Synthetic Aperture Methods for Medical Ultrasonic Imaging

Hongxia Yao

Contents

Chapter 1

Introduction

> *I see... (an) important application of vision by ultrasound in medical diagnostics where it could not only replace X rays, but score above them by making visible fetuses, clogged veins and arteries, and incipient tumors.*
>
> — Dennis Gabor, 1970

Sound waves, a mechanical disturbance, which are scattered from objects, carry much the same image information as do light waves. How to "see" with sound has intrigued engineers and scientists for decades. Bats, whales, and dolphins do it with ease but the human beings have virtually no such natural ability. The history of engineering and science, however, is a vast demonstration that technological solution can compensate for some deficiencies of nature.

Ultrasound is a sound wave having frequency greater than 20 kHz which is the upper limit for human audible range. The potential of ultrasound as an imaging modality was realized in the late 1940s. From early 1970s, it established itself as a useful diagnostic tool when gray-scale ultrasound was introduced. Now it is one of the most utilized diagnostic modality in medicine. The development of the medical ultrasound imaging techniques is described in many literatures (see e.g. [1] and [8]).

1.1 The Goal of the Thesis

This thesis gives a review and an evaluation of synthetic aperture (SA) methods for medical ultrasound imaging, and presents a new synthetic aperture method which increases frame rate of abdominal imaging systems.

Aperture synthesis technique is originally explored in synthetic aperture radar (SAR). The synthetic aperture is synthesized from the successive use of smaller real aperture, in order to realize a physically unexisted large aperture. The concept of replacing a large, impractical aperture with the combination of a much smaller real aperture was attributed in 1951. Since then, the SAR has been extensively studied, and developed as a searching, sensing and mapping tool. Its success led to the idea of using SA method in other active and passive sensing equipments such as sonars. Since 1970s, the SA method has been applied to digitized ultrasonic signals. Over these years, various SA ultrasound methods have been proposed, including synthetic aperture focusing technique (SAFT) [43], multi-element synthetic aperture focusing (M-SAF) [32] [22], synthetic focusing (SF) and synthetic receive aperture (SRA) [44]. The main purposes of using SA approaches in ultrasound imaging systems have been to reduce the cost and the complexity of the imaging system, as the number of elements and the amount of necessary electronics are reduced with SA.

A new synthetic aperture method — the synthetic transmit aperture method (STA) is proposed in this thesis for the purpose of increasing imaging frame rate. It achieves a higher frame rate than that of s phased array (PA) in conventional abdominal ultrasound imaging using composite transmit focusing. Real-time B-mode ultrasound scanners show images of interior soft-tissue structures of human body by employing the pulse-echo method . The images are shown in real-time in order to perceive the relation between the position of the transducer and the location of the tissue structures. Consequently, the frame rate becomes an important issue in real-time imaging. In the proposed STA method, the transmit aperture is split into N_s subapertures. Each subaperture transmits a pre-focused beam near middle of the region of interest. Echo signals reflected from targets are received at full aperture. After data acquisition from all transmit subapertures, data are combined using synthetic aperture processing. The depth of focus for each subaperture is N_s^2 times that of the full aperture. Thus the equivalent of composite transmit focusing with $N_c = N_s^2$ zones is achieved with N_s transmit subapertures and a frame rate increase of N_s is achieved. In addition to the increase in frame rate, the focusing is of more uniform quality since it degrades gradually away from the pre-focus, and does not have any zone patterns.

The synthetic aperture ultrasound imaging methods, SAFT, M-SAF, SRA, and SPA, as well as the new method STA were implemented, and tested by using radio frequency (RF) data from wire targets, cyst and tissue mimicking phantoms. The results of these methods are discussed. The new STA method was compared with the conventional composite transmit focusing method. The proposed method increased the frame rate significantly and gave compatible image quality.

1.2 Thesis Organization

This thesis contains seven chapters, and two appendices. Here is an overview of the rest of the thesis.

Chapter 2 introduces physical principles of wave propagation and some ultrasound properties. Related terminologies are defined in this chapter.

Chapter 3 provides a brief introduction to the medical ultrasound imaging systems, focusing on transducers, scanning methods, scan conversion technique and image quality factors.

Chapter 4 describes delay and sum beamforming algorithms for a linear phased array. Beam pattern are analyzed, and the sampling requirements in spatial and time domain are given.

Chapter 5 is one of the most important part of the thesis. It discusses various synthetic aperture method, focusing on the new mehtod, the synthetic transmit aperture method (STA). Sampling requirements in spatial domain, particularly for synthetic array, are discussed.

Chapter 6 decribes the experiments, evaluating the synthetic aperture methods for medical ultrasound. The experimental results are discussed.

Chapter 7 gives a final discussion.

In addition to these chapters, two appendices, describing the software and the hardware used for the experiments, are included in this thesis.

Chapter 2

Acoustic Background

Sound is our experience of the propagation of pressure wave through some physical elastic medium. Usually the medium is air, but a liquid works well too. A vacuum doesn't. There is therefore no sound in space. The pressure waves are generated from some types of mechanical disturbance. Sound is a transfer of power as well. Mechanical energy is being converted to a wave form that radiates energy away from the disturbance. Mechanical vibrations become vibrating pressure waves, transferring energy to the medium and to objects that the wave contacts. Ultrasound is a sound wave having frequency greater than 20 kHz, containing acoustic wave properties.

In this chapter some of the most important terminologies and equations related to the wave propagation will be introduced. The solution to the wave equation which is used in beam pattern analysis will be given. Properties of ultrasound wave will also be discussed.

2.1 Ultrasound Wave Propagation

There are two basic types of acoustic waves: the longitudinal wave and the shear wave. In a longitudinal wave, the motion of a particle in a acoustic media is only in the direction of propagation. In a shear wave, the motion of a particle is transverse to the direction of propagation. Ultrasound waves in gases and liquides are caused by oscillations of the particles back and forth about their equilibrium points, and are therefore longitudinal waves.

Information about distant events is carried to receiving apertures by propagating waves. The physics of propagation is described by wave equations for the appropriate medium and boundary conditions. Ultrasound waves generated by transducers, are signals governed by laws of the physics of propagation, in particular the wave equation.

Biological Tissues	Velocity (m/sec)
Blood	1550
Fat	1450
Liver	1570
Kidney	1560

Table 2.1: *Velocity of biological tissues at room temperature* $(20^\circ - 25^\circ C)$.

2.1.1 Wave Equation

The propagating ultrasound waves are governed by the wave equation. Detailed derivation of the wave equation can be found in [34] and [49]. The wave equation is

$$\frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2} + \frac{\partial^2 s}{\partial z^2} = \frac{1}{c^2}\frac{\partial^2 s}{\partial t^2} \tag{2.1}$$

where $s(x, y, z, t) = s(\vec{x}, t)$ represents the sound pressure in space and time, and c is the speed of propagation. The speed of ultrasound wave varies in different biological tissues [40], some typical values are shown in Table 2.1. Note that Eq. 2.1 is only valid in homogeneous, linear and lossless media. Unfortunately, the human tissue is not such an ideal medium, and exhibits tremendous complexities in the interaction with sound [8]. Thus, physical phenomena like refraction, dispersion, attenuation and diffraction can occur. The wave equation must thus be modified to model wave propagation in inhomogeneous media.

