Efficient Similarity-Driven Emission Angle Selection for Coherent Plane-Wave Compounding

by

Haroon Ali Akbar
BEE, National University of Sciences and Technology, 2016

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

University of Victoria

Supervisory Committee

Efficient Similarity-Driven Emission Angle Selection for Coherent Plane-Wave Compounding

by

Haroon Ali Akbar
BEE, National University of Sciences and Technology, 2016

Supervisory Committee

Dr. Daler Rakhmatov, Department of Electrical and Computer Engineering

Supervisor

Dr. Mihai Sima, Department of Electrical and Computer Engineering

Departmental Member

Abstract

Supervisory Committee

Dr. Daler Rakhmatov, Department of Electrical and Computer Engineering
Supervisor

Dr. Mihai Sima, Department of Electrical and Computer Engineering
Departmental Member

Typical ultrafast plane-wave ultrasound imaging involves: 1) insonifying the medium with several plane-wave pulses emitted at different angles by a linear transducer array, 2) sampling the returning echo signals, after each plane-wave emission, with the same transducer array, 3) beamforming the recorded angle-specific raw data frames, and 4) compounding the beamformed data frames over all angles to form a final image. This thesis attempts to address the following question: Given a set of available plane-wave emission angles, which ones should we select for acquisition (i.e., which angle-specific raw data frames should we sample), to achieve adequate image quality at low cost associated with both sampling and computation?

We propose a simple similarity-driven angle selection scheme and evaluate its several variants that rely on user-specified similarity measurement thresholds guiding the recursive angle selection process. Our results show that the proposed scheme has a low

computational overhead and can yield significant savings in terms of the amount of sampled raw data.

Table of Contents

List of Tables

List of Figures

Acknowledgements

First and foremost, I would like to express my most sincere gratitude to my supervisor Dr. Daler Rakhmatov for his encouragement, support and guidance throughout the course of my Master's program. Further, I thank Dr. Mihai Sima (my supervisory committee member) and Dr. Venkatesh Srinivasan (my external examiner) for their help with improving my thesis.

Dedication

To My Parents (Muhammad Akbar and Rashida Akbar),

For their endless love, support and encouragement

List of Acronyms

Coherent Plane-Wave Compounding	CPWC
Compressive Sensing	CS
Contrast-to-Noise Ratio	CNR
Delay and Sum [Beamforming]	DAS
Full Acquisition	FA
Full Width Half Maximum	FWHM
Index Measurement of Mean Squared Error	IMMSE
Region of Interest	ROI
Structural Similarity Index Measurement	SSIM
Similarity-Driven Angle Selection	SAS
Similarity-Driven Angle Selection using Raw Data	SASR
Similarity-Driven Angle Selection using Beamformed Data	SASB
Uniform Angle Selection	UAS

Chapter 1: Introduction

Ultrasound imaging is widely used as a diagnostic tool in medical clinics due to its safety, portability, non-invasiveness, and the ease of use. Figure 1.1 shows the block diagram of a typical ultrasound system.

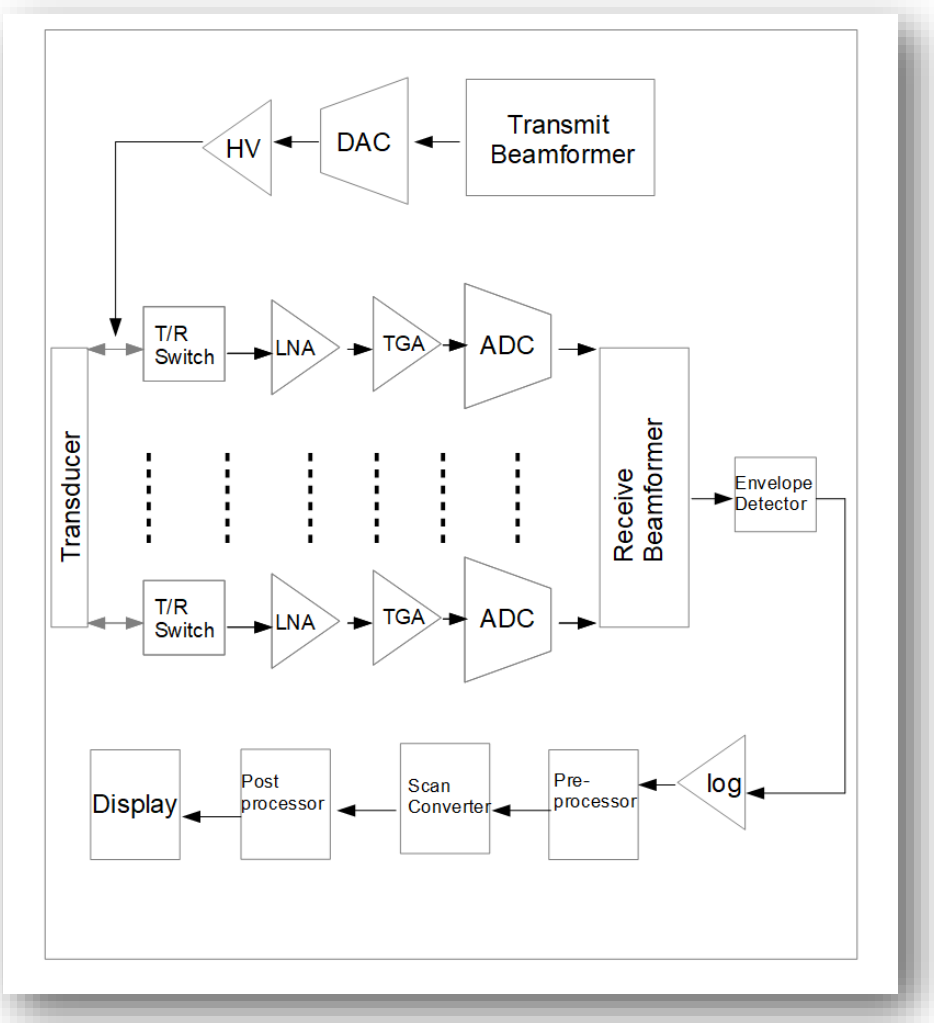

Figure 1.1: Typical ultrasound system.

Usually, an ultrasound system works with frequencies ranging from 2MHz to 18MHz [1]. A transducer array consisting of piezoelectric elements is excited by electric pulses to produce sound waves. The returning echoes generated because of these sound waves are recorded and processed to create an image. As the number of the elements mounted on the transducer array is increased, spatial resolution of the image improves. The pre-defined spacing between elements on the transducer array, referred to as inter-element spacing, should be less than or equal to half of the wavelength in order to suppress the grating lobes. These grating lobes are responsible for causing artifacts and reducing image quality, especially contrast [3].

Steering and (if needed) focusing of the ultrasound beam play a key role in the transmit beamforming process. Its purpose is to apply appropriate transmit delays to equalize the phase of different signals emitted from different transducer elements. This is followed by the conversion of transmit-beamformed digital signals to analog signals using a digital-to-analog converter (DAC), followed by a high voltage (HV) amplifier that drives the transducer elements.

During reception, the returning echoes pass through transmit-receive (T/R) switches that separate the HV amplifiers from harming the echoes. Low noise amplifiers (LNAs) are responsible for improving the signal-to-noise ratio (SNR). Some of the returning echoes are attenuated differently depending on their propagation depth. Therefore, they are passed through a time gain compensation (TGC) block and the resulting analog signals are then passed through an analog-to-digital converter (ADC). The resulting digital signals enter a receive beamformer.

Receive beamforming is followed by envelope detection. It involves taking the absolute value of a complex-valued analytic signal, whose real part is the beamformed signal itself and imaginary part is the Hilbert transform of the beamformed signal [4]. Next, a logarithmic compression takes place to reduce the dynamic range of the beamformed data [2]. The resulting data may need to be spatially remapped (i.e., scan converted), which is achieved via interpolation with appropriate pre- and post-processing, if needed.

Chapter 2: Background

2.1 Ultrasound Image Quality

In this section we discuss several important quantitative factors related to the quality of ultrasound images.

2.1.1 Spatial Resolution

Axial resolution quantifies the ultrasound system's ability to differentiate between objects positioned along the direction of ultrasound wave propagation. It is given by [5]:

$$R_{axial} = \frac{cM}{2f_0} = \frac{\lambda M}{2}, \tag{2.1}$$

where c, f_0, and λ are the speed, frequency, and wavelength of the transmitted pulse, respectively, while M is the number of periods of the transmitted pulse. The factor of 2 is needed to account for the round-trip delay of the wave pulse. As can be seen in the above equation, the higher the frequency, the greater the axial resolution. However, increasing the frequency leads to increased attenuation [6].

The other type of spatial resolution is called lateral resolution, which quantifies the ultrasound system's ability to differentiate between objects positioned perpendicular to the direction of ultrasound wave propagation. It is given by [5]:

$$R_{lateral} = \frac{\lambda z}{D} = \lambda F\#, \tag{2.2}$$

where z is the imaging depth, D is the width of the active aperture, and $F\#$ represents the so-called F-number of the imaging system.

2.1.2 Contrast

Given some region of interest, let s_{in} denote the mean signal value inside that region, and let s_{out} denote the mean signal value outside that region. Then, contrast can be defined as

$$Contrast = \frac{S_{out} - S_{in}}{S_{out}}. \quad (2.3)$$

The most prominent factor that causes degradation in the contrast values is the presence of sidelobes and grating lobes [7].

2.1.3 Frame Rate

Let F_{rate} denote the frame rate in the units of frames per second, and let N_l denote the number of sequentially imaged scan lines. Then, we have the following relationship:

$$F_{rate} \times z \times N_l = \frac{c}{2}. \quad (2.4)$$

Increasing the scan line density improves the lateral resolution, but decreases the frame rate, assuming fixed imaging depth *z*. If we want to increase the imaging depth, we need to either decrease the frame rate or decrease the number of scan lines [5].

2.2 Delay-and-Sum (DAS) Beamforming

Figure 2.1 shows the basic structure of a delay-and-sum (DAS) receive beamformer commonly used in ultrasound imaging. Time delays $\tau_1, \tau_2, \ldots \tau_M$ are used to focus the received data from M sensors. They are determined based on the distance travelled by a reflected wave from the region of interest to a particular sensor of the transducer array, relative to a certain reference position, usually the center of the transducer array. The delayed signals, forming the input vector $x_1[n], x_2[n], \ldots x_M[n]$, are multiplied by their respective beamforming weights $w_1[n], w_2[n], \ldots w_M[n]$ and then summed to produce the output signal $y[n]$. The set of weights can be either fixed (data-independent) or adaptive (data-dependent).

Standard window functions (e.g. rectangular, Hamming, Kaiser, etc) are commonly used to realize the fixed weights. The choice of a particular window depends on the desired balance between the mainlobe width and the sidelobe level, which translates into a balance between the image resolution and contrast. There is a fundamental trade-off between the image contrast and resolution: reducing the sidelobe level leads to a wider mainlobe width and vice versa.

On the other hand, adaptive beamforming is capable of achieving a narrow mainlobe width and at the same time suppress the sidelobe levels, thus improving both the image resolution and contrast in comparison to fixed beamforming. Such an improvement comes at a significant computational cost, as the data-dependent adaptive weight vector must be continuously recalculated based on the characteristics of received input vectors.

The adaptive beamforming weights are also dependent on the choice of criterion that preserves the desired signal while rejecting the unwanted interference and noise [9].

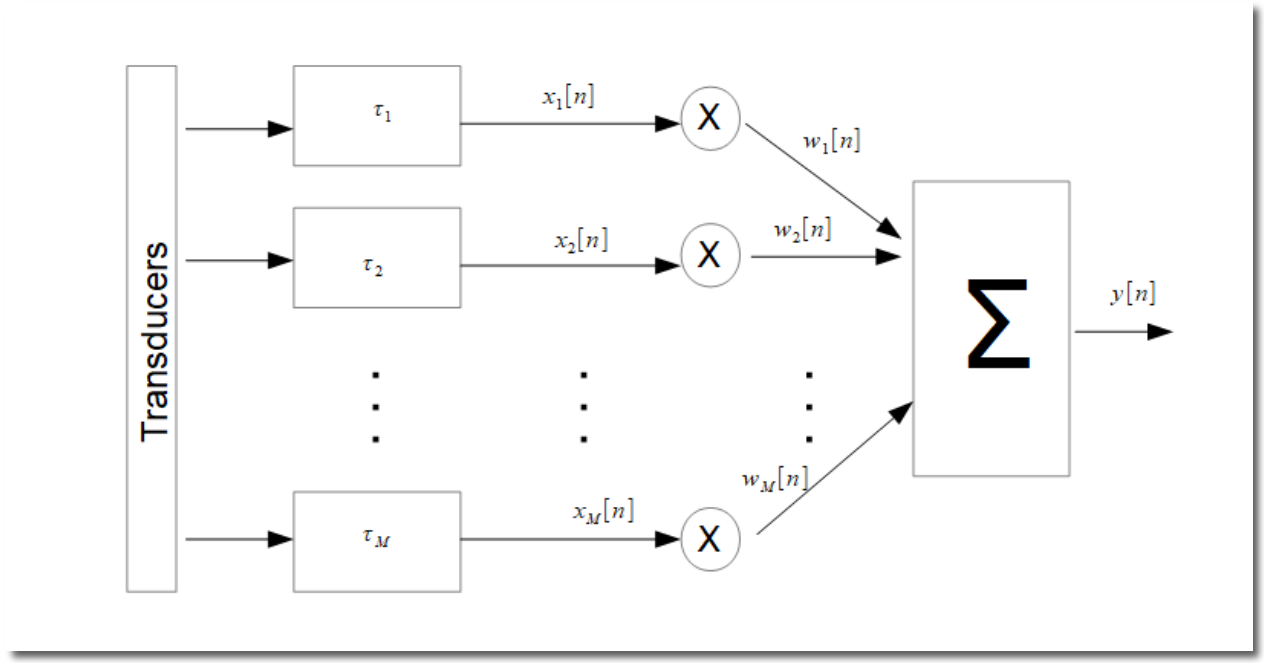

Figure 2.1: Delay-and-Sum (DAS) beamformer.

In this work, we use a non-adaptive DAS beamformer with the rectangular window. To adapt our discussion of DAS beamforming to plane-wave imaging, we shall closely follow Montaldo *et al.* [48] and change our notation accordingly. First, let $BF(z, x_p)$ denote a beamformed data point at a pixel location (z, x_p), and let $RF(t, x_s)$ denote a raw RF data point sampled at time t by a sensor located at position x_s. Note that x_p and x_s represent to the lateral coordinates, while $RF(t, x_s)$ represents one of the inputs supplied to the DAS beamformer, and $BF(z, x_p)$ represents the output of the DAS beamformer, where $z = ct/2$ is the axial coordinate.

In order to compute $BF(x_p, z)$, we need to sum individual values $RF\big(\tau\big(x_s, x_p, z\big), x_s\big)$ over all sensor locations x_s. Each $RF\big(\tau\big(x_s, x_p, z\big), x_s\big)$ in question is obtained from the corresponding $RF(t, x_s)$ signal value interpolated at $t = \tau\big(x_s, x_p, z\big)$. Reference [48] have shown that for a plane-wave pulse emitted at an angle θ:

$$\tau\big(x_s, x_p, z\big) = \left(z\cos(\theta) + x_p \sin(\theta) + \sqrt{z^2 + \big(x_s - x_p\big)^2}\right) / c. \tag{2.5}$$

Note that after acquiring a full θ-specific raw RF data frame $RF(t, x_s)$ over all x_s and t values, computing the corresponding beamformed data frame $BF(z, x_p)$ over all x_p and z values involves as many as M×S×T calculations of $\tau\big(x_s, x_p, z\big)$, where M is the number of sensors (i.e., the number of processed x_s values), S is the number of image scan lines (i.e., the number of processed x_p values), and T is the number of acquired input vector snapshots (i.e., the number of processed t values, which is the same as the number of processed z values).

Figure 2.2 shows a pseudo code snippet for the basic DAS beamformer used in this work. Its inputs are an M-element array of x_s values, an S-element array of x_p values, a T-element array of $z = ct/2$ values, and a 2D raw RF data frame $RF(t, x_s)$ denoted by *RawData*[j], where index j identifies a specific plane-wave emission angle among N available angular values $\theta_1, \theta_2, \ldots, \theta_N$. The output is a 2D beamformed data frame $BF(z, x_p)$ denoted by *BeamformedData*[j]. To produce each data point of the latter, our raw data frame $RF(t, x_s)$ is first interpolated along the t-axis using the calculated values of $\tau\big(x_s, x_p, z\big)$ given by equation 2.5, and then it is summed along the x_s-axis. As detailed

later in this thesis, we will be using M = S = 128, T = 3328, N = 75, and our interpolation will be based on the nearest-neighbor scheme.

```
% index j identifies specific emission angle
raw_frame = RawData[j];                    % 2D raw RF data frame
for n = 1:T
      for k = 1:S
            tau_tx = (z[n]*cos(theta[j]) + xp[k]*sin(theta[j]))/c;
            for m = 1:M
                  tau[m] = tau_tx + sqrt(z[n]^2 + (xs[m] - xp[k])^2)/c;
            end
            beamformed_frame[n][k] = sum(interpolate(raw_frame, tau));
      end
end
BeamformedData[j] = beamformed_frame;      % 2D beamformed data frame
```

Figure 2.2: Pseudo code snippet for DAS beamforming.

