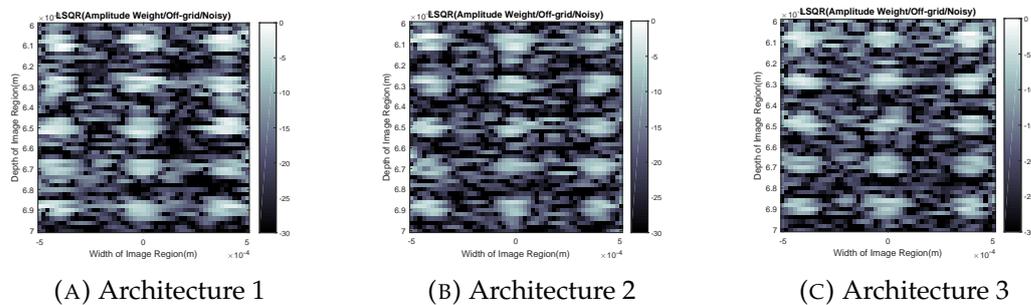


FIGURE B.1: Regularized Least Squares(Amplitude Weight/Off-axis Point Objects/Noisy/dB mode)

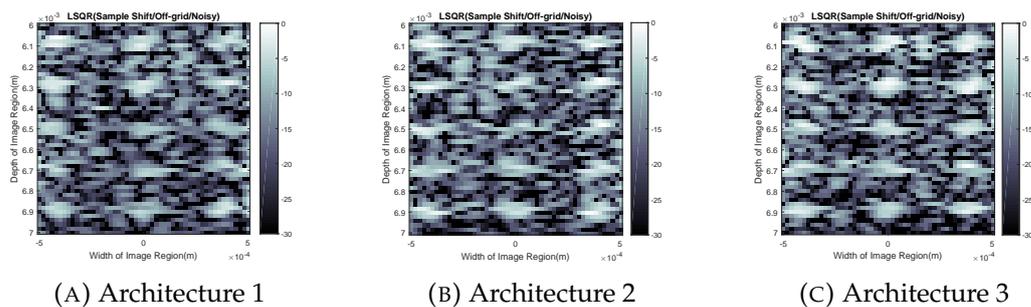


FIGURE B.2: Regularized Least Squares(Sample Shift up to 20 samples/Off-axis Point Objects/Noisy/dB mode)

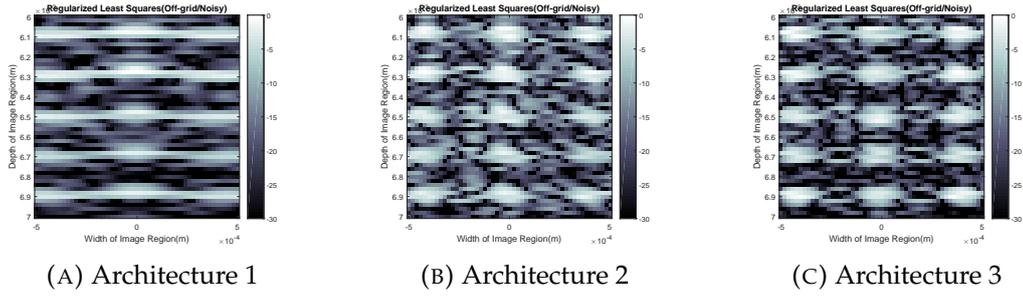


FIGURE B.3: Regularized Least Squares(Random Subsampling/Off-axis Point Objects/Noisy/dB mode)