2.1.2 Solutions to the Wave Equation

Eq. 2.1 has many solutions. When solved in Cartesian coordinates, the harmonic solution has a complex exponential form

$$s(\vec{x}, t) = Ae^{j(\omega t - \vec{k}\vec{x})} \tag{2.2}$$

where A is a complex constant, $\vec{k} = (k_x, k_y, k_z)$ is the wave number and ω is the radian frequency.

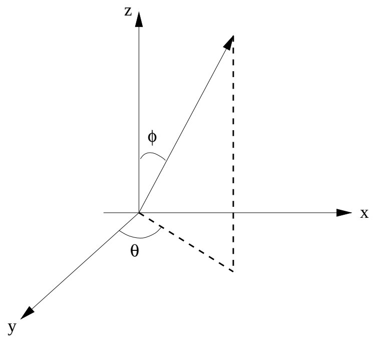
Substituting Eq. 2.2 into the wave equation, we obtain

$$|\vec{k}|^2 = \frac{\omega^2}{c^2}$$

As long as this constraint is satisfied, signals with the form of Eq. 2.2 satisfy the wave equation.

The solution given by Eq. 2.2 may be interpreted as monochromatic plane wave. Monochromatic means one color. Here it refers to a wave with one temporal frequency ω. If we place a sensor at some fixed position $\vec{x}^o = (x^o, y^o, z^o)$, to observe the signal, the received signal has the form

$$s(\vec{x}^o, t) = Ae^{j(\omega t - \vec{k}\vec{x}^o)} \tag{2.3}$$



$$\begin{aligned} r &= \sqrt{x^2+y^2+z^2} \\ \theta &= \sin^{-1}\left(\frac{x}{\sqrt{x^2+y^2}}\right) \\ \phi &= \cos^{-1}\left(\frac{z}{r}\right) \end{aligned}$$

Figure 2.1: *The relationship between Cartesian Coordinates and Spherical Coordinates.*

The linearity of the wave equation implies that many plane waves propagating in different directions can exist simultaneously. Any signal can then be expressed as a weighted superposition of the complex exponentials.

In spherical coordinates, the solution is

$$s(r,t) = \frac{A}{r} e^{j\omega(t-\frac{r}{c})}$$

The relation between the spherical coordinates (r, ϕ, θ) and the Cartesian coordinates (x, y, z) is shown in Fig. 2.1. Detailed derivation can be found in [21]. This solution can be interpreted as a spherical wave propagating outward from the origin with a temporal frequency of ω.

A complete solution of the equation is very complex and will not be discussed here. It can be found in [23] [34].

2.1.3 Wave Parameters

Huygens' principle states that each point on a transducer surface acts as a source of a spherical wave. The main lobe of an aperture's directivity pattern is called a beam. The beam of an ultrasound transducer can be calculated accordingly. Just as in optics, the beam is composed of two distinct regions of interest, the near-field region (the Fresnel zone), and the far-field region (the Fraunhofer zone) [23]. The near-field condition is

$$z < \frac{a^2}{\lambda}$$

where z is range from the transducer surface. a and λ are the transducer radius and the ultrasound wavelength, respectively. Waves propagate as plane or spherical waves depending on the propagation distance from the source point. In the near-field, wave front is curved and the spherical wave are assumed, while plane waves are assumed in the far-field (see Fig 2.2).

In traversing a medium, a sound wave can be characterized both by temporal and spatial parameters. For waves propagating in the a media with a temporal frequency f,

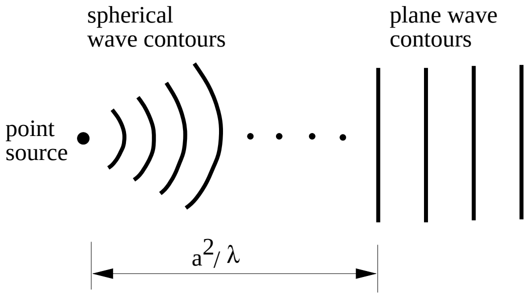


Figure 2.2: *Wave propagation varies in the near-field and the far-field region.*

the corresponding temporal angular frequency is $\omega = 2\pi f$. $\vec{k}$, called wavenumber vector, is considered as a spatial frequency variable. It is parallel to the propagation direction of the wave with magnitude $|k| = 2\pi/\lambda$. For simplicity, $\vec{\alpha} = \vec{k}/\omega$ called a slowness vector is used. Its magnitude is $|\alpha| = 1/c$. The wave length is the distance propagated during one temporal period, given by

$$\lambda = c \cdot t = \frac{c}{f} \tag{2.4}$$

where c and t are the propagation speed and the temporal period, respectively. λ can be considered as a spatial period variable as t is a temporal period variable.

2.1.4 Pulsed Ultrasound Wave

Ultrasound can propagate as continuous waves (CW) as well as pulsed waves (PW). Pulsed waves are commonly used in active sensing methods, also in ultrasound imaging, while both PW and CW are used for ultrasound Doppler measurements. When using CW, the ultrasound probe must transmit and receive simultaneously and therefore it contains separate transmit and receive aperture. In case of PW, the transducer transmits a pulse, waits a predefined time interval, and then receives echo signals. The length of the pulse generated by a transducer is inversely proportional to the frequency bandwidth of the transducer, as illustrated in Fig. 2.3. The relation can be easily found by applying Fourier transform to the pulse wave form.

2.2 Ultrasound Properties

The most essential fact of medical ultrasound physics is that acoustic energy is transported through tissue as a sound wave. This acoustic wave propagates by longitudinal compression. Because it is a wave, ultrasound is subjected to all the wave-like behaviors of classical physics — scattering, refraction, attenuation, and diffraction.