2.3 Coherent Plane-Wave Compounding (CPWC)

After DAS beamforming of the raw RF data frames over all plane-wave emission angles $\theta_1, \theta_2, \ldots, \theta_N$, we have as many as N beamformed data frames. The next step is called *coherent plane-wave compounding* (CPWC), whose purpose is to improve the resolution and contrast quality of a final image [48]. CPWC simply means that the beamformed data frames in question are summed together to form a single compounded beamformed

frame, before performing envelope detection and log compression to obtain the final image.

In this thesis, we ask the following question: Given an ordered set of angular values $\theta_1, \theta_2, \ldots, \theta_N$, which ones should we select for acquisition to achieve adequate image quality at low cost associated with both sampling and computation? In other words, our objective is to come up with an angle selection scheme deciding on a subset of indices *j* among 1, 2, …, N, which implies the acquisition of the selected raw RF data frames *RawData*[*j*], yielding the corresponding beamformed data frames *BeamformedData*[*j*]. Summation of the latter over the selected *j* indices will produce the final beamformed frame, to be used for generating the final image. Selecting fewer emission angles means compounding fewer beamformed frames, which translates into savings in terms of the number of acquired raw RF data frames, but may negatively affect image resolution and contrast. Chapter 3 describes our proposed scheme for angle selection that attempts to strike a balance between sample savings and image quality.

Another way to reduce data acquisition costs is called *compressive sensing*. It is highly effective and has been extensively studied in the signal processing literature. The next section briefly outlines this well-known technique.

2.4 Compressive Sensing (CS)

The Shannon-Nyquist theorem states that the sampling frequency should be at least twice the highest frequency contained in the signal. Compressive sensing (CS) improves on this fundamental result: it allows for the reconstruction of a signal from a linear combination

of a small number of random measurements. CS is a fairly recent concept in the field of medical imaging, and it has quickly gained popularity.

A simple method for data compression would be to compute a signal from its frequency components and then encode the location and values of the most significant coefficients for later decoding. Such a process requires knowledge of all the coefficients of the signal. Although their locations may not be known in advance, they tend to be clustered around edges in an image, which is an example of *sparsity* that has become a fundamental modeling tool in signal processing [39, 46]. According to [12, 30], sparsity expresses the idea that the information rate of a continuous time signal could be much lower than conveyed by its bandwidth, or that a discrete-time signal depends on a number of degrees of freedom, which could be much smaller than its length. CS takes advantage of the fact that many natural signals are sparse or compressible in the sense that they have concise representation using a proper basis. In [23, 38], the authors have suggested several ways for improving CS application to medical imaging: optimizing sampling trajectories, developing improved sparse transforms that are incoherent in relation to a sampling operator, targeting reconstruction quality of clinically significant image sections, and improving the speed of reconstruction algorithms. In this section, we provide a basic introduction to CS and describe how we have applied it in our case.

Let $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}$ denote a one-dimensional signal to be reconstructed, and let $\boldsymbol{y} \in \mathbb{R}^{\boldsymbol{m}}$ denote a small number of random measurements of $\boldsymbol{x}$, where $m < n$. We rely on the assumption that $\boldsymbol{x}$ has a compressed representation in some orthogonal *model basis* Ψ:

$$\boldsymbol{x} = \Psi \mathbf{v}, \tag{2.6}$$

where $\mathbf{v}$ is an s-sparse vector having $s < m < n$ non-zero coefficients. According to [12], $\boldsymbol{y}$ may be acquired in some *sensing basis* Φ, which can be viewed as an $m \times n$ matrix having non-zero entries at random locations, with the rest of the entries set to zero. Then,

$$\boldsymbol{y} = \Phi\mathbf{x}, \tag{2.7}$$

or equivalently,

$$\boldsymbol{y} = \Phi\Psi\mathbf{v} = \mathbf{A}\mathbf{v}, \tag{2.8}$$

where $\mathbf{A} = \Phi\Psi$, which is an $m \times n$ full-rank matrix. The CS theory [12, 13, 26] proves that sparsity allows for an exact recovery of $\boldsymbol{v}$ with overwhelming probability for a certain class of matrices Φ and Ψ: the sensing basis must be *incoherent* with the model basis, which is ensured by the randomness of the non-zeros entries in Φ [13].

In such settings, Candès in [12] and Boyd and Vanderberghe in [28] showed that the CS reconstruction problem can be solved through the following l_0-minimization problem:

$$\hat{\boldsymbol{v}} = argmin||\boldsymbol{v}||_0, \text{ subject to } \boldsymbol{y} = \mathbf{A}\mathbf{v}, \tag{2.9/P0}$$

where $||\boldsymbol{v}||_0 = |\{i, v_i \neq 0\}|$. Formulation (2.9)/P0 enforces the choice of the sparsest solution $\hat{\boldsymbol{v}}$ among all possible solutions, from which one can obtain the reconstructed signal as follows: $\hat{\boldsymbol{x}} = \Psi\hat{\boldsymbol{v}}$.

In our case, we have a data volume comprised of N raw RF data frames, as illustrated in figure 2.3(a). Using CS, we wish to sample a limited subset of points forming our data volume and recover the rest via optimization. Rather than trying to randomly subsample and then reconstruct individual "vertical" planes, labeled *A* in figure 2.3(a), for each

emission angle under consideration (which is a typical approach), we have found that it is better (for our evaluation dataset discussed in chapter 4) to randomly subsample and then reconstruct individual "horizontal" planes, labeled *B* in figure 2.3(a), for each time instance under consideration.

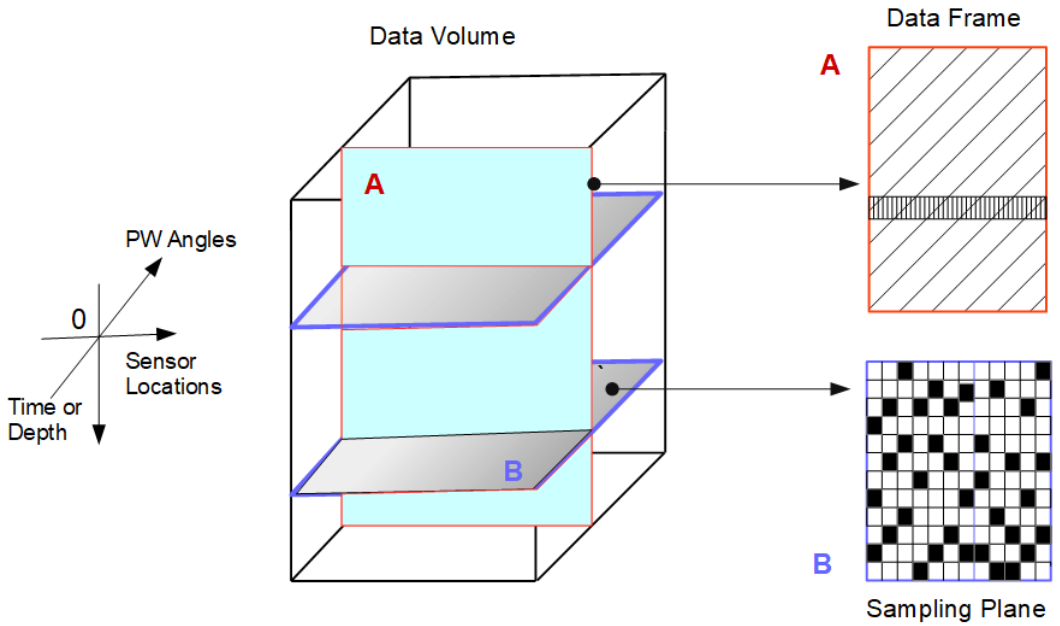

Figure 2.3(a): Proposed CS application.

Figure 2.3(b) shows the pseudo code for our CS application technique used in this work. The 3D variable *DataVolume* represents all N raw RF data frames, each having T rows and M columns. Recall that T and M refer to the number of sampling time instances and the number of sensors, respectively. For any fixed row *n* and column *m*, the signal vector $\boldsymbol{x}$ = *DataVolume*[*n*][*m*] holds the N-element data along the angular axis $(\theta_1, \theta_2, \ldots, \theta_N)$. We pad $\boldsymbol{x}$ with zeroes on both sides, and then compute its Fourier representation $\boldsymbol{v}$ using the W×W Discrete Fourier Transform (DFT) matrix, where W is the length of our padded

signal $\boldsymbol{x}$. The DFT matrix serves as Ψ^{-1} according to equation (2.6). To obtain the matrix $\boldsymbol{A}$ from equation (2.8), we simply take K rows of Ψ at random, whose indices correspond to K randomly sampled elements of $\boldsymbol{x}$, forming the measurement vector $\boldsymbol{y}$. The number of measurements K is determined by the compression factor variable *cf*.

```
% cf  – compression factor
% tol – duality gap tolerance
% pad – padding subvector of zeros
W = N + 2*length(pad);              % size of signal to be reconstructed
K = round(W/cf);                    % number of measurements to take
Psi_inv = dftmtx(W);                % W-by-W DFT matrix
Psi = conj(Psi_inv)/W;              % inverse DFT matrix (norm. conjugate)
qK = random(T, K, W);               % T-by-K matrix of random indices 1:W
for each row n = 1:T                % rows = time instances
      for each column m = 1:M       % columns = sensor positions
            x = DataVolume[n][m];   % angular data vector
            x = [pad x pad];        % padded angular data vector
            v = Psi_inv * x;        % Fourier representation of x
            q = qK[n];              % vector of K random indices 1:W
            y = x[q];               % randomly sampled elements of x
            A = Psi[q];             % random rows of inverse DFT matrix
            v_hat = optimize(A, y, tol);        % sparse solution
            x_hat = real(Psi * v_hat);          % reconstructed signal
      end
end
```

Figure 2.3(b): Pseudo code for simulating CS.

Rather than solving the l_0-minimization problem (2.9)/P0, which is computationally difficult, we follow the standard practice of using its convex l_1-approximation:

$\boldsymbol{v}\hat{} = argmin||\boldsymbol{v}||_1$, subject to $\boldsymbol{y} = \mathbf{A}\mathbf{v}$, (2.10)/P1

where $||\boldsymbol{v}||_1 = \sum_i |v_i|$. The above problem is solved using the L1-MAGIC package [35], a collection of MATLAB routines for solving convex programming problems central to CS. Specifically, we employ the "l1eq_pd" routine that solves (2.10)/P1-type problems (l_1-minimization under equality constraints) using a primal-dual interior-point method with a duality gap tolerance specified by the variable *tol*. After obtaining $\boldsymbol{v}\hat{}$, we compute reconstructed $\boldsymbol{x}\hat{}$ by taking the real part of $\Psi\boldsymbol{v}\hat{}$, where Ψ is the inverse DFT matrix.

2.5 Related Work

Ultrasound imaging literature has been growing steadily with significant focus on the application of various saving techniques in areas of efficient sampling and computation [8, 10, 18, 20, 23, 26, 32, 46]. Liebgott, Basarab, Kouame, Bernard, and Friboulet in [13] came up with classification based on the sparsity assumptions about the scatterer distribution, the prebeamformed data, the postbeamformed data, and the Doppler imaging data. The sparse diffusion map model [13] assumes that most of the scatterers have an echogenicity close to zero. The authors of [13] have shown that this technique keeps only the strongest echoes, ignoring the weak ones which can be vital to the reconstruction of the speckle patterns. The sparse raw RF data model [13] assumes that the raw channel data gathered at each transducer element during reception have a sparse form in some

basis. This naturally helps put forth a clear objective: reduce the quantity of pre-beamformed data acquired. The authors of [13] believe that acquiring the data this way is probably as difficult as just acquiring all of the channel data. The more direct approach in terms of savings would be to remove entire columns of the raw dataset which would entail disconnecting some transducer elements of the ultrasound probe. This most advantageous case for this approach would be 3D imaging with matrix arrays. Assuming the sparsity of the beamformed dataset, on the other hand, the authors of [13] successfully reconstructed 2D image frames using a gradient-descent algorithm within the Bayesian framework. Doppler imaging is another potential application for CS. For example, Richy, Liebgott, Prost, and Friboulet in [14] applied CS on duplex ultrasonography that allows for simultaneous visualization of the inner structure and the blood flow in a particular section in the body. According to [13], traditional strategies either halve the maximum measurable velocity, or introduce gaps in the flow data. Meanwhile, Jensen in [29] proposed a method for preserving the full velocity range measurements based on sparse datasets, evaluated in part using the popular *Field II* simulation package [36, 27].

Building on [13], the authors of [11] discuss the feasibility of CS for the reconstruction of raw RF data frames. This application involves selecting a representation basis where the data to be reconstructed has a sparse expansion. As the data typically consists of warped oscillatory patterns, the authors of [11] proposed to use a sparse representation based on wave atoms [47, 37]. Wave atoms appeared to be a better fit in comparison to Fourier and Daubechies wavelets in terms of sparsely representing warped oscillatory patterns. Using wave atoms and sampling 20% of the channel RF data, the authors in [13] managed to reconstruct missing data samples without exceeding a 5-dB error range.

The authors of [43] improved upon signal reconstruction from a small universal sample of Fourier measurements, using advanced methods of geometric functional analysis and probability theory in Banach spaces. The authors of [25] came up with a compressed beamforming approach based on the idea of *Xampling*, which combines the classic methods from sampling theory with recent developments in CS [44, 45]. It involves applying low-rate sampling schemes to individual transducer elements, whose analog input signals are prefiltered in advance [33]. In [21], Chernyakova and Eldar generalized the concept of compressed beamforming and showed 4-10-fold reduction in sampling rate can be achieved. Also, Cohen, Sde-Chen, Chernyakova, Fraschini, Bercoff, and Eldar previously demonstrated in [50] that a frequency-domain DAS beamformer using only a quarter of input samples was capable of producing similar-quality image frames as those obtained by a conventional DAS beamformer processing all data samples.

The authors in [22] have tried to address the question of achieving good image quality at low cost by using machine learning techniques. The paper claims to reduce the number of emitted plane waves by training a convolutional neural network to reconstruct ultrasound images. Full compounding relied on 31 plane waves, but the authors of [22] were able to obtain high-quality CPWC results using only 3 plane waves.

2.6 Our Contribution

The main objective of our work is to reduce the number of acquired raw RF data samples without significant degradation in the ultrasound image quality. Our approach relies on *similarity-driven plane-wave emission angle selection scheme* to sample appropriate raw

RF data frames that appear to have sufficiently small redundancy with respect to the already acquired data. In our study, we have used well-known Index Measurement Mean Squared Error (IMMSE) and Structural Similarity Index Measurement (SSIM) as our data similarity (i.e., redundancy) metrics with the corresponding user-defined thresholds, to guide the selection process described in chapter 3.

In chapter 4, we have applied our scheme to the two experimental datasets from the *Plane-wave Imaging Challenge in Medical UltraSound* (PICMUS) [49], acquired using 75 plane waves whose emission angles ranged from –16° to +16°. Each dataset is available in RF (modulated) and IQ (demodulated) format; we have used the former. In addition to evaluating the imaging performance of our proposed angle selection scheme with respect to full data acquisition, chapter 4 also provides comparisons against our CS technique described in section 2.4.

To summarize, this thesis makes the following contributions that have been overlooked in the existing literature on ultrasound imaging:

- We propose a simple scheme for deciding which angle-specific raw RF data frames to select for acquisition and subsequent coherent compounding, so that the amount of sampled data is reduced, while the resulting image resolution and contrast are still acceptable.
- We provide quantitative evaluation results for four variants of our similarity-driven scheme applied to real-world experimental ultrasound data. Each variant is a combination of using either SSIM or IMMSE as a similarity metric, and using either pre-beamformed or post-beamformed data for similarity measurements.
- We also evaluate our CS technique, which is new as well.