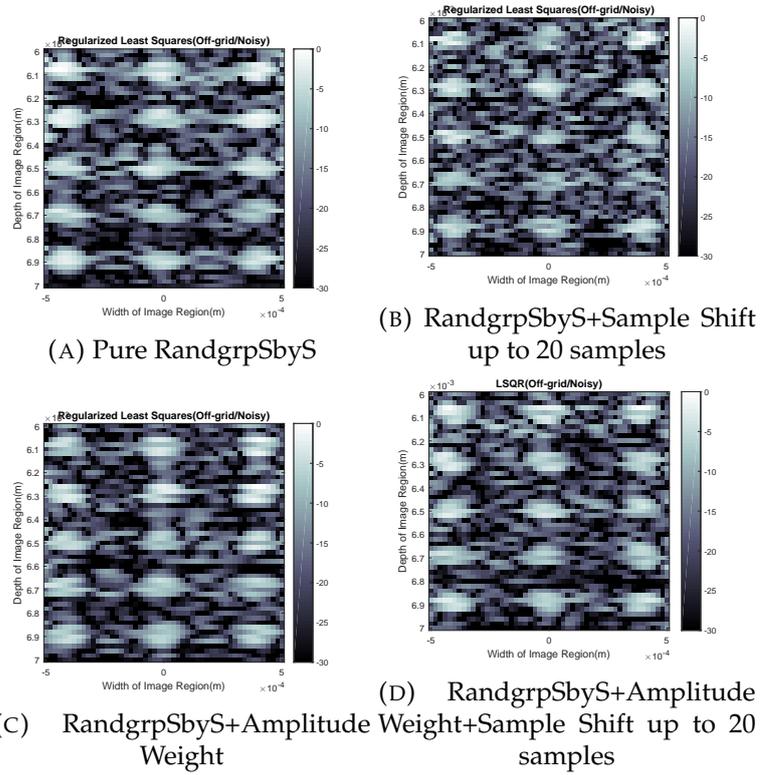


FIGURE B.4: Regularized Least Squares(Random Grouping Sample by Sample+Amplitude Weight+ Sample Shift/Offgrid/Noisy/dB mode)

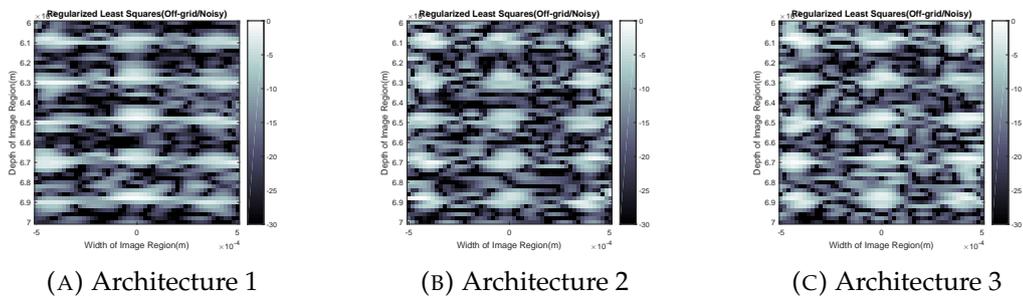


FIGURE B.5: Regularized Least Squares(Naive Compression/Off-axis Point Objects/Noisy/dB mode)

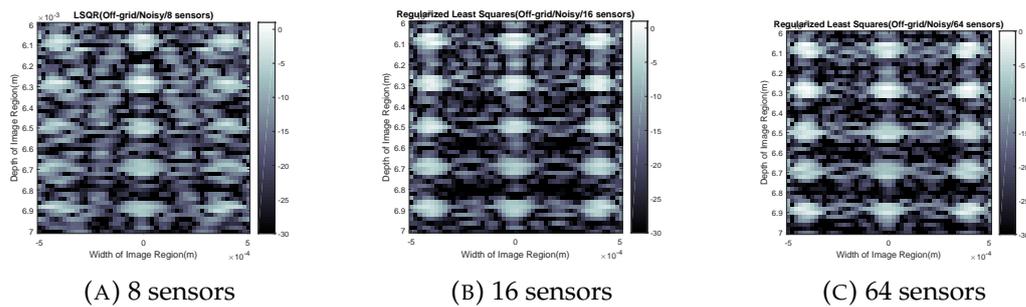


FIGURE B.6: Regularized Least Squares(Off-axis Point Objects/Noisy/dB mode). Raw data from sensors without compression.

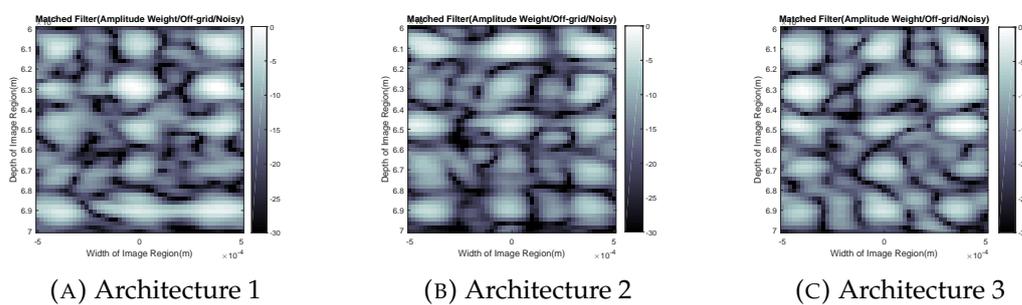


FIGURE B.7: Matched Filtering(Amplitude Weight/Off-axis Point Objects/Noisy/dB mode)