2.2.1 Reflection, Refraction and Scattering

When an ultrasound transducer emits sound energy into the body, it travels through tissue unimpeded until it scatters from the abrupt changes in acoustic impedance at the interfaces

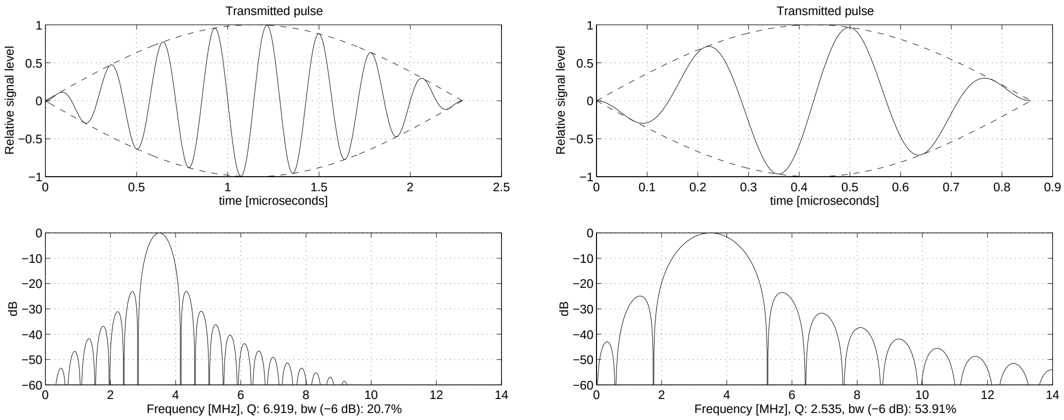


Figure 2.3: *Shorter pulsed waves result wider frequency bandwidth. left: the wave form and the frequency response of a long pulse; right: those of a short pulse.*

between different tissue types. Some of the sound is scattered back to the transducer to be received and processed to extract information, while the remainder travels on and to the next reflector and so on. Thus, the reflection property of sound makes ultrasound imaging possible. Each tissue type has a characteristic acoustic impedance defined as $z = \rho c$, where ρ is the density of the tissue and c is the velocity of sound in the tissue.

Wave refraction is a change in direction of the wave as it travels through tissue which is caused by change in the sound velocity in the tissue. This phenomenon can cause image distortions, because the reflected ultrasound used to form an image may not travel in a straight line from transducer to target and back again.

Scattering is the result of reflection. Each small tissue volume which contains tissues of two or more impedances is a scattering center. If the characteristic size of the scatterer is small compared to the ultrasound wavelength, as in the case of red blood cells, then the sound energy is scattered uniformly in all directions. This is called diffuse scattering and the energy reflected back to the transducer is small. If, on the other hand, the scatter site is comparable to the wavelength as in the case of a vessel wall, then the energy is reflected mostly at an angle equal to that of incidence. This is called specular scattering and the reflected energy can be large or small depending of the incident angle.

2.2.2 Attenuation

As the ultrasonic beam travels through the medium, the wave intensity is attenuated. Attenuation is caused by local energy absorption, reflection, scattering, etc. Absorption is the local conversion of ultrasound wave energy to other energy forms, mostly heat. Attenuation constant can be modeled as [37]

$$\mu(f) = \alpha f^{\beta}$$

α and β are acoustic parameters of the medium, and f is the transducer center frequency. The equation shows that the attenuation is proportional to the center frequency for $\beta = 1$.

Consequently, higher frequency ultrasound transducers result in an increase of attenuation. In human tissue, typical values for α, β are $\beta = 1$ and $\alpha = 1$dB/cm/MHz [37]. Attenuation of tissues is significant and limits not only the depth of penetration but the spatial resolution as well.

2.2.3 Diffraction

Diffraction where the wave is deviated from a straight line occurs due to wave nature. Because of diffraction, the beam emitted from the transducer will increase its diameter with distance. Within the near-field region, the outside diameter of the beam remains essentially uniform; the beam then spreads beyond this region. Therefore, it is the primary determinant of the shape of the ultrasound beam. A detailed discussion of diffraction can be found in [23].

2.2.4 Speckle

Speckle is another important result of a particular property of ultrasound — wave coherence. Speckle is produced because the advancing wavefront is of finite size and simultaneously reflects from many scatters. The received wavefront is the coherent sum of the amplitude of sound reflected from each one. Speckle is multiplicative noise and its characteristic size is determined primarily by the transducer frequency, shape of transmitted pulse and the beam width. It adds a braininess or texture to the ultrasound image which limits spatial and contrast resolution.

Chapter 3

Medical Ultrasound Imaging

Medical images, which display internal structures of human bodies, are inevitable tools in medical diagnosis. Medical ultrasound, magnetic resonance (MR), and computed tomography (CT) are presently most advanced methods for medical imagery. This thesis deals with medical ultrasound techniques. Compared to the MR and CT devices, the medical ultrasound devices are often less expensive and simpler to use. Ultrasound imaging systems are attractive for diagnosis because of their non-invasive nature and their apparent safety.

This chapter gives the basic principles of medical ultrasound imaging systems, including transducers, scanning methods and scan conversion technique. A medical image must be of sufficient quality to insure an accurate interpretation by the clinician. The most commonly used image quality measures are spatial resolution, contrast resolution, temporal resolution, and signal-to-noise ratio. Some recently developed methods to improve image quality are reviewed.

3.1 Transducer and Arrays

Piezoelectric transducers are the most widely used ultrasonic transducers for converting electrical energy into mechanical energy and converting mechanical energy into electrical energy [40]. They emit short pulses and receive reflected echoes. Single transducer element and various types of transducer arrays, such as linear phased array and annular array, are applied according to different applications. In abdominal imaging, a linear phased array is used to show internal organs of the human body, such as kidneys, liver, pancreas, spleen and gallbladder. In intravascular imaging, a circular prefocused single element transducer is mounted on the tip of a catheter for visualizing blood vessel lumen and vessel wall.

3.1.1 Transducer Materials

The key component of an ultrasound transducer is the piezoelectric element which transmits and receives sound. During transmit, this element converts electrical to mechanical energy. In the receive mode, the returning acoustic wave is converted by the piezoelectric element to an electrical signal, and then routed to the receiver electronics. The element can be made from a variety of materials. A measure of the performance of a material as a transducer is the electrical-to-mechanical coupling coefficient (ECC), defined as [40]

$$ECC = \frac{\text{stored mechanical energy}}{\text{total stored energy}}$$

Barium titanate was the first to be discovered. Today PZT (lead zirconate titanate) ceramic is the most commonly employed due to its high ECC, high dielectric permittivity and low dielectric losses. The limitations of PZT such as high acoustic impedance and the lack of mechanically flexible, have led to an interest in copolymer and polymer materials [10]. Polymer materials such as polyvinylidene fluoride (PVDF) provide a good acoustic match to tissue but exhibit a poor electrical-to-mechanical coupling efficiency which has limited the applications of these materials in medical imaging. Composites have good electrical-to-mechanical coupling. They can be matched acoustically to tissue, and can be flexible enough to be formed into a variety of shapes. The disadvantage is its high fabrication cost. Recently, they are taking the place of PZT and used for most transducers [17].

3.1.2 Single Element and Array Transducers

The simplest transducer uses one single piezoelectric element. For imaging it is mounted on a motorized pointing device. This type of transducer is simple to build and requires the least complex electronic signal processing — only one transmitter and receiver. It doesn't require beamforming techniques.

Multi-element transmission and reception can be used to increase the electronic signal to noise ratio (SNR). An ultrasound array consists of a collection of transducer elements located at distinct spatial locations. Each element can transmit and receive ultrasound signals. There are several different types of array element arrangements, such as linear phased array, annular array and 2-D array [40].