Chapter 3: Similarity-Driven Angle Selection

Our proposed similarity-driven angle selection (SAS) scheme is based on the classic binary search algorithm. The latter is a half-interval target-search technique, whose pseudo code is shown in figure 3.1(a), applied to a sorted *n*-element array *A*, where *T* is the target value.

```
binary_search(A, n):
    L = 0
    R = n - 1
    while L <= R:
        m = floor((L + R) / 2)
        if A[m] < T
            L = m + 1
        else if A[m] > T
            R = m - 1
        else
            return m
    return Nil
```

Figure 3.1(a): Binary search pseudo code.

Figure 3.1(b) illustrates finding the target value $T = 15$ in a sorted array with $n = 12$ elements. The worst-case time complexity of the binary search algorithm is $O(\log n)$.

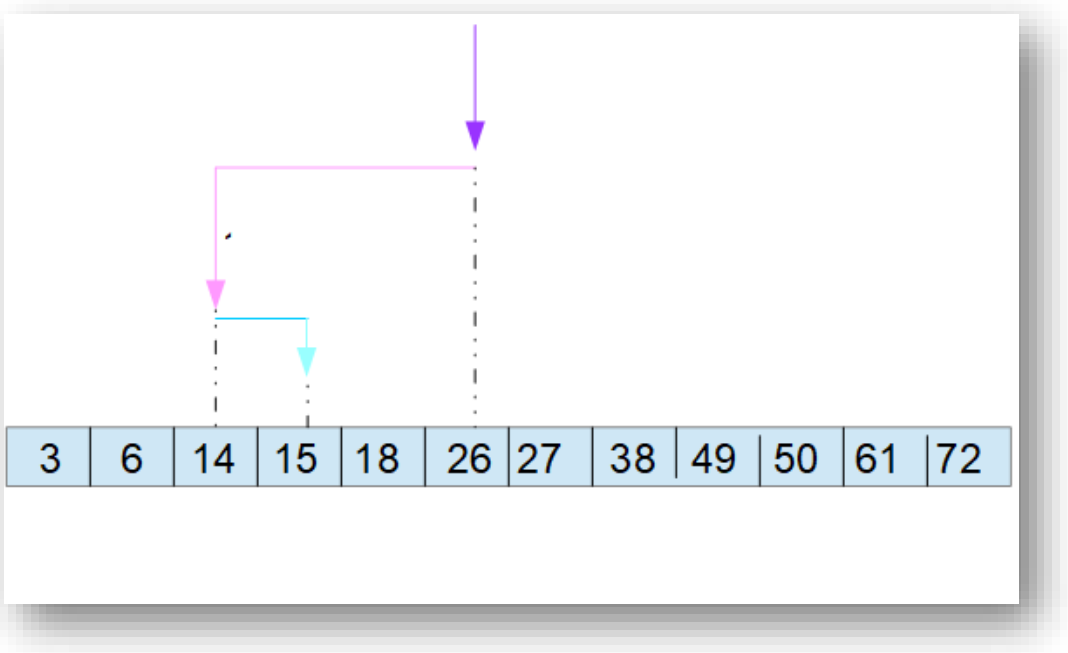

Figure 3.1(b): Illustration of the binary search execution.

In our case, the sorted array consists of indices 1, 2, …, N representing the plane-wave emission angles. For example, the PICMUS benchmark dataset (see chapter 4) contains N = 75 raw RF frames corresponding to the plane-wave emission angles ranging from θ_{min} = –16° (index 1) to θ_{max} = +16° (index 75), with the angular step of 0.432° per index increment. Our SAS scheme starts by acquiring RF frames 1 and N, which yields our initial reference (compounded) image. The reference image data is continuously updated as new frames are acquired and compounded. The decision to acquire a new frame is based on the outcome of comparing a user-chosen data similarity metric with a user-defined threshold value, which will be described in detail later in this chapter.

3.1 Similarity Metrics

In this work, we have used the *structural similarity index measurement* (SSIM) and *index measurement of mean squared error* (IMMSE) values as two possible choices of a data similarity metric.

3.1.1. Index Measurement of Mean Squared Error (IMMSE)

IMMSE takes two quantities, a known reference image data and a newly observed image data and measures the mean point-wise squared error (MSE) between them. The closer it is to zero, the greater the similarity between the two images in question.

The MSE is the second moment of the error and hence incorporates both the variance and bias (i.e., difference between expected value and the true value of a parameter being estimated). The MSE of an estimate $X\hat{}$ with respect to an unknown parameter X is defined as

$$MSE(\hat{X}) = E[(\hat{X} - X)^2], \tag{3.2}$$

where E denotes the expected value. We have

$$MSE(\hat{X}) = Var_{\hat{X}}(\hat{X}) + Bias_{\hat{X}}(\hat{X}, X)^2, \tag{3.3}$$

where $Var_{\hat{X}}$ represents variance with respect to $\hat{X}$ and $Bias_{\hat{X}}$ represents the bias with respect of $\hat{X}$.

An important feature of IMMSE is that it heavily weighs outliers, highlighting large differences more than small ones.

3.1.2. Structural Similarity Index Measurement (SSIM)

Wang *et al.* [15] introduced SSIM as a flexible image quality assessment metric, which is specifically designed for measuring the similarity between two images. SSIM targets the following three characteristics of images: *luminance*, *contrast*, and *structure*. The luminance of the surface on an object being observed is the product of illumination and the reflectance, but the structures of the objects in the scene are independent of the illumination [15]. Therefore, to use the structural information in an image, the illumination information is discarded. Also, since luminance and contrast can vary across a scene, the local luminance and contrast are used. Consequently, the overall SSIM value, obtained for two images *x* and *y*, is given by

$$SSIM(x,y) = [l(x,y)]^{\alpha}.[c(x,y)]^{\beta}.[s(x,y)]^{\gamma}, \tag{3.4}$$

where

$$luminance = \ l(x,y) = \frac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_y^2+C_2},$$

$$contrast = c(x,y) = \frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2+\sigma_y^2+C_2},$$

$$structure = s(x,y) = \ (\sigma_{xy}+C_3)\frac{1}{\sigma_x\sigma_y+C_3},$$

where μ_x, μ_y, σ_x, σ_y, and σ_{xy} are the local means, standard deviations, and cross-covariance, respectively, while C_1, C_2, and C_3 are some constants. Note that $\alpha > 0$, $\beta > 0$ and $\gamma > 0$ are parameters used to adjust the relative importance of the three components. Default SSIM settings (used in this work) are $\alpha = \beta = 0 = \gamma = 1$ and $C_1 = C_2 = C_3 = 0$.

3.2. Similarity-Driven Angle Selection Using Beamformed Data (SASB)

Our first proposed method for raw RF data frame acquisition is called *"Similarity-driven Angle Selection using Beamformed data"*, abbreviated as SASB in the sequel. It involves the following basic steps outlined below:

1. Acquire raw RF data frames 1 and N, denoted by *RawData*[1, N].
2. Apply DAS beamforming to *RawData*[1, N] to obtain the corresponding beamformed data frames, denoted by *BeamformedData*[1, N].
3. Let *angles* = [1, N], *pivot1* = 1, and *pivot2* = N.
4. [*angles, BeamformedData*] =

   SASB (*angles, pivot1, pivot2, threshold, BeamformedData*).
5. Perform beamformed frame compounding, envelope detection, and log-compression to obtain the final image.

The 3D array variable *RawData* holds the raw 2D RF data frames 1, 2, 3, …, N–1, N. Initially, *RawData*[2, 3, …, N–1] is filled with zeros, i.e., we start with *RawData*[1, N] in step 1. Raw RF data frames 1 and N as expected to carry the most amount of differing information since they correspond to the extreme emission angle values. Recall that in our case of PICMUS datasets, our extreme indices 1 and N = 75 correspond to the plane-wave emission angles θ_{min}= –16° and θ_{max}= +16°, respectively.

In step 2, both *RawData*[1] and *RawData*[N] undergo DAS beamforming to obtain the beamformed data frames *BeamformedData*[1, N]. After this, the recursive function SASB is called; it has five input arguments and two output arguments. The SASB input arguments are the 1D array variable *angles* holding the currently sampled raw RF data

frame indices ([1, N] initially), the two boundary indices of the index search interval, denoted by *pivot1* and *pivot2* (1 and N initially), a user-specified *threshold* value for a chosen similarity metric to guide the next angle selection, and the 3D array variable *BeamformedData* holding the currently beamformed data frames to be compounded later. The SASB output arguments are the updated variable *angles* containing both previous and newly selected emission angle indices, and the updated variable *BeamformedData* that contains the corresponding beamformed data frames for all the indices recorded in *angles* upon return from the recursive SASB function call in step 4. Sections 3.2.1 and 3.2.2 provide further implementation details on our SASB function.

In step 5, the *BeamformedData* frames are summed, which yields the 2D compounded pre-image data. Then, we apply envelope detection and log-compression to obtain the final 2D image. This image is the result of sampling a limited subset of raw RF data frames, selected by our SASB function among emission angle indices 1, 2, 3, …, N–1, N.

3.2.1. SASB Using SSIM (SASB-SSIM)

Figure 3.2(a) shows the pseudo code of our proposed recursive SASB function that uses SSIM as the similarity metric, written as SASB-SSIM.

Given inputs *pivot1* and *pivot2*, the next emission angle index, denoted by *new_angle*, is calculated as their rounded mean and checked whether it is not already contained in *angles* (the set of already sampled indices). For example, starting with initial *pivot1* = 1 and intial *pivot2* = N = 75, we obtain *new_angle* = 38. If the emission angle θ_{new_angle} has not been encountered before, the corresponding raw RF data frame is acquired and

stored as *RawData*[*new_angle*]. Applying DAS beamforming to *RawData*[*new_angle*] yields new *BeamformedData*[*new_angle*].

Next, we introduce the variables *arg1* = *BeamformedData*[*angles*, *new_angle*] and *arg2* = *BeamformedData*[*angles*], i.e., *arg1* contains the same previously beamformed data frames as *arg2*, plus new *BeamformedData*[*new_angle*]. We let the reference "image" be *abs*(*sum*(*arg2*))/L, and we let the new "image" be *abs*(*sum*(*arg1*))/(L+1), where L is the length of the *angles* array (the number of previously beamformed data frames). In other words, we compound (sum) the beamformed data frames in question, take the absolute values of the resulting 2D data, and normalize them by the number of frames summed. Then, we make use of SSIM to quantify similarity, recorded in the variable *sim*, between our reference "image" and new "image".

If *sim* ≤ *threshold*, it means that a sufficient degree of similarity has been reached, and our SASB-SSIM function can stop searching further: the amount of new "information" brought by *BeamformedData*[*new_angle*] into *BeamformedData*[*angles*] was relatively insignificant. Otherwise, when *sim* < *threshold*, we are directed to acquire more frames. To decide which half-interval to explore first (between boundaries *pivot1* and *new_angle*, or between boundaries *new_angle* and *pivot2*), we calculate the SSIM-based similarities between *abs*(*BeamformedData*[*new_angle*]) and *abs*(*BeamformedData*[*pivot1*]), as well as between *abs*(*BeamformedData*[*new_angle*]) and *abs*(*BeamformedData*[*pivot2*]); these similarity values are captured by the variables *temp1* and *temp2*, respectively. If *temp1* < *temp2*, we explore the left half-interval first (between boundaries *pivot1* and *new_angle*), followed by exploring the right half-interval (between boundaries *new_angle* and *pivot2*). Otherwise, we explore the right-interval first, followed by exploring the left

half-interval. The rationale behind this approach is to target a half-interval whose boundary “images” are more dissimilar first (it may offer more new “information”), and afterwards search the other half-interval, whose exploration may terminate earlier.

To explore the left half-interval, our SASB-SSIM function calls itself with updated *pivot2 = new_angle*. On the other hand, to explore the right half-interval, our SASB-SSIM function calls itself with updated *pivot1 = new_angle*. Note that in both cases, it uses updated *angles* = [*angles*, *new_angle*] and updated *BeamformedData* that includes the *new_angle* frame.

```
[angles, BeamformedData] =
            SASB-SSIM (angles, pivot1, pivot2, threshold, BeamformedData):


new_angle = round((pivot1 + pivot2)/2);
if (new_angle does not belong to angles)
      Acquire new raw RF data frame RawData[new_angle];
      BeamformedData[new_angle] = DAS(RawData[new_angle]);
      arg1 = BeamformedData[angles, new_angle];       % new data
      arg2 = BeamformedData[angles];                  % reference data
      L = length(angles);
      angles = [angles, new_angle];
      sim = SSIM(abs(sum(arg1))/(L+1), abs(sum(arg2))/L);
      if sim < threshold            % smaller sim values indicate lesser similarity
            temp1 =
              SSIM(abs(BeamformedData[new_angle]), abs(BeamformedData[pivot1]));
            temp2 =
              SSIM(abs(BeamformedData[new_angle]), abs(BeamformedData[pivot2]));
            if temp1 < temp2        % explore left half-interval first
                  [angles,BeamformedData] =
                    SASB-SSIM(angles, pivot1, new_angle, threshold, BeamformedData);
                  [angles,BeamformedData] =
                    SASB-SSIM(angles, new_angle, pivot2, threshold, BeamformedData);
            else                    % explore right half-interval first
                  [angles,BeamformedData] =
                    SASB-SSIM(angles, new_angle, pivot2, threshold, BeamformedData);
                  [angles,BeamformedData] =
                    SASB-SSIM(angles, pivot1, new_angle, threshold, BeamformedData);
            end
      end
end
```

Figure 3.2(a): Recursive SASB using SSIM.

3.2.2. SASB Using IMMSE (SASB-IMMSE)

Figure 3.2(b) shows the pseudo code of our proposed recursive SASB function that uses IMMSE as the similarity metric, written as SASB-IMMSE.

The main difference between SASB-IMMSE and SASB-SSIM (described in the previous section) is that the former relies on IMMSE instead of SSIM to calculate our similarity variables *sim, temp1*, and *temp2*. Unlike SSIM, larger IMMSE values indicate a lesser degree of similarity; therefore, the inequalities in our if-statements change from "less than" to "greater than". Other than that, the pseudo code is the same for both functions.

3.3. Similarity-Driven Angle Selection Using Raw Data (SASR)

Our second proposed method for raw RF data frame acquisition is called *"Similarity-driven Angle Selection using Raw data"*, abbreviated as SASR in the sequel. It involves the following basic steps outlined below:

1. Acquire raw RF data frames 1 and N, denoted by *RawData*[1, N].
2. Let *angles* = [1, N], *pivot1* = 1, and *pivot2* = N.
3. [*angles, RawData*] = *SASR* (*angles, pivot1, pivot2, threshold, RawData*).
4. Apply DAS beamforming to acquired raw RF data frames *RawData*[*angles*] to obtain the corresponding beamformed data frames, denoted by *BeamformedData*[*angles*].
5. Perform beamformed frame compounding, envelope detection, and log-compression to obtain the final image.

```
[angles, BeamformedData] =
            SASB-IMMSE (angles, pivot1, pivot2, threshold, BeamformedData):


new_angle = round((pivot1 + pivot2)/2);
if (new_angle does not belong to angles)
      Acquire new raw RF data frame RawData[new_angle];
      BeamformedData[new_angle] = DAS(RawData[new_angle]);
      arg1 = BeamformedData[angles, new_angle];       % new data
      arg2 = BeamformedData[angles];                  % reference data
      L = length(angles);
      angles = [angles, new_angle];
      sim = IMMSE(abs(sum(arg1))/(L+1), abs(sum(arg2))/L);
      if sim > threshold            % larger sim values indicate lesser similarity
            temp1 =
              IMMSE(abs(BeamformedData[new_angle]), abs(BeamformedData[pivot1]));
            temp2 =
              IMMSE(abs(BeamformedData[new_angle]), abs(BeamformedData[pivot2]));
            if temp1 > temp2        % explore left half-interval first
                  [angles, BeamformedData] =
                    SASB-IMMSE(angles, pivot1, new_angle, threshold, BeamformedData);
                  [angles, BeamformedData] =
                    SASB-IMMSE(angles, new_angle, pivot2, threshold, BeamformedData);
            else                    % explore right half-interval first
                  [angles, BeamformedData] =
                    SASB-IMMSE(angles, new_angle, pivot2, threshold, BeamformedData);
                  [angles, BeamformedData] =
                    SASB-IMMSE(angles, pivot1, new_angle, threshold, BeamformedData);
            end
      end
end
```

Figure 3.2(b): Recursive SASB using IMMSE.

As one can see, SASR operates directly on raw RF data frames *RawData*, as opposed to beamformed data frames *BeamformedData* used in SASB as described in section 3.2. The other input and output arguments, such as *angles*, *pivot1*, *pivot2*, and *threshold*, are handled in the same way as in the SASB case. Since our recursive SASR function uses *RawData* instead of *BeamformedData* in step 3, beamforming can be postponed until the selection and acquisition of all raw RF data frames of interest have been completed. Consequently, in step 4, the user is free to employ any desired beamforming method for any acquired raw RF data frame (not just DAS) and to apply pre- and post-beamforming filtering techniques to further enhance the image formation process. In this work, we use DAS beamforming for all acquired raw RF data frames in step 4, so that the performance of SASR can be directly compared to that of SASB.