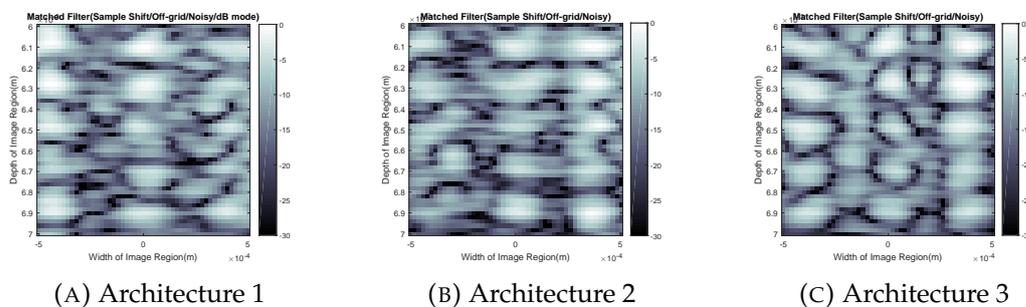


FIGURE B.8: Matched Filtering(Sample Shift up to 20 samples/Off-axis Point Objects/Noisy/dB mode)

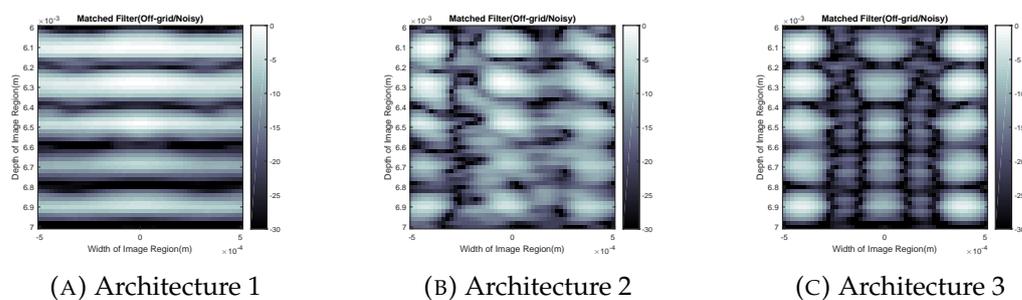


FIGURE B.9: Matched Filtering(Random Subsampling/Off-axis Point Objects/Noisy/dB mode)

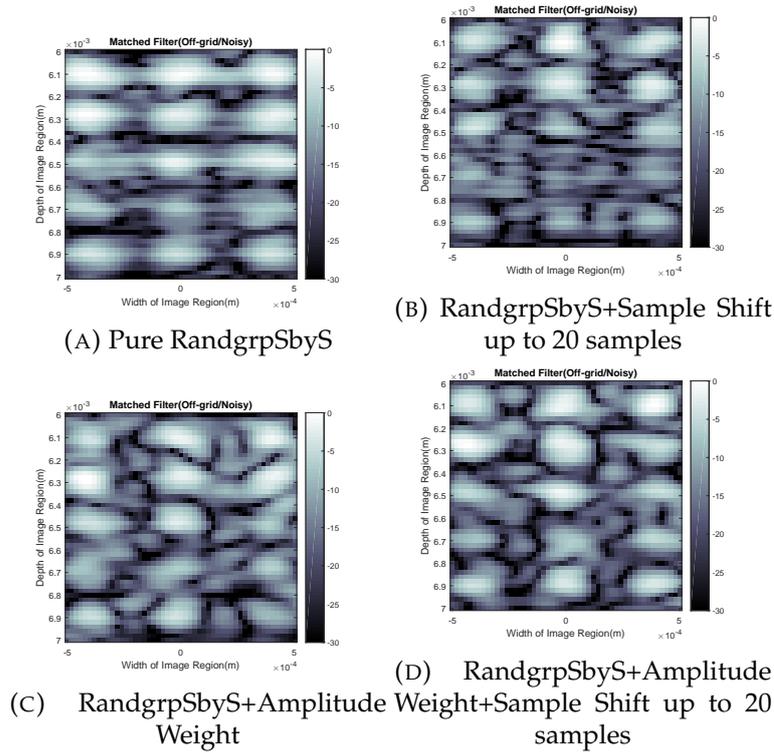


FIGURE B.10: Matched Filtering(Random Grouping Sample by Sample+Amplitude Weight+Sample Shift/Off-axis Point Objects/Noisy/dB mode)