Annular Array
An annular array is fabricated by cutting a spherically shaped single crystal into concentric rings. Each ring is acoustically independent and has its own signal processing circuitry. The number of rings is typically 2 to 8. By putting defined delays on the transmit for each ring and by varying the relative receive times for each ring, a variable electronic lens is formed. The focus for a annular array is symmetric, providing both in-plane and out-of-plane focus. This geometry provides optimum image contrast because the 3-D volume at the focus is minimized. To steer beam direction, the transducer has to be rotated producing a sector image format, or moved linearly producing a linear image format.

Linear Phased Array
A linear phased array consists of a set of elements located along a line with equal inter-element distance, each of which can individually transmit pulses and receive echo signals. If steering is required, the elements should be spaced no further than one-half of a wave

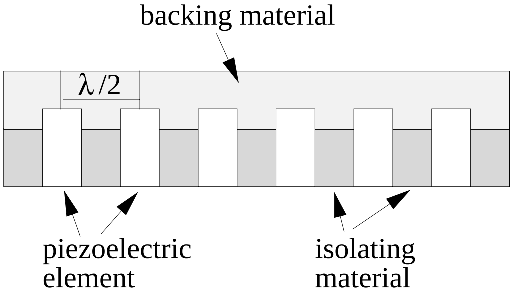


Figure 3.1: *A view of linear phased array with 6 elements.*

length ($\lambda/2$) apart for preventing unwanted grating lobes in the radiation pattern (see Fig. 3.1). Otherwise, larger element distances can be used. Linear phased arrays are commonly used in modern clinical ultrasound scanners.

Curved Linear Array
A curved linear array contains a collection of elements located along a slightly curved line. With this design it permits better contact with a curved abdomen, and the general handling of the array is easier. The array is used to produce a sector image with sufficient width.

2-D Arrays
A 2-D array can be square in shape and arranged in rows and columns, and contains a large number of elements. They are difficult to construct because of the many channels to handle. For instance, a linear phased array usually has 32 to 128 elements [40], while a 2-D array may have $32 \times 32 = 1024$ to $128 \times 128 = 16384$ elements. Most current studies of 2-D arrays are based on computer simulations [46]. Nevertheless, 2-D arrays are promising for volumetric (3-D) imaging [42][5]. Some techniques have been suggested to reduce the number of the elements of 2-D arrays. For instance, sparse arrays reduce the number of elements by randomly removing elements from periodic dense arrays. Lockwood *et al.* [27] developed methods for designing sparse periodic linear and 2-D arrays. Holm *et al.* [18] proposed a weight optimization algorithm to minimize peak sidelobe level in order to seek an optimized sparse arrays.

3.2 Signal Processing Algorithms

A major application of ultrasound in medicine is ultrasound imaging, in which sound waves are used to produce a pictorial representation of biological soft tissues. A simplified ultrasound imaging system is shown in Fig. 3.2. Transducer generates and transmits pulsed ultrasound waves, and receives reflected echo signals. The transducer is controlled by an RF-unit, which is responsible for focusing and steering of the ultrasound waves. The RF-unit, equipped with the radio frequency (RF) transmitter/receiver electronics, applies a beamforming on signals. The second block, real time scanline processing, extracts the

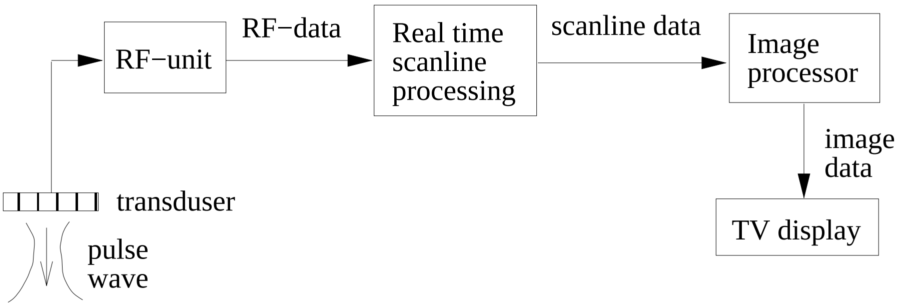


Figure 3.2: *Block diagram of an ultrasound imaging system.*

data acquired by the RF-unit. Speckle reduction can be done here. The scanline data can be sent to image processor, which performs scan conversion as well as compression and interpolation. Finally, the image is displayed on a TV monitor. We note that this block diagram presents the most simple and most commonly used ultrasound imaging systems, in order to illustrate the basic principle of the ultrasound imaging technique.

For a conventional phased array imaging system, all elements transmit simultaneously pulsed wave to generate a prefocused beam. The prefocused beam is transmitted into tissue at a particular direction. Received RF data is delayed for focusing and summed to produce a single line of RF data. Then a prefocused transmit beam is fired at a new direction and formation of the next image line begins. After the transmit beam is swept a 90° sector, a polar image with beam angle as horizontal axis and range as vertical axis is generated. Using scan conversion techniques, a Cartesian image can be displayed on the monitor.

3.2.1 Scanning Methods

To form an image, sweeping a beam over a cross-section of an object is necessary. In medical imaging, two methods are used to sweep beams: mechanical and electronic scan. For each method, two scan formats are usually used: linear and sector scan.

Mechanical scan does not need complex electronic circuitry. A beam is swept by moving or wobbling the transducer with a stepping motor to perform a linear or sector scan.

- The sector scan usually requires a smaller acoustic window compared to the linear scan, and is therefore particularly useful in cardiac imaging.
- The linear scan usually needs a larger acoustic window.

Electronic scan is usually produced with array transducers. There are two types of array transducers: 1-D, and 2-D array transducers.

- In a sector scan, a beam is steered by signals that are applied to the elements of an array with delays that are a linear function of the element positions in the scan direction. In order to avoid grating lobes, inter-element distance should be $\lambda/2$ or less.

- In a linear scan, the elements of an array transducer are excited or multiplexed one by one or group by group sequentially in one direction to sweep a beam. Elements can be spaced larger than $\lambda/2$ apart.
- In a curved linear scan, the elements of a curved array transducer are excited or multiplexed one by one or group by group sequentially in one direction to sweep a beam.

3.2.2 Pulse-Echo B-mode Imaging

The ultrasound pulse-echo technique [40] is a commonly used imaging method based on the waves reflected or scattered from tissue interfaces. Even though the complexity of the interaction of sound with soft tissues considerably complicates the interpretation of pulse-echo data, the clinical versatility of this method makes one of the most important ultrasound imaging techniques existing today. Two families of classical ultrasound examination are available [1]:

1. One family consists of pulse-echo imaging methods. Amplitude demodulation results in a series of display methods showing echo amplitude vs. depth along the path of the ultrasound pulses emitting from the transducer.