Given two similarity metric choices, SSIM and IMMSE, our recursive SASR function has two variants presented in sections 3.3.1 and 3.3.2.

3.3.1. SASR Using SSIM (SASR-SSIM)

Figure 3.3(a) shows the pseudo code of our proposed recursive SASR function that uses SSIM as the similarity metric, written as SASR-SSIM.

Note that the logical flow of SASR-SSIM closely imitates that of SASB-SSIM shown in Figure 3.2(a), except that the similarity comparisons are done using *RawData* instead of *BeamformedData*.

```
[angles, RawData] =
            SASR-SSIM(angles, pivot1, pivot2, threshold, RawData):


new_angle = round((pivot1 + pivot2)/2);
if (new_angle does not belong to angles)
      Acquire new raw RF data frame RawData[new_angle];
      arg1 = RawData[angles, new_angle];              % new data
      arg2 = RawData[angles];                         % reference data
      L = length(angles);
      angles = [angles, new_angle];
      sim = SSIM(abs(sum(arg1))/(L+1), abs(sum(arg2))/L);
      if sim < threshold            % smaller sim values indicate lesser similarity
            temp1 = SSIM(abs(RawData[new_angle]), abs(RawData[pivot1]));
            temp2 = SSIM(abs(RawData[new_angle]), abs(RawData[pivot2]));
            if temp1 < temp2        % explore left half-interval first
                  [angles, RawData] =
                    SASR-SSIM(angles, pivot1, new_angle, threshold, RawData);
                  [angles, RawData] =
                    SASR-SSIM(angles, new_angle, pivot2, threshold, RawData);
            else                    % explore right half-interval first
                  [angles, RawData] =
                    SASR-SSIM(angles, new_angle, pivot2, threshold, RawData);
                  [angles, RawData] =
                    SASR-SSIM(angles, pivot1, new_angle, threshold, RawData);
            end
      end
end
```

Figure 3.3(a): Recursive SASR using SSIM.

3.3.2. SASR Using IMMSE (SASR-IMMSE)

Figure 3.3(b) shows the pseudo code of our proposed recursive SASR function that uses IMMSE as the similarity metric, written as SASR-IMMSE.

Note that the logical flow of SASR-IMMSE closely imitates that of SASB-IMMSE shown in Figure 3.2(b), except that the similarity comparisons are done using *RawData* instead of *BeamformedData*.

3.4. Sample Savings

We define the amount of sample savings as follows:

$$Savings = \left(1 - \frac{Number\ of\ \ raw\ RF\ data\ points\ sampled}{Total\ number\ of\ \ raw\ RF\ data\ points}\right) \times 100\%.$$

In the case of SASB and SASR, all data points of each selected raw RF data frame are sampled; therefore, the corresponding savings are determined by the ratio of the number of selected raw RF data frames over N (the total number of available emission angles). However, in the case of CS (see section 2.3), we need to use the general savings formula shown above, as all available raw RF data frames are selected for acquisition, but the number and location of individual data point samples vary from one frame to another.

Clearly, the amount of sample savings depends on the sample selection method (CS, SASB, or SASR), the chosen similarity metric (SSIM or IMMSE), and the threshold value in use. Chapter 4 provides quantitative evaluation results for various configurations in terms of both sample savings and final image quality indicators.

```
[angles, RawData] =
            SASR-IMMSE(angles, pivot1, pivot2, threshold, RawData):


new_angle = round((pivot1 + pivot2)/2);
if (new_angle does not belong to angles)
      Acquire new raw RF data frame RawData[new_angle];
      arg1 = RawData[angles, new_angle];              % new data
      arg2 = RawData[angles];                         % reference data
      L = length(angles);
      angles = [angles, new_angle];
      sim = IMMSE(abs(sum(arg1))/(L+1), abs(sum(arg2))/L);
      if sim > threshold            % larger sim values indicate lesser similarity
            temp1 = IMMSE(abs(RawData[new_angle]), abs(RawData[pivot1]));
            temp2 = IMMSE(abs(RawData[new_angle]), abs(RawData[pivot2]));
            if temp1 > temp2        % explore left half-interval first
                  [angles, RawData] =
                    SASR-IMMSE(angles, pivot1, new_angle, threshold, RawData);
                  [angles, RawData] =
                    SASR-IMMSE(angles, new_angle, pivot2, threshold, RawData);
            else                    % explore right half-interval first
                  [angles, RawData] =
                    SASR-IMMSE(angles, new_angle, pivot2, threshold, RawData);
                  [angles, RawData] =
                    SASR-IMMSE(angles, pivot1, new_angle, threshold, RawData);
            end
      end
end
```

Figure 3.3(b): Recursive SASR using IMMSE.

Chapter 4: Evaluation Results

4.1 PICMUS Evaluation Setup

PICMUS (Plane-wave Imaging Challenge in Medical UltraSound) was a part of the 2016 IEEE International Ultrasonics Symposium [16]. Two experimental datasets for seven point phantoms and two cyst phantoms were acquired using the *Verasonics Vantage 256* ultrasound research scanner and the L11 linear array probe (Verasonics Inc., Redmond, WA) [16]. The datasets were recorded on a *CIRS Multi-Purpose Ultrasound Phantom* (Model 040GSE) in the regions shown in figure 4.1.

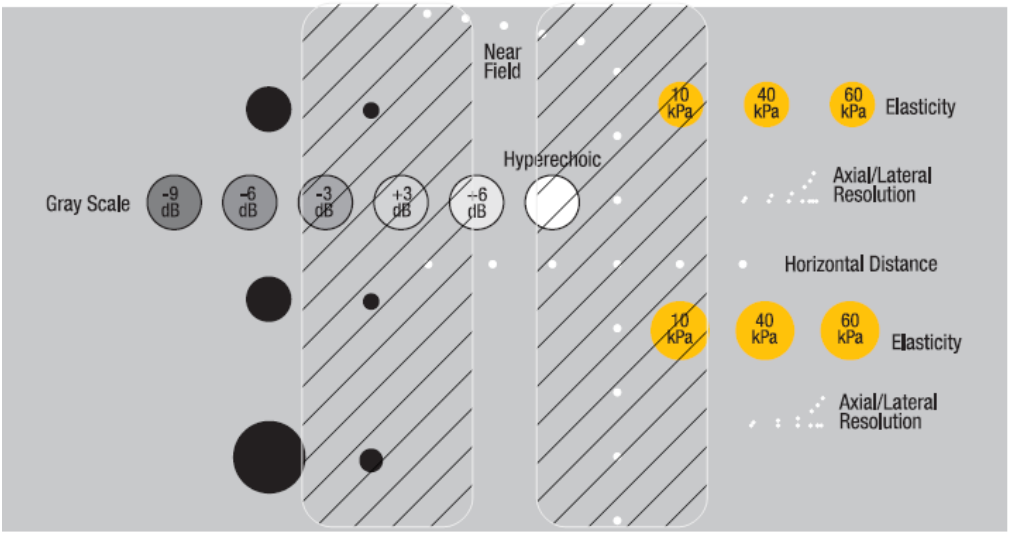

Figure 4.1: Schematic of the upper part of the CIRS Model 040GSE Phantom used to collect the experimental data. The hatched left region was acquired for contrast evaluation, while the right region was acquired for resolution evaluation.

4.1.1 Description of Datasets

Each PICMUS dataset consists of N = 75 raw RF data frames. Each 2D frame represents a T×M matrix of raw RF data samples acquired for a particular emission angle, where T = 3328 (the number to sampling time instances) and M = 128 (the number of transducer elements). The angles range from θ_1 = –16° to θ_N = +16° with an increment of 0.432°. Table 4.1 lists the other parameters used during ultrasound data acquisition [16].

Table 4.1 Description of PICMUS data acquisition parameters.

Element width	0.27 mm
Element height	5 mm
Elevation focus	20 mm
Pitch	0.30 mm
Aperture width	38.4 mm
Sampling frequency	20.832 MHz
Transmit frequency	5.20 MHz
Pulse bandwidth	67%
Excitation	2.5 cycles

When all 75 raw RF data frames are acquired, beamformed, and compounded, we obtain a *Full Acquisition* (FA) image, which will serve as our best-case reference. The recorded datasets were obtained by imaging the two regions of a CIRS Multi-Purpose Ultrasound Phantom (Model 040GSE) as shown in figure 4.1. Specifically, we will be interested in

the seven point phantoms labeled A-G in figure 4.2(a) and the two cyst phantoms labeled X and Y in figure 4.2(b). The FA case will be discussed further in section 4.2.

4.1.2 Description of Metrics

Following the PICMUS evaluation criteria, we assess the image quality as follows. The full width at half maximum (FWHM) in both axial and lateral directions, measured for individual point targets A-G, shown in figure 4.2(a), will be the resolution quality indicator: lower FWHM values mean better image resolution.

The contrast-to-noise ratio (CNR), measured for both X and Y cyst targets, shown in figure 4.2(b), will be the contrast quality indicator: higher CNR values mean better image contrast. The CNR is given by

$$CNR = 20\log_{10}\left(\frac{|\mu_{in} - \mu_{out}|}{\sqrt{((\sigma_{in}^2 + \sigma_{out}^2)/2)}}\right), \tag{4.1}$$

where μ_{in} is the mean gray level inside the anechoic cystic region, μ_{out} is the mean gray level outside the anechoic cystic region, σ_{in} is the gray level standard deviation inside the anechoic cystic region, and σ_{out} is the gray level standard deviation outside the anechoic cystic region.

We are also interested in the speckle background, which carries important information useful for tissue classification, structure segmentation, motion estimation, etc. Each of the three predefined regions S1-S3, shown figure 4.2(c), is subjected to the Kolmogorov-Smirnov (KS) test to verify whether there is enough evidence in the data to allude that the hypothesis under consideration (that is, the data follows a Rayleigh distribution) is true.

The regions that pass the KS test are considered to have the speckle quality preserved, which is the desired outcome [49].

4.1.3 Evaluated Techniques

Sampling choices for raw data have a huge impact on how FWHM, CNR and speckle tests perform. In the sequel, we evaluate the following sampling scenarios:

- Compressive sensing (CS) described in section 2.4,
- Similarity-driven angle selection using beamformed data and SSIM (SASB-SSIM) described in section described in section 3.2.1,
- Similarity-driven angle selection using beamformed data and IMMSE (SASB-IMMSE) described in section 3.2.2,
- Similarity-driven angle selection using raw RF data and SSIM (SASR-SSIM) described in section 3.3.1,
- Similarity-driven angle selection using raw RF data and IMMSE (SASR-IMMSE) described in section 3.3.2.

For each sampling method under consideration, its acquired raw RF data undergoes DAS beamforming using S = M = 128 (the number of scan lines), followed by compounding and envelope detection. The envelope data is then normalized and log-compressed, which produces the final image data in dB units. The dynamic range of all images is 60 dB. We compare these images to our FA reference images in terms of FWHM, CNR, speckle test passage, and execution times.

4.2 Full Acquisition (FA) and Compressive Sensing (CS)

In this section, we shall explore the case of FA and compare it with CS using FWHM and CNR values, and whether they pass the speckle test or not. Figures 4.2 and 4.3 show the FA and CS images, respectively. In the CS pseudo code shown in figure 2.3(b), we have used the compression factor *cf* = 4, which results in 75% savings. The padding sub-vector contained 40 zeros, and the error tolerance was set at *tol* = 1E-13.

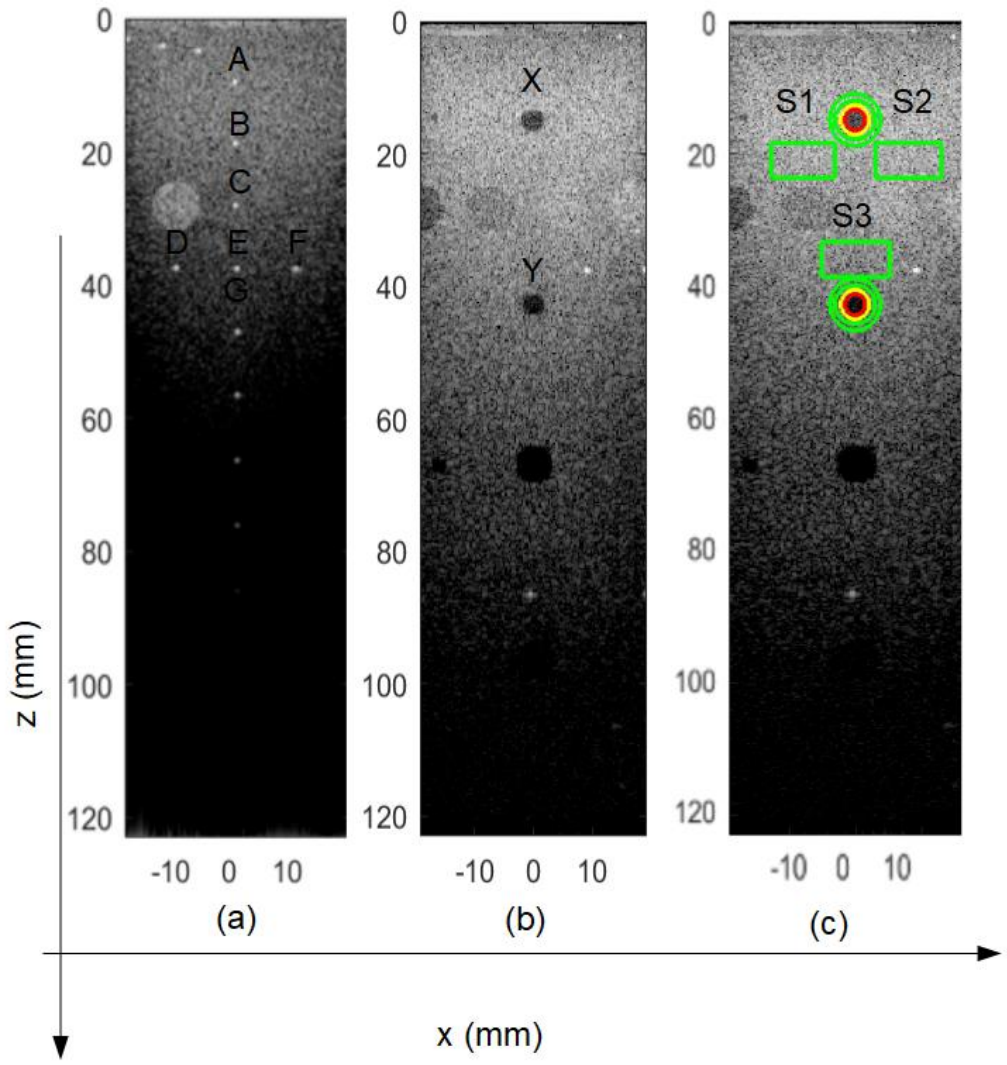

Figure 4.2: Reference FA images – point phantoms A-G (a), cyst phantoms X and Y (b), speckle regions S1-S3 (c).