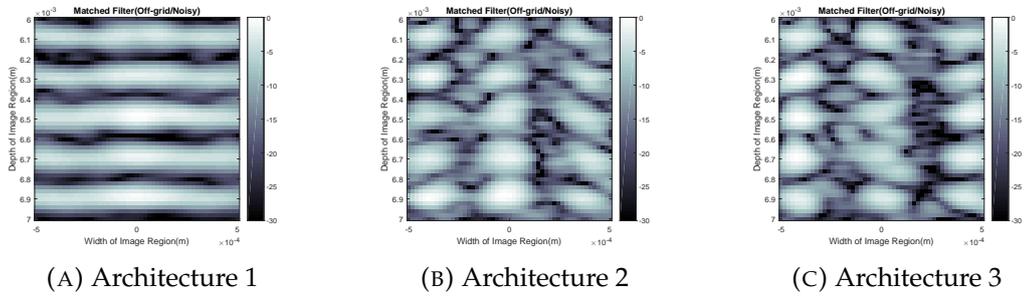


FIGURE B.11: Matched Filtering(Naive Compression/Off-axis Point Objects/Noisy/dB mode)

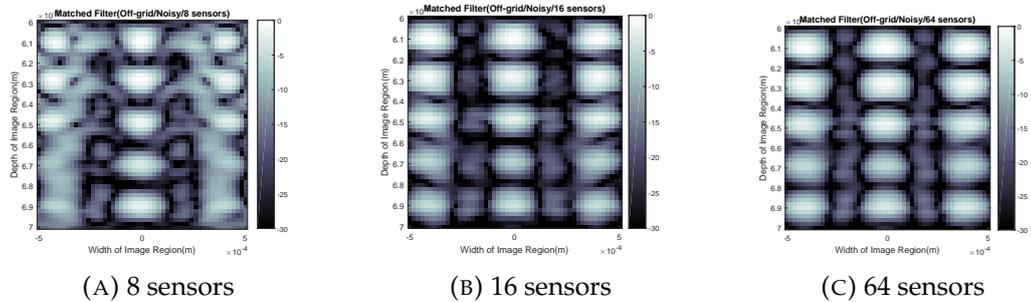


FIGURE B.12: Matched Filtering(Off-axis Point Objects/Noisy/dB mode). Raw data from sensors without compression.

Bibliography

- [1] H. Liebgott, A. Basarab, D. Kouame, O. Bernard and D.Friboulet, "Compressive sensing in medical ultrasound", *2012 IEEE International Ultrasonics Symposium*, 2012.
- [2] N. Wagner, Y. Eldar and Z. Friedman, "Compressed Beamforming in Ultrasound Imaging", *IEEE Transactions on Signal Processing*, vol. 60, no. 9, pp. 4643-4657, 2012.
- [3] T. Szabo, "Diagnostic ultrasound imaging", *Academic Press*, Kidlington, Oxford, UK, 2014.
- [4] A. Liutkus, D. Martina, S. Popoff, G. Chardon, O. Katz, G. Lerosey, S. Gigan, L. Daudet and I. Carron, "Imaging With Nature: Compressive Imaging Using a Multiply Scattering Medium", *Scientific Reports*, vol. 4, no. 1, 2014.
- [5] F. Viola, M. Ellis and W. Walker, "Time-Domain Optimized Near-Field Estimator for Ultrasound Imaging: Initial Development and Results", *IEEE Transactions on Medical Imaging*, vol. 27, no. 1, pp. 99-110, 2008.
- [6] C. Shannon, "Communication in the Presence of Noise", *Proceedings of the IRE*, vol. 37, no. 1, pp. 10-21, 1949.
- [7] E. Candes and M. Wakin, "An Introduction To Compressive Sampling", *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21-30, 2008.
- [8] E. Candes, J. Romberg and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information", *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489-509, 2006.
- [9] E. Candès, "Compressive Sampling", *Proceedings of the International Congress of Mathematician*, Madrid, Spain, 2006.
- [10] D. Woodruff, "Sketching as a Tool for Numerical Linear Algebra", *Foundations and Trends R in Theoretical Computer Science*, vol. 37, no. 1, pp. 10-21, 1949.
- [11] C. Graff and E. Sidky, "Compressive sensing in medical imaging" *Applied Optics*, vol. 54, no. 8, p. C23, 2015.
- [12] P. Mishra, R. Bharath, P. Rajalakshmi and U. Desai, "Compressive sensing ultrasound beamformed imaging in time and frequency domain", *17th International Conference on E-health Networking, Application and Services (HealthCom)*, 2015.
- [13] R. Bethel, B. Shapo and H. Van Trees, "Single snapshot spatial processing: optimized and constrained", *Sensor Array and Multichannel Signal Processing Workshop Proceedings*, 2002.
- [14] R. Llavorello, F. Kamalabadi and W. O'Brien, "A regularized inverse approach to ultrasonic pulse-echo imaging", *IEEE Transactions on Medical Imaging*, vol. 25, no. 6, pp. 712-722, 2006.