2. Another family consists of phase demodulated Doppler methods. The methods work on a well-known principle: The sound emitted by or reflected from a moving object will shift to a higher frequency when moving toward the stationary observer and to a lower frequency when the object is moving away. They are used to show the location and speed of blood flow in two different ways: The continuous-wave (CW) method is performed by two transducers, one for continuously transmitting the sound wave and another for receiving echo signals. And the pulsed Doppler method is performed by sending bursts of sound into the tissue and receiving them at a predetermined Pulse Repetition Frequency.

Depending on how the information is displayed, pulse-echo imaging methods can be classified as A, B, C and M modes.

A (Amplitude) mode display is the oldest and simplest type of pulse-echo ultrasound instrument. The echo amplitude is displayed as a function of depth of penetration.

B (Brightness) mode instrument can represent the information carried by the A-mode as bright spots. Grayscale is often used, and a 2-D display of a scan sector is the most commonly used.

C (Constant depth) mode instrument is very similar to that of B mode. Difference is that the scanning plane is fixed at a constant depth from the probe.

M (Motion) mode instrument displays the depth of penetration along one axis, while the other axis represents time. The echo amplitude is shown in a grayscale.

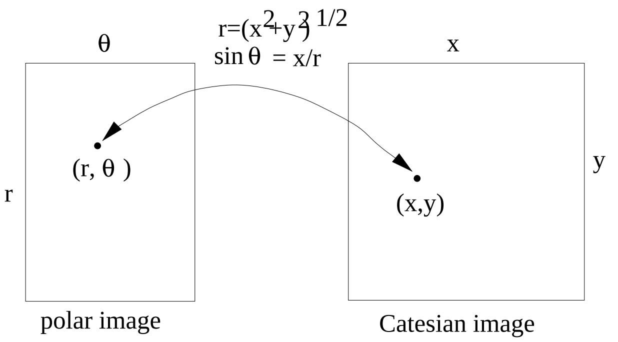


Figure 3.3: *Output-to-input mapping method for scan conversion. For each pixel (x,y) in the Cartesian image, its corresponding position in the polar image can be found by Eq. 3.1 and 3.2.*

Among pulse-echo techniques, the B-mode imaging, which shows images of interior soft-tissue structures, is the most common. In this method, a short ultrasound pulse is emitted into the body and is reflected at boundaries between tissues with different characteristic impedances. By controlling the direction of the emitted ultrasound beam either mechanically or electronically, a set of A-mode scanlines can be formed. Applying scan conversion technique, a B-mode cross sectional image is reconstructed.

3.2.3 Scan Conversion

After a beam is swept over a cross-section of an object, a polar image with vertical axis as range r and horizontal axis as angle θ is formed. Scan conversion is necessary to convert from a polar coordinates associated with the ultrasound data to the Cartesian coordinates. The Cartesian image is an image space broken up into small, usually rectangular, regions called pixels. The pixel size is set by the desired clarity of the image. There should be at least 256 × 256 pixels and preferably up to 512 × 512 to avoid the “blocky” appearance obtained with low number of pixels [38]. As shown in fig. 3.3, the value of each pixel (x,y) can be obtained by using an output-to-input mapping [30]. A pixel is first identified in the Cartesian image. Its coordinates are then mapped to a fractional pixel location in the polar image. For each pixel (x,y) in the Cartesian image, the corresponding position in the polar image can be obtained by following equations:

$$r = \sqrt{x^2 + y^2} \tag{3.1}$$

$$\theta = \arcsin\left(\frac{x}{r}\right) \tag{3.2}$$

the signal value at the (r, θ) position is then obtained by an interpolation. Since the determined pixel position is fractional, some form of interpolation must be performed.

3.2.4 Interpolation

In ultrasound literature, little information can be found about reconstructing and resampling of polar images into rectangular images. The theory behind optimum reconstruction

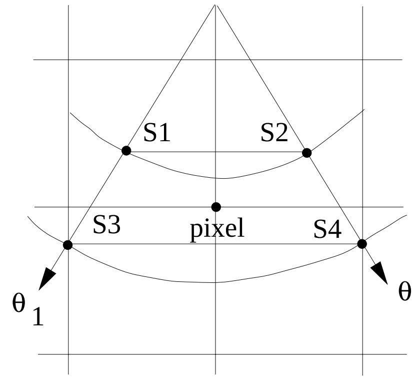


Figure 3.4: *Bilinear interpolation. The four sample values (S1-S4) surrounding each pixel are used to calculate the value of that pixel.*

and interpolation of 1-D and 2-D signals is well-known [33], but few studies are concerned with the interpolation of ultrasonic images.

There are many different conventional interpolators including 2-D windowed sinc, bicubic spline, 4 × 4 point bicubic spline, bilinear and nearest neighbour interpolation method [3]. For real time implementations, either nearest-neighbour or bilinear interpolation is usually used in scan conversion, as these two methods are computational efficient. Recently, several algorithms were proposed to enhance the speed of computation. One was implemented by hardware [36], and the others were implemented by software [3]. In [3], a subjective image quality (contrast and sharpness) analysis was made for various interpolators. The authors found that the subjective differences between the sinc, bicubic spline and bilinear interpolators were very small for complete sector scans, and bilinear interpolator consumed least computation time. Thus, the bilinear interpolator is applied to the experiments in the thesis.

Bilinear interpolation is often used in real-time scan conversion [24] [38]. The method is illustrated in Fig. 3.4. The interpolation transforms from a polar image I_1 to a Cartesian image I_2. For each pixel in I_2, the corresponding position in I_1 is computed using Eq. 3.1 and 3.2. In I_1, four sampled data points S1, S2, S3 and S4, closest to this position, are used for the interpolation. The pixel value is determined by a weighted average of the sampled data value.

Assume that the positions of S1, S2, S3 and S4 are $(r1, \theta1)$, $(r1, \theta2)$, $(r2, \theta1)$ and $(r2, \theta2)$. The value of the pixel to be interpolated is

$$\begin{aligned} p &= S1(\theta2 - \theta)(r2 - r) + S2(\theta2 - \theta)(r - r1) \\ &\quad + S3(\theta - \theta1)(r2 - r) + S4(\theta - \theta1)(r - r1) \end{aligned} \tag{3.3}$$

where (r, θ) is the position of the pixel p.

3.2.5 Gain Compensation

As mentioned previously, the ultrasound energy is attenuated when traveling through the medium. Two identical targets at different depths will produce echoes of different amplitude. The echoes produced by the closer target are stronger. In order to solve this problem, the time-gain compensation (TGC) is used. In this method, the gain of the amplifier is increased as a function of depth to compensate for the loss in energy. Various forms of TGC, such as exponential, linear and windowed TGC are available. In this thesis, a linear TGC with $\beta = 1$ (see Sec. 2.2.2) is applied, as it is the most simple one, and suitable for the most applications. Echo amplitude increases as a linear function of depth.