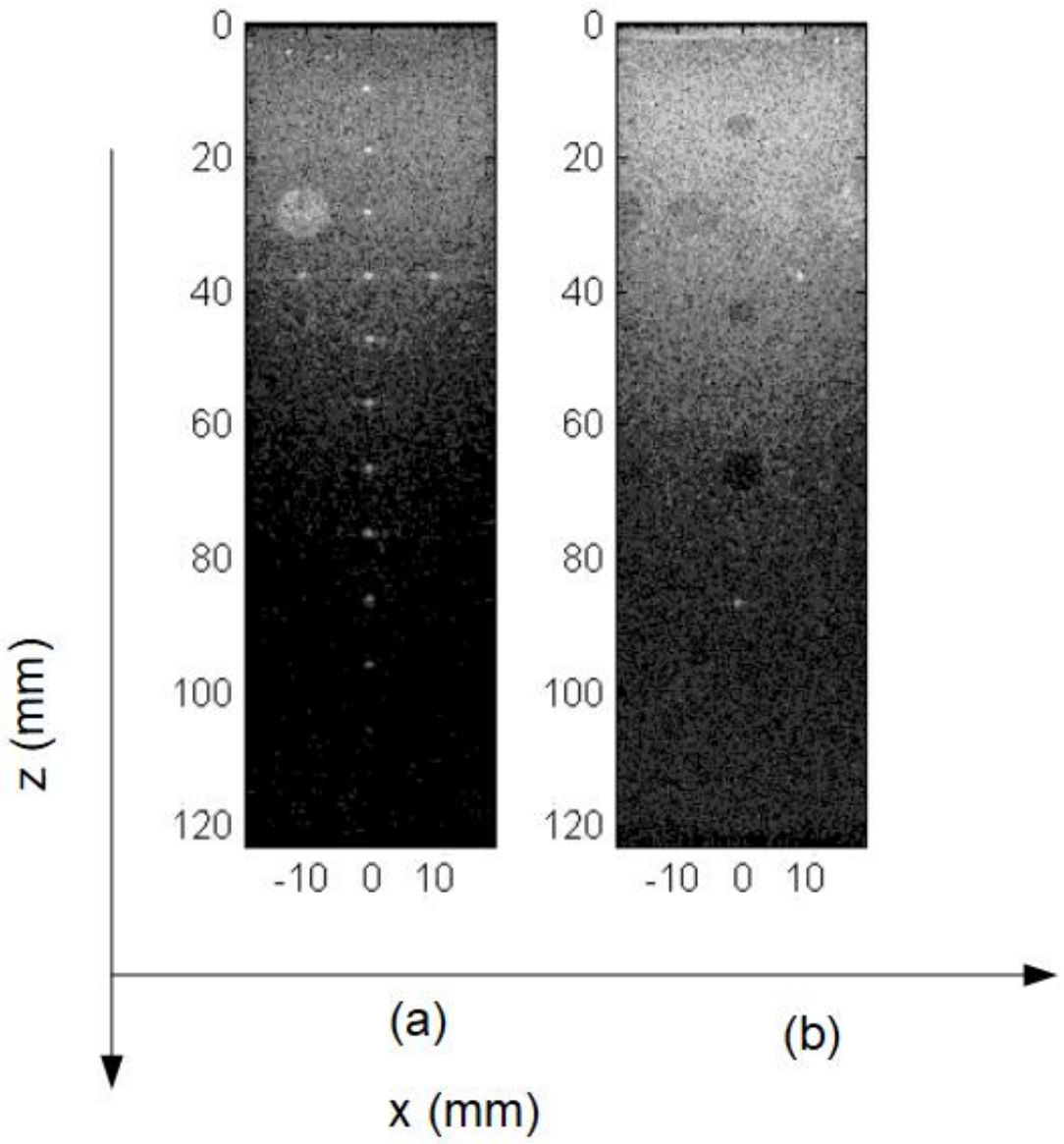

Figure 4.3: Example CS images (75% savings) – point phantoms (a), cyst phantoms (b).

Table 4.2 shows the corresponding FWHM values for both FA and CS images. The average lateral and axial FWHM values in the FA case are 0.54 mm and 0.57 mm, respectively. Despite having acquired only 25% of raw RF data samples, CS performed equally well, achieving the same average FWHM values as those in the FA case.

Table 4.2: Comparative FWHM values for FA and CS images.

Acquisition Scenario	Lateral/Axial FWHM (mm)							
	A	*B*	*C*	*D*	*E*	*F*	*G*	Average
FA	0.45/0.56	0.54/0.54	0.53/0.54	0.55/0.57	0.52/0.58	0.67/0.59	0.52/0.59	0.54/0.57
CS	0.53/0.50	0.48/0.49	0.52/0.43	0.59/0.53	0.58/0.68	0.51/0.73	0.57/0.60	0.54/0.57

Table 4.3: Comparative CNR values and speckle test results for FA and CS images.

Acquisition Scenario	CNR (dB)		Speckle Test		
	X	*Y*	*S1*	*S2*	*S3*
FA	11.98	11.46	Pass	Pass	Pass
CS	9.18	8.40	Pass	Pass	Pass

Table 4.3 shows the corresponding CNR values and speckle test results for both FA and CS images. The FA image offers the best CNR values for both near-field X and far-field Y cysts, equal to 11.98 dB and 11.45 dB, respectively. The respective CNR values in the CS case are 9.18 dB (23% worse) and 8.40 dB (27% worse); however, it is important to emphasize that the amount of sampling savings is as much as 75%.

Note that our CS technique adds zero padding on both sides of a signal vector prior to random subsampling and conversion into the frequency domain. Usually zero padding is added at the end of a signal vector, which increases its DFT length, implying better frequency resolution (i.e., frequency bins become narrower). This approach did not work

in our case. Moving the prepended zero-valued subvector *pad* to the end of $\boldsymbol{x}$ (see figure 2.3(b)), i.e., changing the signal from [*pad* $\boldsymbol{x}$ *pad*] to [$\boldsymbol{x}$ *pad pad*] while keeping the other settings the same, resulted in severe image degradation: the average lateral and axial FWHM values became significantly larger (1.38 mm and 1.10 mm), the CNR values for X and Y became negative, and the speckle tests failed.

4.3 SASB Using SSIM (SASB-SSIM)

In this section, we examine the imaging performance of the SASB-SSIM acquisition method, whose pseudo code is shown in figure 3.2(a). Tables 4.4 and 4.5 list the corresponding FWHM and CNR values, with the speckle test results, for several SSIM threshold settings. The thresholds range from 0.850 to 0.885, translating into the sample savings from 93% to 76%. As an illustrative example, figure 4.4 shows the final images obtained using the SSIM threshold of 0.875.

Table 4.4: SASB-SSIM image resolution quality.

Threshold Setting	Sample Savings	Lateral/Axial FWHM (mm)							
		A	*B*	*C*	*D*	*E*	*F*	*G*	Average
0.85	93%	0.53/0.34	0.53/0.40	0.57/0.45	0.57/0.40	0.60/0.61	0.61/0.62	0.60/0.48	0.57/0.47
0.875	88%	0.54/0.46	0.53/0.54	0.57/0.64	0.57/0.53	0.60/0.71	0.59/0.75	0.60/0.65	0.57/0.61
0.8775	80%	0.55/0.50	0.53/0.55	0.58/0.72	0.57/0.57	0.60/0.71	0.59/0.77	0.59/0.63	0.57/0.64
0.885	76%	0.55/0.53	0.53/0.58	0.57/0.77	0.57/0.58	0.59/0.73	0.58/0.84	0.59/0.67	0.57/0.67

Table 4.5: SASB-SSIM image contrast and speckle quality.

Threshold Setting	Sample Savings	CNR (dB)		Speckle Test		
		X	*Y*	*S1*	*S2*	*S3*
0.85	93%	3.25	4.59	Pass	Pass	Pass
0.875	88%	5.02	6.31	Pass	Pass	Pass
0.8775	80%	6.13	7.58	Pass	Pass	Pass
0.885	80%	6.13	7.58	Pass	Pass	Pass

Table 4.4 shows that increasing the SSIM threshold from 0.850 to 0.885 does not affect the average lateral FWHM value of 0.57 mm, which is 5.6% worse than in the FA case. On the other hand, the axial FWHM values range from 0.47 mm (17.5% better than FA) to 0.67 mm (17.5% worse than FA). As expected, the amount of sample savings is greater at lower thresholds, because relaxing the similarity requirement on the beamformed data lets the binary search terminate earlier.

Table 4.5 shows that the CNR values increase as the threshold is increased from 0.850 to 0.8775. Increasing the latter value to 0.885 has not led to any additional raw data frames being acquired (the amount of sample savings remained at 80%), which prevented further potential improvement of the image contrast quality. The highest CNR values in table 4.5 are 6.13 dB for X and 7.58 dB for Y, which are substantially worse than those in the FA case (11.98 dB and 11.46 dB, respectively).

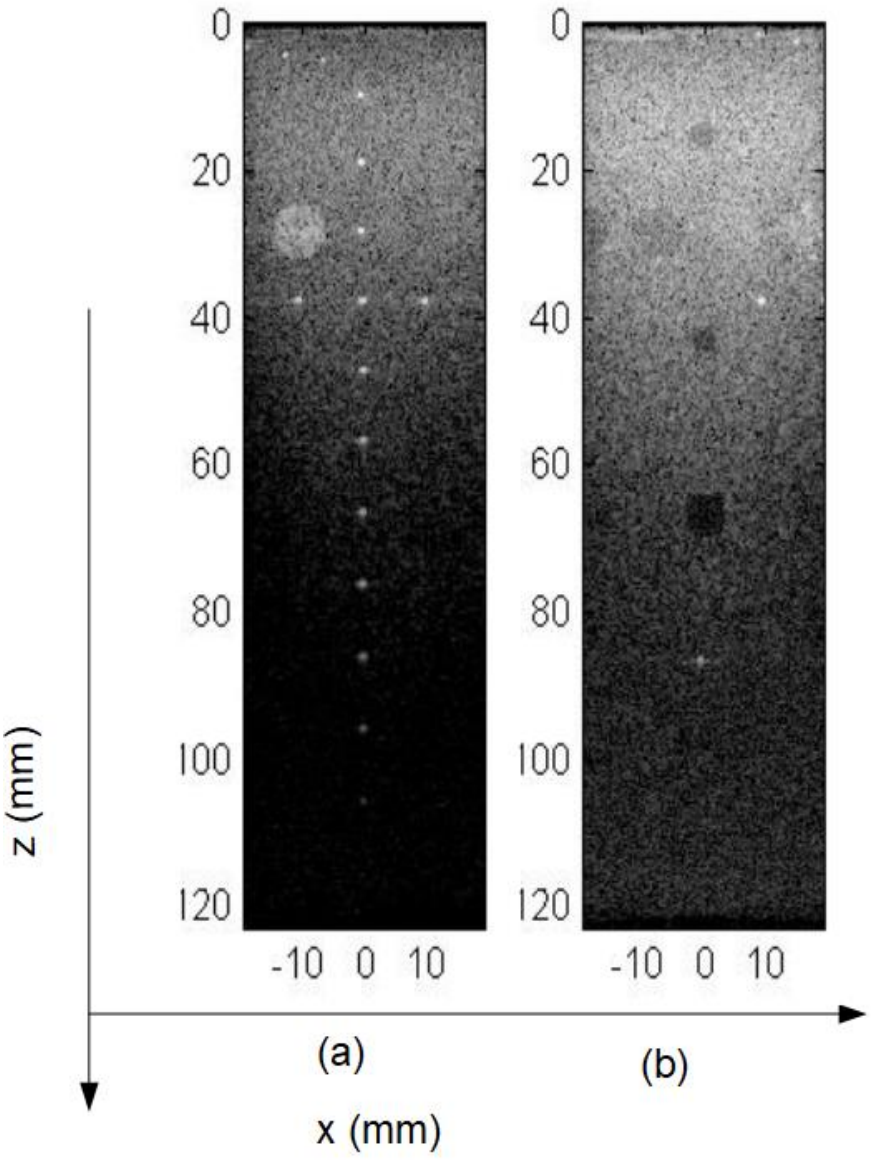

Figure 4.4: Example SASB-SSIM images (0.875 threshold, 88% savings) – point phantoms (a), cyst phantoms (b).

4.4 SASB Using IMMSE (SASB-IMMSE)

In this section, we examine the imaging performance of the SASB-IMMSE acquisition method, whose pseudo code is shown in figure 3.2(b). Tables 4.6 and 4.7 list the corresponding FWHM and CNR values, with the speckle test results, for several IMMSE threshold settings. The thresholds range from 3.00E-4 to 1.75E-4, translating into the sample savings from 93% to 75%. As an illustrative example, figure 4.5 shows the final images obtained using the IMMSE threshold of 1.75E-4.

Table 4.6: SASB-IMMSE image resolution quality.

Threshold Setting	Sample Savings	Lateral/Axial FWHM (mm)							
		A	*B*	*C*	*D*	*E*	*F*	*G*	Average
3E-4	93%	0.53/0.34	0.53/0.40	0.57/0.45	0.57/0.40	0.60/0.61	0.61/0.62	0.60/0.48	0.57/0.47
2E-4	88%	0.54/0.40	0.53/0.50	0.57/0.60	0.57/0.52	0.61/0.66	0.60/0.72	0.60/0.58	0.57/0.57
1.85E-4	80%	0.55/0.50	0.53/0.55	0.58/0.72	0.57/0.57	0.60/0.71	0.59/0.77	0.59/0.63	0.57/0.64
1.75E-4	77%	0.55/0.53	0.53/0.57	0.57/0.76	0.57/0.59	0.59/0.73	0.58/85	0.59/0.67	0.57/0.67

In table 4.6, the average lateral FWHM value of 0.57 mm (5.6% worse than FA) does not change as the IMMSE threshold is decreased. Similar to SASB-SSIM (see table 4.4), the axial FWHM values deviate around ±17.5% with respect to the FA case (0.57 mm), while the amount of sample savings ranges from 77% to 93%.

In table 4.7, the CNR values get better as the IMMSE threshold is decreased, but they plateau temporarily at 5.02 dB and 6.31 dB (for X and Y, respectively) when the threshold changes from 2E-4 to 1.85E-4. Since the amount of sample savings, equal to

88%, is the same for both thresholds, it means that no additional raw data frames have been acquired when the similarity requirement became more stringent, changing from IMMSE ≤ 2E-4 to IMMSE ≤ 1.85E-4. The best CNR values in table 4.5 are 9.34 dB for X and 8.00 dB for Y (22.0% and 30.2% worse than in the FA case). They correspond to the IMMSE threshold of 1.75E-5, which results in 75% sample savings. Note that these values are very close to those obtained by CS from table 4.3 (9.16 dB and 8.40 dB, respectively).

Table 4.7: SASB-IMMSE image contrast and speckle quality.

Threshold Setting	Sample Savings	CNR (dB)		Speckle Test		
		X	*Y*	*S1*	*S2*	*S3*
3E-4	93%	3.25	4.59	Pass	Pass	Pass
2E-4	88%	5.02	6.31	Pass	Pass	Pass
1.85E-4	88%	5.02	6.31	Pass	Pass	Pass
1.75E-4	75%	9.34	8.00	Pass	Pass	Pass

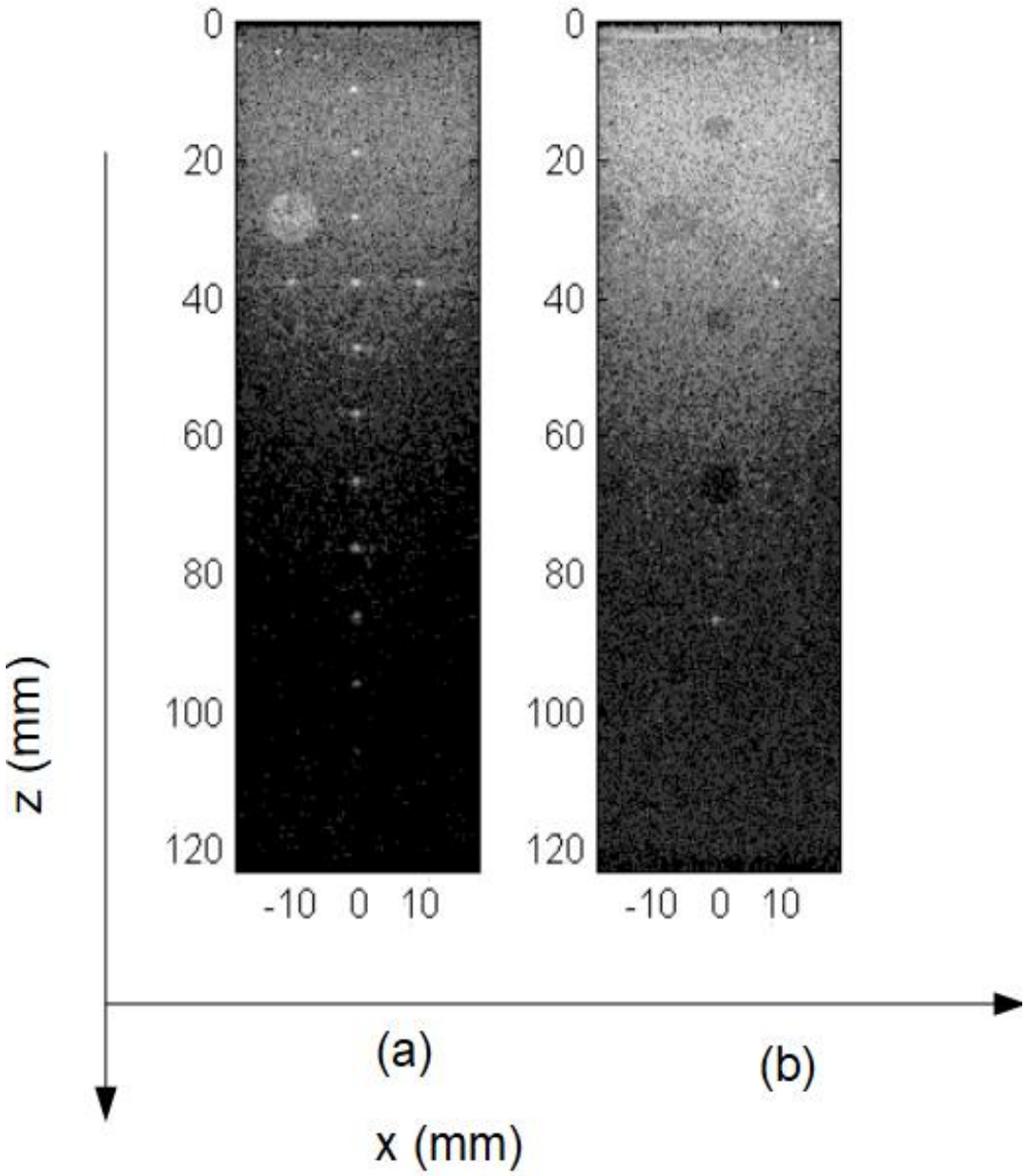

Figure 4.5: Example SASB-IMMSE images (1.75E-4 threshold, 75-77% savings) – point phantoms (a), cyst phantoms (b).