- [15] M. Schiffner, T. Jansen and G. Schmitz, "Compressed Sensing for Fast Image Acquisition in Pulse-Echo Ultrasound", *Biomedical Engineering*, vol. 57, pp. 192-195, 2012.
- [16] M. Schiffner and G. Schmitz, "Fast Pulse-Echo Ultrasound Imaging Employing Compressive Sensing", *IEEE International Ultrasonics Symposium*, Orlando, Florida, USA, 2011, p. in press.
- [17] H. Liebgott, R. Prost, and D. Friboulet "Pre-beamformed RF signal reconstruction in medical ultrasound using compressive sensing", *Ultrasonics*, p. accepted, 2012.
- [18] D. Friboulet, H. Liebgott, and R. Prost, "Compressive sensing for raw RF signals reconstruction in ultrasound", *IEEE International Ultrasonics Symposium*, San Diego, California, USA, 2010, pp.367-370.
- [19] C. Quinsac, A. Basarab, and D. Kouamé, "Frequency domain compressive sampling for ultrasound imaging", *Advances in Acoustics and Vibration, Advances in Acoustic Sensing, Imaging, and Signal Processing*, vol. 12, pp. 1-16, 2012.
- [20] D. Friboulet, H. Liebgott, and R. Prost, "Compressed sensing of ultrasound images: sampling of spatial and frequency domains", *IEEE Workshop on Signal Processing Systems*, San Francisco, California, USA, 2010, pp. 231-236.
- [21] J. A. Jensen and N. B. Svendsen, "Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers", *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, 39:262-267, 1992.
- [22] J. A. Jensen, "Field: A program for simulating ultrasound systems", *Med. Biol. Eng. Comp.*, 10th Nordic-Baltic Conference on Biomedical Imaging, Vol. 4, Supplement 1, Part 1:351-353, 1996b.
- [23] D. Jain and R. Kanwal, "The Born approximation for the scattering theory of elastic waves by two-dimensional flaws", *Journal of Applied Physics*, vol. 53, no. 6, pp. 4208-4217, 1982.
- [24] F. Lingvall, "A method of improving overall resolution in ultrasonic array imaging using spatio-temporal deconvolution", *Ultrasonics*, vol. 42, no. 1-9, pp. 961-968, 2004.
- [25] P. R. Stepanishen, "Pulsed transmit/receive response of ultrasonic piezoelectric transducers", *J. Acoust. Soc. Am.*, 69:1815-1827, 1981.
- [26] A.N. Tikhonov and V. Y. Arsenin, "Solutions of Ill-Posed Problems", *Wiley*, New York, 1977.
- [27] P. C. Hansen and D. P. O'Leary "The use of the L-Curve in the Regularization of Discrete Ill-posed Problems", *SIAM J. SCI. COMPUT.*, Vol. 14, No. 6, pp. 148-1503, November 1993
- [28] C. Paige and M. Saunders, "LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares", *ACM Transactions on Mathematical Software*, vol. 8, no. 1, pp. 43-71, 1982.
- [29] G.H. Golub and W.M. Kahan "Calculating the singular values and pseudoinverse of a matrix", *SIAM J. Num. Anal. Ser. B*, 2:205-224, 1965.

-
- [30] Y. Huang and Z. Jia "Some results on the regularization of LSQR for large-scale discrete ill-posed problems", *Science China Mathematics*, vol. 60, no. 4, pp. 701-718, 2016.
- [31] A. Liutkus, D. Martina, S. Popoff, G. Chardon, O. Katz, G. Lerosey, S. Gigan, L. Daudet and I. Carron, "Imaging With Nature: Compressive Imaging Using a Multiply Scattering Medium", *Scientific Reports*, vol. 4, no. 1, 2014.