3.2.6 Logarithmic Compression

Logarithmic compression helps in reducing the dynamic range of the B-scan images for display on a monitor as well as enhancing weak backscatterers. In most imaging systems, the amplitude of the raw signal from the object is compressed so that weak targets are visualized together with the strong targets. All images shown in this thesis are compressed by logarithmic compression which is defined by

$$s'(t) = 20 \log_{10} (s(t)) \tag{3.4}$$

where $s(t)$ and $s'(t)$ are the original and compressed envelopes, respectively.

3.3 Image Quality Factors

An ultrasound diagnosis is ultimately made from a video image. This image must be of sufficient quality to insure an accurate interpretation by the clinician. The most commonly used image quality measures are spatial resolution, contrast resolution, temporal resolution, and signal-to-noise ratio (SNR).

3.3.1 Spatial Resolution

Spatial resolution is defined as the minimum distance at which two bright point reflectors can be resolved. Its value is a key parameter in quantitating the ability to spatially resolve small structures and to provide good edge definition within the ultrasound image. In ultrasound the spatial resolution is determined mainly by basic physics and has three components. The axial component is measured along the distance from the transducer to the objects and parallel to the ultrasound beam. The lateral component is measured in the direction perpendicular to the ultrasound beam axis. These two components form a scan plane. In 2-D ultrasound imaging, spatial resolution refers only to axial and lateral resolution. In 3-D imaging, a third component, the out-of-plane component, perpendicular to the scan plane, is used.

Axial Resolution

Axial or range resolution refers to the ability of the system to separate objects located along the propagation axis of the ultrasound beam. It is determined by the length of

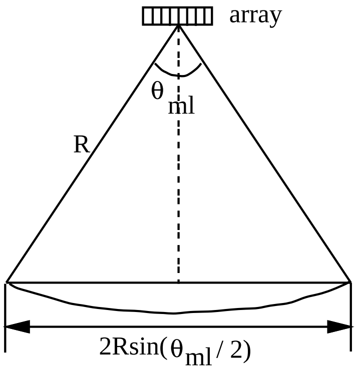


Figure 3.5: *The geometry of a transducer array and its main lobe.*

the transmitted pulse, T_p, which is inversely proportional to the transducer frequency bandwidth B_w. The axial resolution is

$$\Delta r = c \cdot \frac{T_p}{2} = \frac{c}{2B_w}$$

Where c is the propagation velocity. Consequently, higher transducer frequency producing shorter pulse results in an improved axial resolution.

On the other hand, higher frequency results in an increase of attenuation of ultrasound in tissue. Therefore, efforts to improve resolution by increasing the frequency, will decrease the maximum penetration depth. The compromise should be obtained between the desired penetration depth and the axial resolution.

Lateral Resolution

Lateral resolution is used to measure the ability of an imaging system to separate objects located perpendicular to the beam axis. It is primarily determined by the beam width of the transducer. It is also governed by the shape of the sound which, in general, is characterized by the aperture (determined by number of elements and pitch), the transmitting frequency and by focusing of the transducer. For phased arrays the beam in the scan plane is focused electronically. The lateral resolution is defined by

$$\Delta l = 2R \sin \frac{\theta_{ml}}{2}$$

The lateral resolution is dependent on both the angular width of the main lobe θ_{ml}, and the range R from the transducer surface (see Fig. 3.5). Since for small β we have

$$\sin(\beta) \approx \beta$$

the lateral resolution may be given by

$$\Delta l \approx R\theta_{ml}$$

The width of main lobe will be discussed in details in the next chapter. Increasing the transducer frequency results in a narrower beam width and gives a better lateral resolution.

Another important issue in lateral resolution is the presence of side lobes. sidelobes are in fact interference patterns generated by the regular element pattern of the arrays and are undesired responses in directions other than the main lobe.

A special sidelobe known as the grating lobe may appear when the main lobe is steered at large angles. Echoes from the grating lobe may be erroneously interpreted as coming from the steering beam direction, causing artifacts in the image. One factor determining the appearance of grating lobes is the pitch, p, which is the center-to-center distance between two adjacent elements. Theoretically, the grating lobes can be completely removed if

$$p \leq 0.5\lambda_{min}$$

as discussed in [21]. Here, λ_{min} the minimal wavelength, is the wavelength of the highest frequency component in the pulse spectrum.

3.3.2 Contrast Resolution

Contrast resolution can be defined as the smallest difference in tissue acoustic impedance which can be resolved and displayed as two different gray levels in the image. Adequate contrast resolution is essential for the tissue discrimination required for diagnostic ultrasound imaging. The primary factors influencing contrast resolution are spatial resolution, systems dynamic range, sidelobe level, and image noise. Detection of high contrast targets is limited by the imaging systems spatial resolution, while detection of low contrast targets is restricted by the image speckle [41]. Reduced sidelobe levels can also result in a better contrast resolution.

Sometimes the contrast resolution is referred to as the dynamic range of the image which is defined as the ratio of the strongest signal to the weakest signal, often presented in dB.

3.3.3 Temporal Resolution

Temporal resolution is defined as the ability to resolve movements of structures in the image. The primary determinant of the temporal resolution is the time required to form an image and usually refers to frame rate which is the number of images per unit time. The time, T, takes to generate an image is equal to the time to process a beam, T_b, multiplied by the number of beams, M. We have to wait at least the time $2R/c$ to collect the data from the depth R, before the next beam can be transmitted. To make sure that the signal from the preceding pulse is sufficiently attenuated, an extra period of time, T_0 needs be counted. The frame rate is the inverse of T

$$\text{frame rate} = \frac{1}{T} = \frac{1}{MT_b} = \frac{1}{M(2R/c + T_0)}$$

Frame rate is an important issue in real-time imaging system. Low frame rate means that moving structures (e.g. heart valves) are not easily imaged and diagnosis may be impaired.

3.3.4 Signal-to-noise Ratio

Signal-to-noise ratio (SNR) is a measure of the amount of signal information relative to the interfering noise which is always present in imaging. If a single transducer element is located at the spatial origin, its response to a noise-corrupted signal is

$$y(t) = s(t) + n(t)$$

where $s(t)$ represents the signal field and $n(t)$ the noise field. For simplicity, assume that the desired signal is a wide band plane wave of the form $s(t) = s(t - \vec{\alpha}^o \vec{x})$. In addition, assume that s and n are stational random field. The SNR is

$$\text{SNR} = \frac{\varepsilon[s^2(t)]}{\varepsilon[n^2(t)]}$$

This equation is valid when signal and noise components are uncorrelated. For increasing SNR, noise should be reduced. Noise is of two basic types: speckle and random electronic noise. Random noise is due to the random motions of electrons in the signal processing circuitry. It can be reduced and minimized by careful electronic and acoustic engineering design. Speckle noise is produced by the coherent scattering of the acoustic wave from many small scattering centers. For a M-element transducer array, the SNR is M times that of a single element.