4.5 SASB with Region-of-Interest (ROI) Restrictions

Another useful application of SASB (using SSIM or IMMSE) is ROI-driven acquisition. Given the location of an imaging region of interest, SASB can be used to focus on just a cyst, a tumor, or an anomaly in an ultrasound image in question. Recall that step 2 of our SASB scheme (described at the beginning of section 3.2) computes two beamformed frames *BeamformedData*[1, N] that corresponds to the boundary emission angles θ_1 and θ_N. One can immediately compound them, form a preliminary low-quality image, and then identify a desired ROI (in that image) whose quality needs to be improved. All subsequent sample acquisitions and similarity measurements can now be restricted to that particular ROI. Image quality improvements produced as a result and the amount of sample savings achieved will depend on the ROI size and the threshold choice.

Figure 4.6 illustrates ROI-driven SASB-SSIM acquisition using the threshold of 0.8775, where the imaging ROI is the section spanning from 18 mm to 40 mm in depth. Such ROI implies that we need to acquire only rows 500-1080 of every subsequently selected 3328×128 raw data frame, which immediately yields 3328/581 = 5.7-fold reduction in the number of acquired samples. Similarly, figure 4.7 illustrates ROI-driven SASB-IMMSE using the same ROI and the threshold of 6.00E-6. The total amount of sample savings is 85% in the SASB-SSIM case and 93% in the SASB-IMMSE case.

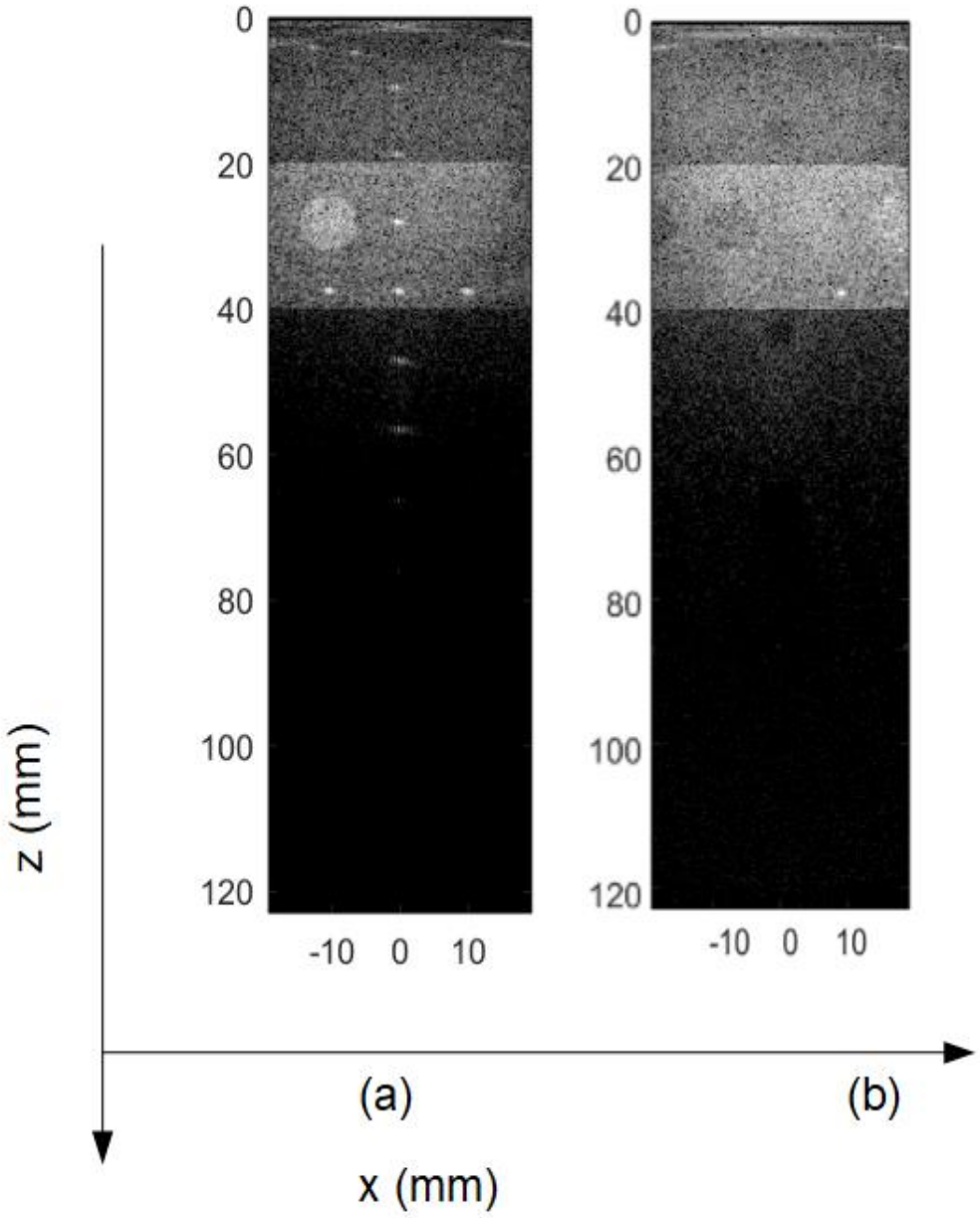

Figure 4.6: Example SASB-SSIM images (0.8775 threshold, 85% savings) – point phantoms (a), cyst phantoms (b). ROI: depth range from 18 to 40 mm.

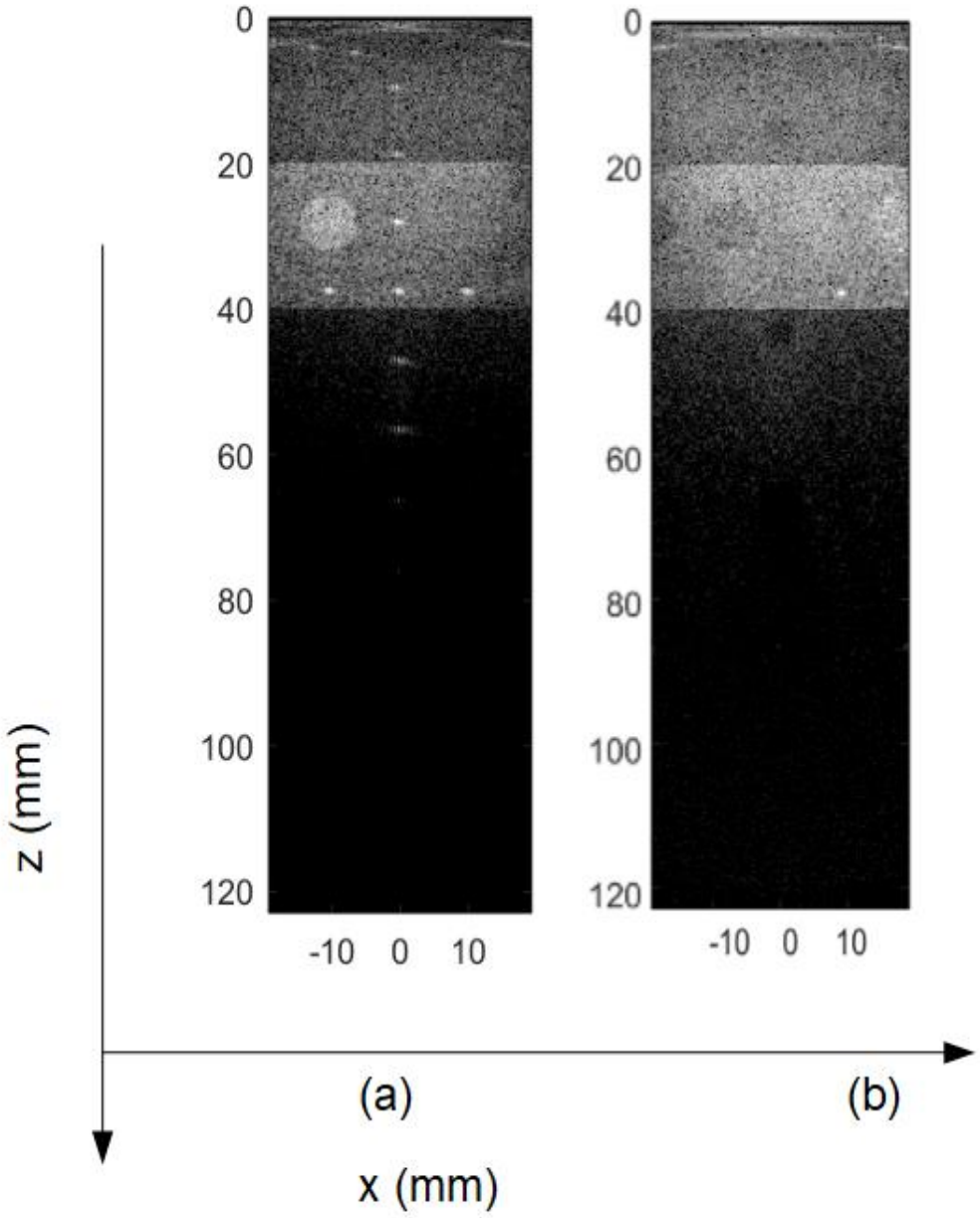

Figure 4.7: Example SASB-IMMSE images (6E-6 threshold, 93% savings) – point phantoms (a), cyst phantoms (b). ROI: depth range from 18 to 40 mm.

4.6 SASR Using SSIM (SASR-SSIM)

In this section, we examine the imaging performance of the SASR-SSIM acquisition method, whose pseudo code is shown in figure 3.3(a). Tables 4.8 and 4.9 list the corresponding FWHM and CNR values, with the speckle test results, for several SSIM threshold settings. The thresholds range from 0.995 to 0.9995, translating into the sample savings from 88% to 76%. As an illustrative example, figure 4.8 shows the final images obtained using the SSIM threshold of 0.9995.

Table 4.8: SASR-SSIM image resolution quality.

Threshold Setting	Sample Savings	Lateral/Axial FWHM (mm)							
		A	*B*	*C*	*D*	*E*	*F*	*G*	Average
0.995	88%	0.54/0.44	0.54/0.57	0.57/0.63	0.57/0.67	0.59/0.69	0.61/0.72	0.59/0.57	0.57/0.61
0.9975	85%	0.56/0.46	0.53/0.56	0.59/0.62	0.58/0.62	0.59/0.65	0.63/0.69	0.61/0.56	0.58/0.59
0.9985	81%	0.55/0.49	0.53/0.61	0.60/0.66	0.57/0.66	0.60/0.71	0.64/0.73	0.61/0.60	0.59/0.64
0.9995	76%	0.56/0.51	0.54/0.67	0.59/0.72	0.55/0.71	0.60/0.81	0.63/0.78	0.61/0.64	0.58/0.69

As can be seen in table 4.8, the average lateral and axial FWHM values vary with the change in the SSIM threshold. The former ranges from 0.57 mm to 0.59 mm, which is 5.6% and 9.3% worse than in the FA case, respectively; whereas, the latter ranges from 0.59 mm to 0.69 mm, which is 3.5% and 21.1% worse than in the FA case, respectively.

Table 4.9: SASR-SSIM image contrast and speckle quality.

Threshold Setting	Sample Savings	CNR (dB)		Speckle Test		
		X	*Y*	*S1*	*S2*	*S3*
0.995	88%	5.63	6.65	Pass	Pass	Pass
0.9975	85%	7.32	8.20	Pass	Pass	Pass
0.9985	81%	8.12	8.69	Pass	Pass	Pass
0.9995	76%	8.80	8.95	Pass	Pass	Pass

Table 4.9 shows that the CNR values become better as the SSIM threshold is increased, which also leads to reduced savings. The threshold of 0.9995 has resulted in the CNR values of 8.80 dB for X and 8.95 dB for Y. While they respectively are 26.5% and 21.9% worse than those achieved by FA acquisition, they have been obtained using only 24% of raw data samples.

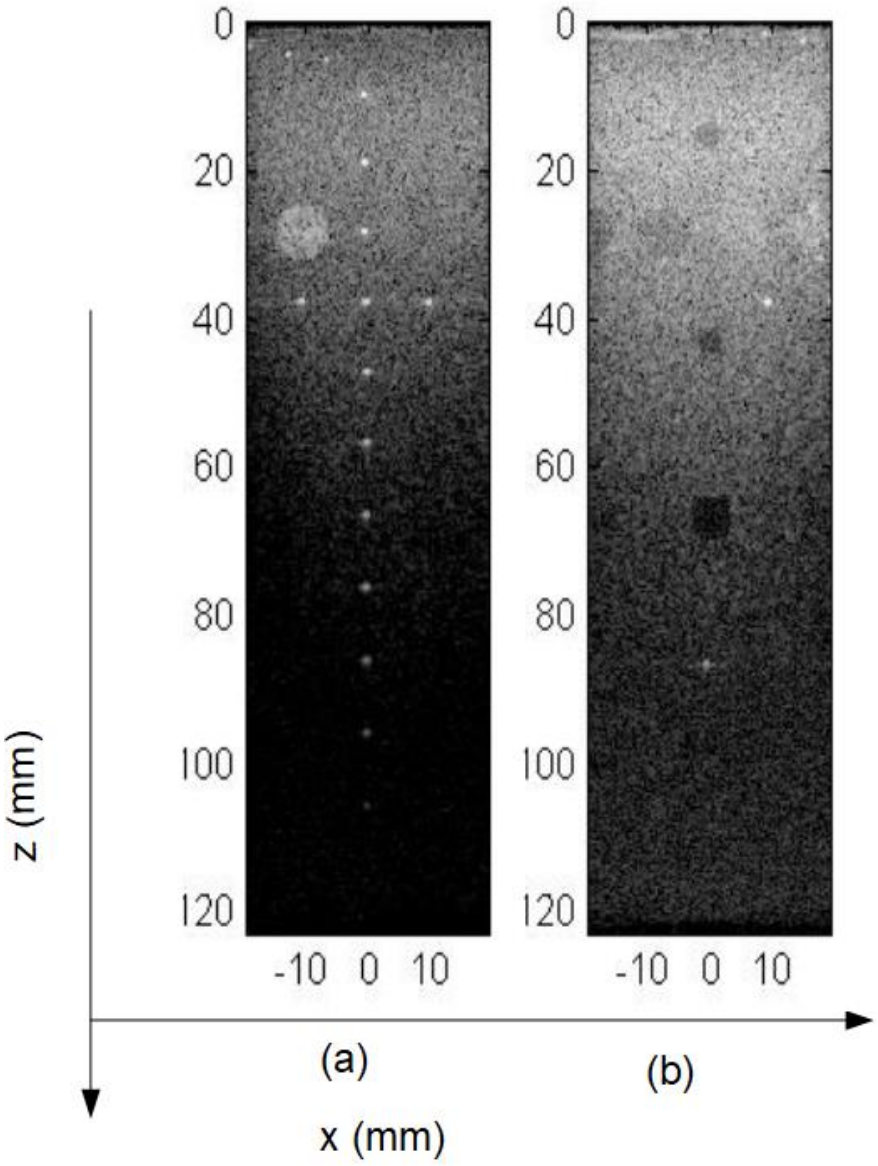

Figure 4.8: Example SASR-SSIM images (0.9995 threshold, 76% savings) – point phantoms (a), cyst phantoms (b).

4.7 SASR Using IMMSE (SASR-IMMSE)

In this section, we examine the imaging performance of the SASB-IMMSE acquisition method, whose pseudo code is shown in figure 3.3(b). Tables 4.10 and 4.11 list the

corresponding FWHM and CNR values, with the speckle test results, for several IMMSE threshold settings. The thresholds range from 9E-6 to 2E-4, translating into the sample savings from 90% to 80%. As an illustrative example, figure 4.9 shows the final images obtained using the IMMSE threshold of 9E-6.

Table 4.10: SASR-IMMSE image resolution quality.