3.4 Discussion

This chapter gives the basic principles of the medical ultrasound imaging technique. A ultrasound beam can be scanned either mechanically or electronically. This thesis will concentrate on electronic beamforming and B-mode image representation. In order to form beams electronically, a transducer array must be used. There are different types of array, as discussed in this chapter. In the rest of the thesis, only the linear array is to be used, as it is the most simple and commonly used array form. However, the method developed in this thesis can be extended to other forms of array.

Interpolation is one of the important process in ultrasound imaging. It is used during the scan conversion, a transform associating a polar image with a Cartesian image. It is also used during the delay and sum beamforming, since signal is sampled in discrete points, and the computed time delay may not be exactly on a sampling point. This the reason why an interpolation is needed. In the case of scan conversion, a 2-D interpolation is required. In the case of time delay computation, a 1-D interpolation is sufficient. There are many different interpolation methods. The bilinear interpolation will be used in the rest of thesis as it is easy to implement, relatively efficient and has sufficiently good performance.

Usually, a Cartesian representation of a ultrasound image is required, in order to display the true geometry of the objects. The transform from the polar coordinates to the Cartesian coordinates is called a scan conversion. An output-to-input scan conversion method is presented in this chapter. The algorithm goes through all the pixels in the output polar image to find the corresponding pixels in the input polar image. The advantage of this method is that all the pixels in the output image are totally scanned without upsampling the input image. However, in this method an interpolation must be used to determine the corresponding value at the input image. On the other hand, if one goes through all the pixels in the input image and computes the corresponding locations in the output image, an upsampling should be usually performed in the input image to avoid holes in the output image.

In this chapter, image quality factors are discussed. Interpretation of the acquired images with poor image quality is often difficult. This limits the application of ultrasound in the diagnostic work. Many methods are proposed to improve image quality from various aspects [7] [9] [11] [13] [39] [47] [44].

A good resolution and a high SNR are particularly important to obtain high quality images. Elgarem [7] proposed a multi-depth synthetic aperture processing technique to increase lateral resolution and SNR. Freeman *et al.* [9] proposed a retrospective filtering technique to lengthen the depth of focus and thus to improve the contrast resolution. Shen *et al.* [39] presented an object-dependent post-beamforming processing technique for improving contrast resolution of ultrasonic imaging. Trahey *et al.* [44] suggested a synthetic receive aperture method to give a higher SNR and a better imaging resolution at the cost of reduced frame rate. Many methods [11] [47] were proposed to reduce speckle noise. The speckle reduction will improve the contrast resolution. Apodization [37] is also commonly used to improve contrast resolution by reducing the sidelobe level. Unfortunately, it causes a loss in lateral spatial resolution due to the increase in the width of the main lobe. Apodization is usually used with arrays and is fairly simple to implement.

In this thesis, a new method is introduced for the improvement of image quality. In this new method, the synthetic aperture approach is applied to increase the image frame rate, resulting in a better temporal resolution.

Chapter 4

Real Aperture Beamforming

Beamforming techniques play one of the most important roles in ultrasonic imaging [28]. Beamforming is an array processing algorithm that focuses the array's signal-capturing abilities in a particular direction. The beam is controlled by beam control methods, such as dynamic aperture, dynamic focusing and apodization.

This chapter describes the basic principles of the delay and sum beamformer for a linear arrays, and the methods for analyzing and controlling the beam pattern. The delay and sum beamformer is commonly used in medical ultrasound imaging as it can focus the beam to objects in short range.

4.1 Delay and Sum Beamfomer

Delay and sum beamformer [21], the oldest and simplest array signal processing algorithm, remains a powerful approach today.

4.1.1 Basics

Electronic beamforming can be done by using transducer arrays. There are different types of array, such as linear array, curved array, rectangular array and circular array. The idea behind the delay and sum beamforming is simple. If a propagating signal is present in an array, the outputs of the beamformer, obtained by delaying the array inputs in an appropriate amount of time and adding them together, reinforce the signal with respect to noise or waves propagating in other directions.

Consider a linear array of N elements equally spaced by d units along x-axis. The origin of the coordinate system is located at the center of the array. The location of the n'th element is

$$x_n = \left(n - \frac{N-1}{2}\right) d, \quad \text{for } 0 \leq n \leq N-1 \tag{4.1}$$

See Fig. 4.1.

When we observe a wavefield $f(\vec{x}, t)$ through the array ($\vec{x} = (x, y, z)$), the waveform measured by the n'th element is $y_n(t) = f(\vec{x}_n, t)$. The delay and sum beamformer consists

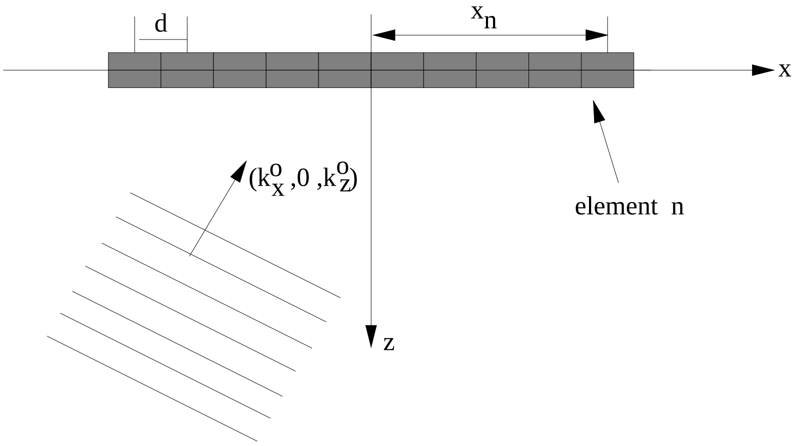


Figure 4.1: *An N-element linear array with inter-element spacing d. A monochromatic plane wave with wave number* $\vec{k}^o = (k^o_x, 0, k^o_z)$ *is propagating towards the array.*

of applying a delay Δ_n to the output of each element, then summing the resulting signals. The output of delay and sum beamformer is

$$z(t) = \sum_{n=0}^{N-1} w_n y_n(t - \Delta_n)$$

The amplitude weighting w_n is sometimes referred to as shading, taper or apodization, and can be any value between 0 and 1.

Assume that the wavefield is a monochromatic plane wave with temporal frequency ω^o and wave number $\vec{k}^o$, we have

$$y_n(t) = f(\vec{x}_n, t) = e^{j(\omega^o t - \vec{k}^o \vec{x}_n)} \tag{4.2}$$

Since 1-D array is used, some notations can be simplified. In this case, $\vec{x}_n = x_n$ and $k^o_x = k^o \sin\theta$, where $\vec{k}^o = (k_x, 0, k_z)$, $k^o = |\vec{k}^o|$ and θ is the direction of propagation. Eq. 4.2 can be simplified as

$$y_n(t) = e^{j(\omega^o t - k^o x_n \sin\theta)}$$

We have,

$$\begin{aligned} z(t) &= \sum_{n=0}^{N-1} w_n e^{j(\omega^o (t-\Delta_n) - k^o x_n \sin\theta)} \\ &= e^{j\omega^o t} \sum_{n=0}^{N-1} w_n e^{-j(\omega^o \Delta_n + k^o x_n \sin\theta)} \end{aligned}$$

When

$$\Delta_n = -\frac{k^o}{\omega^o} x_n \sin\theta = -\alpha^o x_n \sin\theta = -\frac{1}{c} x_n \sin\theta$$

i.e., the element delay Δ_n is chosen to look at the plane wave propagating direction, the beamformer's output reaches its maximum value of $e^{j\omega^o t}\sum_{n=0}^{N-1} w_n$.