Threshold Setting	Sample Savings	Lateral/Axial FWHM (mm)							
		A	*B*	*C*	*D*	*E*	*F*	*G*	Average
9E-6	90%	0.54/0.52	0.54/0.60	0.58/0.68	0.57/0.57	0.60/0.70	0.61/0.79	0.60/0.64	0.58/0.64
5.5E-6	85%	0.56/0.54	0.53/0.57	0.58/0.72	0.57/0.56	0.59/0.68	0.60/0.75	0.59/0.64	0.57/0.64
4E-6	84%	0.55/0.55	0.53/0.57	0.57/0.73	0.57/0.56	0.59/0.68	0.60/0.74	0.59/0.64	0.57/0.64
2E-6	80%	0.56/0.60	0.53/0.61	0.57/0.80	0.57/0.58	0.59/0.72	0.59/0.81	0.60/0.67	0.57/0.68

According to table 4.10, the average lateral and axial FWHM values have not improved after lowering the IMMSE threshold below 5.5E-6. The corresponding values of 0.57 mm and 0.64 mm (5.2% and 12.2% worse than in the FA case) are the same as those achieved in the SASB-IMMSE case using the IMMSE threshold of 1.85E-4; however, the amount of sample savings is higher in the SASR-IMMSE case (85% vs. 80%).

Table 4.11: SASR-IMMSE image contrast and speckle quality.

Threshold Setting	Sample Savings	CNR (dB)		Speckle Test		
		X	*Y*	*S1*	*S2*	*S3*
9E-6	90%	5.23	7.14	Pass	Pass	Pass
5.5E-6	85%	8.34	7.95	Pass	Pass	Pass
4E-6	85%	8.34	7.95	Pass	Pass	Pass
2E-6	80%	9.20	8.13	Pass	Pass	Pass

Among the CNR values listed in table 4.11, the best ones are 9.20 dB for X and 8.13 dB for Y, achieved using the threshold of 2E-6. These values are 23.2% and 29.1% worse than in the FA case, but they are also close to 9.18 dB and 8.40 dB obtained in the CS case, while offering more savings (80% vs 75%). It should be noted that tightening the IMMSE threshold from 5.5E-6 to 4E-6 has not resulted in any additional raw data frames being acquired, thus preventing any improvement in the CNR values.

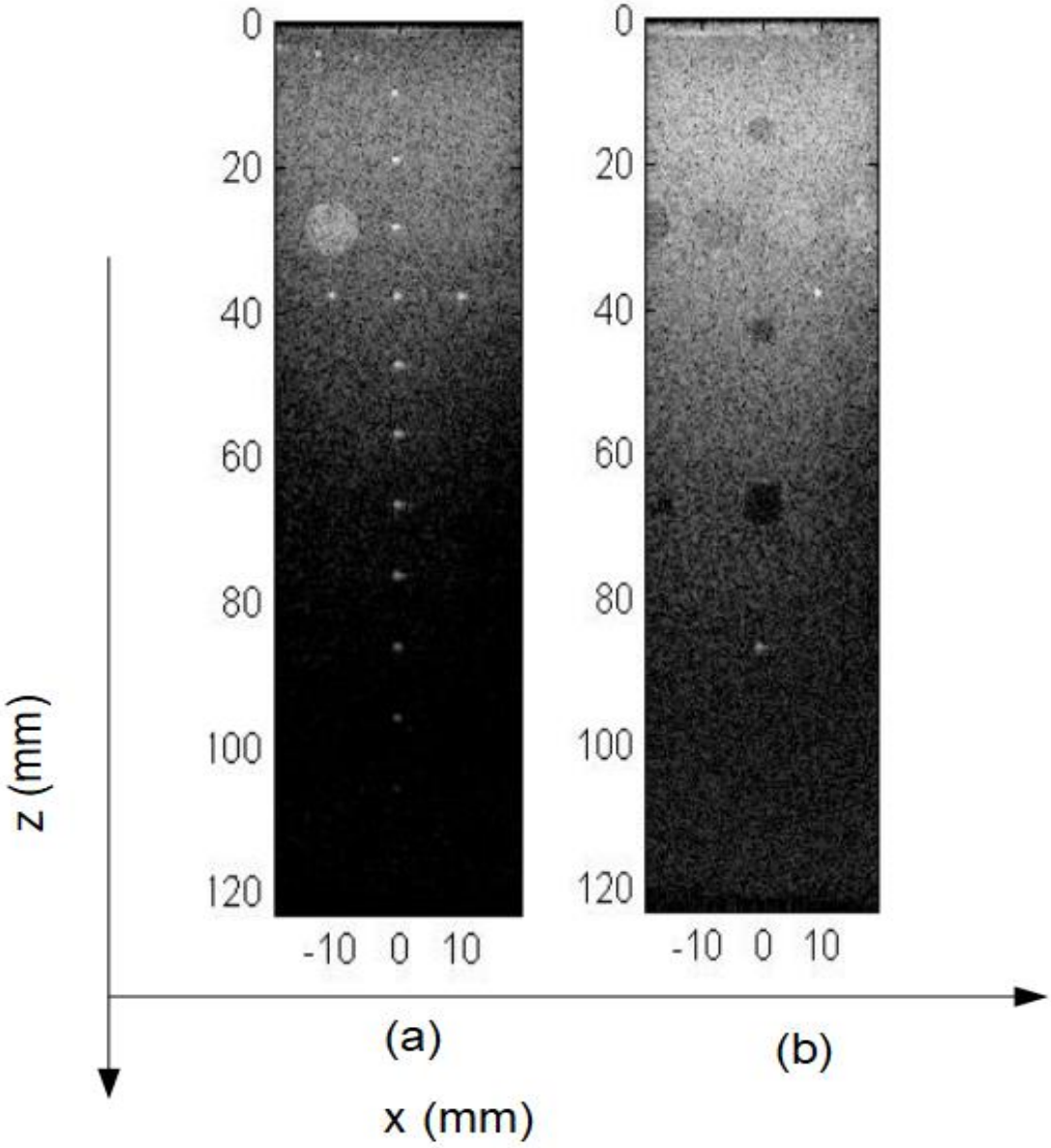

Figure 4.9: Example SASR-IMMSE images (9E-6 threshold, 90% savings) – point phantoms (a), cyst phantoms (b).

4.8 Uniform Angle Selection (UAS)

As the name suggests, Uniform Angle Selection (UAS) involves uniform sampling of the entire range of available angles. The comparison of SASB and SASR with UAS will allow us to evaluate how well our recursive angle-selection schemes perform against the much simpler method targeting the same amount of sample savings.

Table 4.12 shows the FWHM values obtained by UAS that uniformly samples Q out of N = 75 available raw data frames, so that (1 – (Q/75)) × 100% is equal to the targeted amount of sample savings: 90%, 88%, 85%, 82%, 80%, and 75%. As one can see, the best average lateral FHWM value is 0.57 mm (5.6% worse than in the FA case) at 90% savings, and the best average lateral FHWM value is 0.49 mm (14.0% better than in the FA case) at 88% savings. In comparison, our SASB-IMMSE method achieved the average lateral and axial FWHM values of 0.57 mm and 0.47 mm, respectively, using the threshold of 3E-4 at 93% savings (see table 4.6).

Table 4.13 shows the CNR values and the speckle test results for the same targeted savings as in table 4.12. As one can see, the best CNR value for X is 6.80 dB (43.2% worse than in the FA case) at 88% savings, and the best CNR value for Y is 8.49 dB (25.9% worse than in the FA case) at 80% savings. In comparison, our SASR-IMMSE method achieved the respective CNR values of 9.20 dB and 8.13 dB, using the threshold of 3E-6 at 80% savings (see table 4.11).

As an illustrative example, figure 4.10 shows the images formed from raw data obtained by UAS targeting 88% savings.

Table 4.12: UAS image resolution quality.

Targeted Savings	Lateral/Axial FWHM (mm)							
	A	*B*	*C*	*D*	*E*	*F*	*G*	Average
90%	0.53/0.40	0.50/0.47	0.56/0.52	0.57/0.49	0.61/0.56	0.59/0.68	0.60/0.48	0.57/0.51
88%	0.55/0.40	0.53/0.44	0.58/0.50	0.58/0.49	0.60/0.54	0.59/0.63	0.60/0.49	0.58/0.49
85%	0.55/0.45	0.54/0.47	0.55/0.52	0.57/0.50	0.60/0.54	0.59/0.68	0.60/0.48	0.57/0.52
82%	0.54/0.39	0.53/0.47	0.59/0.53	0.58/0.50	0.59/0.57	0.58/0.66	0.60/0.49	0.57/0.51
80%	0.53/0.42	0.52/0.48	0.58/0.53	0.58/0.51	0.61/0.59	0.59/0.69	0.60/0.50	0.57/0.53
75%	0.55/0.42	0.51/0.49	0.58/0.53	0.59/0.50	0.58/0.58	0.60/0.71	0.61/0.51	0.57/0.53

Table 4.13: UAS image contrast and speckle quality.

Targeted Savings	CNR (dB)		Speckle Test		
	X	*Y*	*S1*	*S2*	*S3*
90%	5.14	7.24	Pass	Pass	Pass
88%	6.80	6.88	Pass	Pass	Pass
85%	6.63	7.91	Pass	Pass	Pass
82%	7.97	8.14	Pass	Pass	Pass
80%	6.74	8.49	Pass	Pass	Pass
75%	6.46	7.27	Pass	Pass	Pass

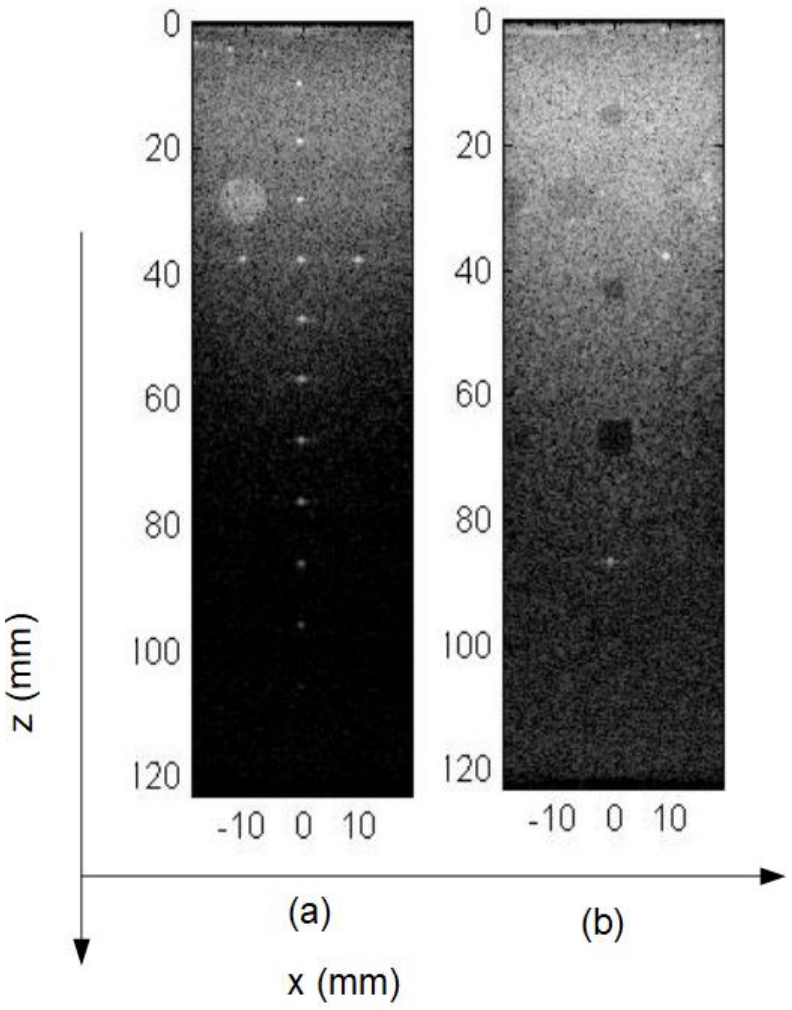

Figure 4.10: Example UAS images (88% savings) – point phantoms (a), cyst phantoms (b).

4.9 Computational Complexity Analysis

The plane-wave imaging modality used in this thesis can be broken down into three major computational parts: a) SAS/CS raw data selection/reconstruction techniques, b) DAS beamforming of raw data frames, and c) coherent compounding of beamformed data frames. In this section we briefly discuss computational complexity of each of these workload components.

Figure 2.2 shows the DAS beamforming pseudo code, which involves four variables: the plane-wave emission angle θ, the sensor lateral position x_s, and the beamformed point coordinates z and x_p (axial and lateral, respectively). As explained in section 2.5, for each beamformed frame, the number of time delay calculations is M×S×T, where M is the number of sensors (i.e., the number of processed x_s values), S is the number of image scanlines (i.e., the number of processed x_p values), and T is the number of acquired input vector snapshots (i.e., the number of processed t values, which is the same as the number of processed z values). Therefore, the computational cost of time delays for n frames is bounded by $O(n \times M \times S \times T)$. In this thesis, we let the number of scanlines be equal to the number of sensors (i.e., $M = S$), and the computational cost of DAS beamforming can be expressed as $O(n \times M^2 \times T)$. Thus, as we increase n (the number of selected angles for acquisition), while keeping the number of sensors and the number of input snapshots constant, the computational cost of DAS beamforming will grow linearly.

Our proposed SAS scheme involves IMMSE/SSIM calculations, as opposed to the UAS scheme that has no computational overhead in terms of data-dependent decision making. Both SSIM and IMMSE are computed using a pair of M×T data matrices in our case, and their computational complexity is bounded by $O(M \times T)$ per acquired frame, or $O(n \times M \times T)$ over all n frames. According to figures 3.2 and 3.3, our selection scheme also involves summing up as many as n data frames, which also takes $O(n \times M \times T)$ time. Hence, the computational complexity of our SASB and SASR methods is $O(n \times M \times T)$, which is lower than that of DAS beamforming.

Coherent compounding is the next step after DAS beamforming, which involves adding n beamformed M×T data frames together. Therefore, the corresponding computational

cost is bounded by $O(n \times M \times T)$. If M and T are kept constant, the computational complexity of coherent compounding grows linearly with respect to n. As in the case of our proposed SAS scheme, coherent compounding is also superseded by DAS beamforming in terms of the amount of computation involved.

Our CS application technique relies on solving the l_1-minimization problem under equality constraints using a primal-dual interior-point (PDIP) method. Its computational complexity is dominated by the Newton update step that takes $O(N^3)$ time [54], where N is the length of a reconstruction vector, which is the number of emission angles in our case (ignoring bilateral zero padding). Since we need to reconstruct M×T vectors, the overall computational time taken by our CS application is bounded by $O(N^3 \times M \times T)$, which is asymptotically more expensive than DAS beamforming.

Putting it all together, when we use the proposed SAS scheme in conjunction with DAS beamforming and coherent compounding, the computational complexity is $O(n \times M^2 \times T)$; on the other hand, when we use our CS application technique in conjunction with DAS beamforming and coherent compounding, the computational complexity is $O(N^3 \times M \times T)$. The latter is computationally more expensive than the former, which is also illustrated in the next section in terms of actual running times.

4.9 Summary

In this section, we summarize the imaging performance of SASB/SASR against CS and UAS when they offer the same amount of sample savings, ranging from 75% to 90%. Additionally, we report the execution times of all the methods under consideration. Our results are listed in table 4.14.

At 75-77% savings, CS offers the best average lateral resolution (FWHM = 0.54 mm), UAS offers the best average axial resolution (FWHM = 0.53 mm). On the other hand, SASB-IMMSE offers the best near-field contrast (X's CNR = 9.34 dB), and SASR-SSIM offers the best far-field contrast (Y's CNR = 8.95 dB).

At 80-81% savings, UAS, SASB-SSIM, and SASR-IMMSE offer the same average lateral resolution (FWHM = 0.57 mm), while the best average axial resolution (FWHM = 0.53 mm) is due to UAS. The best near-field contrast (X's CNR = 9.20 dB) is obtained using SASR-IMMSE, and the best far-field contrast (Y's CNR = 8.69 dB) is obtained using SASR-SSIM.

At 85% savings, UAS and SASR-IMMSE match in terms of the average lateral resolution (FWHM = 0.57 mm), but UAS is the best in terms of the average axial resolution (FWHM = 0.52 mm). Meanwhile, the best near-field contrast (X's CNR = 8.34 dB) is offered by SASR-IMMSE, and the best far-field contrast (Y's CNR = 8.20 dB) is offered by SASR-SSIM.

At 88% savings, SASB-SSIM/IMMSE and SASR-SSIM yield the same average lateral resolution (FWHM = 0.57 mm), while UAS again outperforms the others in terms of the average axial resolution (FWHM = 0.49 mm). It also offers the best contrast (X's CNR = 6.80 dB and Y's CNR = 6.88 dB).

Table 4.14: Cost and quality comparison: FA, CS, SAS, and UAS.

Sample Savings	Acquisition Scenario	Average Lateral/Axial FWHM (mm)	CNR (dB)		Speckle Tests	Execution Time (seconds)
			X	*Y*		
0%	**FA**	**0.54/0.57**	**11.98**	**11.46**	**Pass**	**840**
75%	CS	0.54/0.57	9.18	8.40	Pass	5349
75-77%	UAS	0.57/0.53	6.46	7.27	Pass	113
	SASB-IMMSE	0.57/0.67	9.34	8.00		117
	SASR-SSIM	0.58/0.69	8.80	8.95		113
80-81%	UAS	0.57/0.53	6.74	8.49	Pass	111
	SASB-SSIM	0.57/0.64	6.13	7.58		113
	SASR-SSIM	0.59/0.64	8.12	8.69		117
	SASR-IMMSE	0.57/0.68	9.20	8.13		115
85%	UAS	0.57/0.52	6.63	7.91	Pass	104
	SASR-SSIM	0.58/0.59	7.32	8.20		106
	SASR-IMMSE	0.57/0.64	8.34	7.95		107
88%	UAS	0.58/0.49	6.80	6.88	Pass	99
	SASB-SSIM	0.57/0.61	5.02	6.31		101
	SASB-IMMSE	0.57/0.57	5.02	6.31		103
	SASR-SSIM	0.57/0.61	5.63	6.65		101
90%	UAS	0.57/0.51	5.14	7.24	Pass	85
	SASR-IMMSE	0.58/0.64	5.23	7.14		87

At 90% savings, UAS outperforms SASR-IMMSE in terms of the average lateral/axial resolution (0.57/0.51 mm vs. 0.58/0.64 mm) and the far-field contrast (7.24 dB vs. 7.14 dB). On the other hand, SASR-IMMSE outperforms UAS in terms of the near-field contrast (5.23 dB vs. 5.14 dB).

According to table 4.12, if we wish to focus on the average lateral and axial FWHM values, we should use CS and UAS, respectively. On the other hand, if we are interested primarily in the near-field and far-field CNR values, we should use SASB-IMMSE and SASR-SSIM/IMMSE, respectively. We do not recommend targeting the amount of sample savings in excess of 75-85% when using our proposed acquisition methods, due to the resulting image quality degradation.

Finally, it is also important to consider the execution times, as observed on the following computing system:

- Processor: Intel (R) Core (TM) i5-6200U,
- CPU speed: 2.30 GHz,
- RAM: 8.00 GB,
- OS: Windows 10 (64 bits),
- MATLAB version: R2016b.

Referring to table 4.14, the FA case takes 840 seconds to form the final image. The CS execution time is 5,349 seconds, which is approximately 6.3 times slower than that of the FA case. While CS has resulted in 4-fold reduction in the amount of sampled data (75% sample savings), the computational cost of data reconstruction is high. In contrast, the execution times for the SASB and SASR techniques at 75% sample savings are 113-117

seconds, which is approximately 7.2-7.4 times faster than that of the FA case. Note that the UAS case also takes 113 seconds at 75% sample savings, which confirms that DAS beamforming dominates the overall computational cost even when similarity calculations are involved.

Chapter 5: Conclusion and Future Work

In this chapter we shall summarize our thesis and outline several potential directions of future research and development efforts.

5.1 Conclusion

This thesis makes a case for a new scheme for selecting plane-wave emission angles (i.e., selecting angle-specific raw RF data frames for beamforming and compounding), in a recursive binary search fashion based on the SSIM/IMMSE similarity thresholds. We have abbreviated our methods as SASB and SASR, depending on which data we use (beamformed or raw) to assess similarity. We have also described and evaluated two alternative techniques: theoretically sophisticated but computationally expensive compressive sensing, and simplistic but cheap uniform angle selection targeting a specific amount of savings.

From our results, we can conclude that both SASB and SASR are capable of delivering significant savings, while maintaining acceptable image resolution and contrast quality. In some cases, they perform approximately as well as CS, yet they are approximately as inexpensive as UAS. In other words, the use of SSIM/IMMSE helps us avoid redundant data sampling, while the overhead of computing SSIM/IMMSE during the angle selection process turns out to be insignificant, in comparison to the cost of DAS beamforming that dominates the computational complexity of image formation. Among all four variants of our proposed scheme, we have found that SASR-SSIM and SASR-IMMSE perform relatively better in terms of the far-field and near-field contrast quality indicator values,

respectively. On the other hand, SASB-SSIM performs relatively better in terms of the lateral and axial resolution quality indicator values.

5.2 Future Work

We assume that the SSIM/IMMSE thresholds used to select the plane-wave emission angles are defined by the user in advance and remain constant during the recursive search. As illustrated in chapter 4, these thresholds have an enormous impact on the final image quality and the amount of sample savings achieved.

Ideally, an effective threshold setting should be learned dynamically from the data being processed, which brings us to the first suggestion for future work: investigating the ways an ultrasound system can decide on the threshold values automatically and/or change them adaptively as more data becomes available during acquisition.

Our second suggestion for future work is to investigate other similarity metrics (e.g., peak signal-to-noise ratio measurements). It may also be worthwhile to look into combining different metrics to guide the angle selection process, as opposed to using a single metric throughout.

Finally, our third suggestion for future work is to investigate localized region-of-interest sampling schemes, which may lead to additional savings while improving the image quality where it matters (within user-specified and/or automatically detected segments). We have alluded to ROI-driven acquisition in section 4.5, but more work is needed in this promising direction.

Bibliography

[1] US National Library of Medicine, NIH, “Application of Ultrasound in Medicine” https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3564184.

[2] R. Cobbold, *Foundations of Biomedical Ultrasound*. NY: Oxford University Press, 2007. pp. 420-445, 2007.

[3] T. Szabo, *Diagnostic Ultrasound Imaging: Inside Out*. MA: Elsevier, pp 222-265, 2004.

[4] R. G. Lyons, *Understanding Digital Signal Processing*. NJ: pp. 479-499 Prentice Hall, 2004.

[5] J. A. Jensen, Medical ultrasound imaging. *Progress in Biophysics and Molecular Biology*, 93:153–165, 2007.

[6] T. Bourbie, O. Coussy, B. Zinszner, *Acoustics of Porous Media.* pp. 77, Editions Technip.

[7] M. Albulayli, D. Rakhmatov, “*Evaluation of hybrid GSC-based and ASSB-based beamforming methods applied to ultrasound imaging”*, in Proc. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5130-5134, 2012.

[8] A. Besson, R. Carrillo, O. Bernard, Y. Wiaux, J.P. Tiran, “*Compressed delay-and-sum beamforming for ultrasound imaging”*, in Proc. International Conference on Image Processing, pp. 4651-463, 2016.

[9] H. L. V. Trees, *Detection, Estimation, and Modulation Theory IV:* Optimum Array

Processing. New York: Wiley, 2002.

[10] O. Lorintiu, H. Leggott, M. Alessandrini, O. Bernard, D. Friboulet, "*Compressed sensing reconstruction of 3D ultrasound data using dictionary learning and line-wise sampling"* in Proc. International Conference on Image Processing (ICIP), pp. 2469-2471, 2014.

[11] H. Liebgott, R. Prost, D. Friboulet, "*Pre-beamformed RF signal reconstruction in medical ultrasound using compressive sensing for ultrasonics*", in Ultrasonics, vol. 53, Issue 2, pp. 525-533, 2013.

[12] E. J. Candès and M. B. Wakin, "*An introduction to compressive sensing",* in Proc. IEEE Signal Processing Magazine, March 2008.

[13] H. Liebgott, A. Basarab, D. Kouame, O. Bernard and D. Friboulet, "*Compressive sensing in medical ultrasound imaging"* in Proc. IEEE International Ultrasonics Symposium, 2012.

[14] J. Richy, H. Liebgott, R. Prost, and D. Friboulet, *"Spectral doppler using compressive sensing,"* in Proc. IEEE International Ultrasonics Symposium, Orlando, Fl, USA, 2011.

[15] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, *"Image quality assessment: From error visibility to structural similarity"*, IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, Apr. 2004.

[16] H. Liebgott, A. Rodriguez-Molares, F. Cervenansky, J. Jensen and O. Bernard. "*Plane-wave imaging challenge in medical ultrasound,"*, in Proc. IEEE International Ultrasonics Symposium, pp. 1–4, Oct 2016.

[17] M. Albulayli, D. Rakhmatov. "*Fourier-domain depth migration for plane-wave ultrasound imaging"* in IEEE Transaction on Ultrasonics and Ferroelectrics and Frequency Control, vol. 65, no. 8, pp. 1321-1333, 2018.

[18] C. Scillero and A. Trucco, "*Design of a sparse planar array for optimized 3D medical ultrasound imaging"*, in Proc. 23rd European Signal Processing Conference, pp. 1341-1345, 2015.

[19] A. Ponnle, H. Hasegawa, H. Kanai, "*Suppression of grating lobe artifacts in ultrasound images formed from diverging transmitting beams by modulation of receiving beams"* in Ultrasound Med Biol, vol. 39, no.4, pp.681-691, Apr. 2013.

[20] P. K. Mishra, R Bharath, P Rajalakshmi, and U. B. Desai, "*Compressive sensing ultrasound beamformed imaging in time and frequency domain"*, in Proc. 17th International Conference on E-health Networking, Application & Services (HealthCom), Boston, MA, pp. 523-527, 2015.

[21] T. Chernyakova, and Y. C. Eldar. "*Fourier-domain beamforming: the path to compressed ultrasound imaging"*, Ultrasonics, Ferroelectrics, and Frequency Control, IEEE Transactions, pp. 1252-1267, 2014.

[22] M. Gasse, F. Millionz, E. Roux, D. Garcia, H. Liebgott, and D. Friboulet, *"High-quality plane wave compounding using convolutional neural networks"*, IEEE Trans Ultrasonics, Ferroelectrics and Frequecny Control, vol. 14, no. 8, 2017.

[23] M. Lustig, D. L. Donoho, J. M. Santos and J. M. Pauly, *"Compressed sensing MRI",* in Proc. IEEE Signal Processing Magazine", pp. 72-82, 2008.

[24] Z. Guo, C. Li, L. Song and L. V. Wang. *"Compressed sensing in photoacoustic tomography in vivo*," Journal of Biomedical Optics, vol. 15, no. 2, pp. 021311, 2010.

[25] N. Wagner, Y.C. Eldar, and Z. Friedman, *"Compressed beamforming in ultrasound imaging,"* IEEE Transactions on Signal Processing, vol. 60, no. 9, pp. 4643-4657, 2012.

[26] S. Zobly,and Y.M. Kakah. *"Compressed sensing: doppler ultrasound signal recovery by using non-uniform sampling and random sampling"*, in Proc. 2011 28th National IEEE Radio Science Conference (NRSC), 2011.

[27] J. Arendt Jensen, *"A model for the propagation and scattering of ultrasound in tissue"*, Acoustical Society of America Journal, vol. 89, no. 1, pp. 182-190, 1991.

[28] S. Boyd and L. Vandenberghe. "*Convex optimization"*, Cambridge university press, pp. 288- 312, 2004.

[29] J. A. Jensen, *"Spectral velocity estimation in ultrasound using sparse data sets",* The Journal of the Acoustical Society of America, vol. 120, no. 1, pp. 211-220, 2006.

[30] E. Candès and J. Romberg, *"Sparsity and Incoherence in Compressive Sampling"*, Inverse Problems 23. pp. 969-985, 2007, ISSN 0266-5611.

[31] J. A. Tropp and A. C. Gilbert, *"Signal recovery from random measurements via orthogonal matching pursuit"*, IEE Transactions on Information Theory, vol. 53, no.1, pp. 4655-4666, 2007.

[32] M. Schiffner and G. Schmitz, *"Fast pulse-echo ultrasound imaging employing compressive sensing"*, in Proc. IEEE International Ultrasonics Symposium, Orlando, Florida, USA, pp. 688-691, 2011.

[33] N. Wagner, Y. C. Eldar, A. Feuer, G. Danin, and Z. Friedman, *"Xampling in ultrasound imaging"*, in Proc. SPIE Medical Imaging, 2011.

[34] E. J. Candès and T. Tao, *"Decoding by linear programming"*, IEEE Transactions on Information Theory, vol. 51, no. 1, pp. 4203-4215, 2005.

[35] l1-magic. http://www.acm.caltech.edu/l1magic/.

[36] J. A. Jensen, *"Field: A program for simulating ultrasound systems"*, Medical & Biological Engineering & Computing, vol. 34, no. 1, pp. 351-353, 1996.

[37] A. A. Mohammed, R. Minhas, Q. M. J. Wu, and M. A. Sid-Ahmed, *"An efficient fingerprint image compression technique based on wave atoms decomposition and multistage vector quantization"*, Integrated Computer-Aided Engineering, vol. 17, no. 1, pp. 29-40, 2010.

[38] E. Candès and T. Tao, *"Near optimal signal recovery from random projections: Universal encoding strategies?"*, IEEE Transaction on Information Theory, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[39] D.S. Taubman and M.W. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards and Practice*, 2002.

[40] E. Candès, J. Romberg, and T. Tao, *"Stable signal recovery from incomplete and inaccurate measurements"*, Communications on Pure Applied Mathematics, vol. 59, no. 8, pp. 1207–1223, Aug. 2006.

[41] J. Haupt and R. Nowak, *"Signal reconstruction from noisy random projections"*, IEEE Transaction on Inform. Theory, vol. 52, no. 9, pp. 4036–4048, 2006.

[42] E. Candès and T. Tao, *"The Dantzig selector: Statistical estimation when p is much larger than n"*, The Annals of Statistics, vol 35, no. 6, pp. 2313-2351, 2007.

[43] M. Rudelson and R. Vershynin, *"On sparse reconstruction from Fourier and Gaussian measurements*", Communications on Pure and Applied Mathematics Volume 61, Issue 8, pp. 1025-1045, 2007.

[44] M. Mishali, Y. C. Eldar, O. Dounaevsky, and E. Shoshan, *"Xampling: Analog to digital at sub-nyquist rates"*, IET Journal of Circuits, Devices and Systems, vol. 5, issue 1, pp. 8-20, 2011.

[45] Mishali, M. and Eldar, Y. C., *"Sub-nyquist sampling: Bridging theory and practice"*, IEEE Journal of Selected Topics in Signal Processing, vol 4, no. 2, April 2010.

[46] C. Quinsac, N. Doigeon, A. Basarab, D. Kouame, J.Y. Tourneret, *"Bayesian compressed sensing in ultrasound imaging*", in Proc. IEEE International Workshop on Computational Advances in Multi-Semsor Adaptive Processing (CAMSAP), pp.101-104, 2011.

[47] L. Demanet and L. Ying, *"Wave atoms and sparsity of oscillatory patterns"*, Applied and Computational Harmonic Analysis, vol. 23, pp. 368-387, 2007.

[48] G. Montaldo, M. Tanter, J. Bercoff, N. Benech, and M. Fink, *"Coherent plane-wave compounding for very high frame rate ultrasonography and transient elastography,"* IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control, vol. 56, no. 3, pp. 489–506, 2009.

[49] *Plane-wave Imaging Challenge in Medical UltraSound*, https://www.creatis.insa-lyon.fr/EvaluationPlatform/picmus/about_speckle_quality.html.

[50] R. Cohen, Y. Sde-Chen, T. Chernyakova, C. Fraschini, J. Bercoff, and Y. Eldar, *"Fourier domain beamforming for coherent plane-wave compounding,"* in Proc. IEEE International Ultrasonics Symposium, pp. 1-4, Oct 2015.

[51] R. Dosselmann, X. Yang, "*A comprehensive assessment of the structural similarity index*," Signal, Image and Video Processing, Volume 5, Issue 1, pp 81–91, March 2011.

[52] M. Figueiredo, R. Nowak, and S. Wright, "*Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*," IEEE Journal of Selected Topics in Signal Processing, 1 (2007), pp. 586–597.

[53] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinvesky. "A method for large-scale ℓ1-regularized least squares problems with applications in signal processing and statistics," Tech. Report, Dept. of Electrical Engineering, Stanford University, 2007. Available at www.stanford.edu/˜boyd/l1_ls.html

[54] A. Yang, Z. Zhou, A. Ganesh, S. Sastry, and Y. Ma. *"Fast l1-minimization algorithms for robust face recognition,"* IEEE Transactions on Signal Processing, vol. 22, no. 8, pp. 3234-3246, Aug 2013. Preprint available at https://arxiv.org/abs/1007.3753.