We can steer the array's beam to an assumed propagation direction by using a set of delays $\Delta_n = -\frac{1}{c}x_n \sin\theta^o$. At this time, the beamformed signal for a plane wave propagating at an arbitrary direction θ is given by

$$z(t) = e^{j\omega^o t}\sum_{n=0}^{N-1} w_n e^{jk^o x_n(\sin\theta^o - \sin\theta)}$$

The delay and sum beamformer's response to a monochromatic wave is often called the array pattern. In this case, the array pattern is

$$H(\theta, \theta^o) = \sum_{n=0}^{N-1} w_n e^{jk^o x_n(\sin\theta^o - \sin\theta)} \tag{4.3}$$

The array pattern determines the array's directivity pattern, or beam pattern. Beam pattern is taken the same form as Eq. 4.3 with a fixed steering angle θ^o and variable beam angle θ.

The one-way far-field beam pattern of the array is then

$$W(\theta) = \sum_{n=0}^{N-1} w_n e^{jk^o x_n(\sin\theta^o - \sin\theta)} \tag{4.4}$$

Further discussion about the beam pattern is given in Sec. 4.2.

4.1.2 Depth of Focus

The depth of focus $L_F(Y\text{dB})$ is defined as the region around the focus point where the beam amplitude is within Y dB of the maximum possible value [37]. In this region, a change of the geometrical focus will have minimal effect on the beam diameter. For the 3 dB range definition, the depth of focus is

$$L_F(3\text{dB}) = 6\lambda\left(\frac{R}{D}\right)^2 \tag{4.5}$$

where R and D are the focus depth and aperture diameter, respectively.

Let FN be the ratio of focal length and aperture diameter, or FN $= R/D$, Eq. 4.5 can be rewritten as

$$L_F(3\text{dB}) = 6\lambda(\text{FN})^2 \tag{4.6}$$

The ratio FN is called the F-number.

4.1.3 Steering and Focus Delay

A ultrasonic beam generated by a linear phased array can be both focused and steered by properly delaying the transmitted and received signals. In radar and sonar systems,

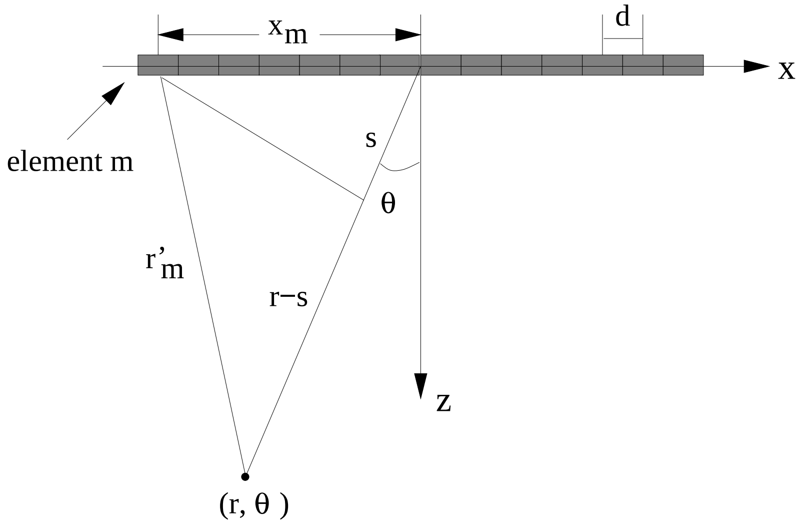


Figure 4.2: *The geometry of a phased array steering to point* (r, θ)

only steering is necessary, because targets are located in the far-field region. In ultrasound imaging systems, not only steering but also focusing is required, since spherical waves propagate in the near field region.

Consider a phased array with N elements and inter-element spacing $d = \lambda/2$. It is to be focused and steered at point (r, θ), where θ is the steering angle and r is the range. For convenience, let the origin of coordinate system locate at the center of the phased array, as shown in Fig. 4.2. The one way delay for element m is given by

$$\tau_m = (r - r'_m)/c \tag{4.7}$$

where r'_m is distance between element m and point (r, θ), $0 \leq m \leq N - 1$.

According to Law of Cosine, the distance from the point to the m'th element, r'_m, is

$$r'_m = \sqrt{x_m^2 + r^2 - 2x_m r \sin\theta} \tag{4.8}$$

where x_m is the position of the m'th element. The one way delay for the m'th element can be obtained by substituting Eq. 4.8 in Eq. 4.7 and is

$$\begin{aligned} \tau_m &= \frac{1}{c}\left(r - \sqrt{x_m^2 + r^2 - 2x_m r \sin\theta}\right) \qquad (4.9) \\ &= \tau_m^s + \tau_m^f \end{aligned}$$

where τ_m^s is the steering delay and τ_m^f is the focusing delay.

$$\tau_m^s = \frac{s}{c} = \frac{x_m \sin\theta}{c} \tag{4.10}$$

$$\tau_m^f = \frac{r - s - \sqrt{x_m^2 + r^2 - 2x_m r \sin\theta}}{c} \tag{4.11}$$

The primary function of focusing is to improve the lateral resolution. Note that focusing delay, τ^f is dependent on the point's range r. Therefore, focusing can be fixed, dynamic, or composite.

- Fixed focusing synthesizes a focus at only one particular range R. Eq. 4.11 can be rewritten as

  $$\tau_m^f = \frac{R - s - \sqrt{x_m^2 + R^2 - 2x_m R \sin\theta}}{c}$$

  A typical focus point is set in the middle of the view region. Such a focused beam has the smallest beamwidth at the range R. Away from this range, the lateral beamwidth increases.

- In dynamic focusing, the focusing delay is adjusted as function of range r. Dynamic focus on reception is commonly used so that depth of field is extended without any reduction of frame rate. Dynamically focused reception increases the focal length of the receiving beam electronically with time. The echoes returned from all depth within tissue are continuously in focus. The dynamic focusing can also be applied to transmission of beams. It can be shown that the lateral resolution of images can be further improved and the sidelobes can be further suppressed if dynamic focus is used in both transmission and reception. The two way dynamic focusing can only be used in synthetic aperture imaging which will be discussed in next chapter. In real aperture imaging however it is impractical to implement dynamic focusing on transmit as it will require an unlimited data acquisition time.

  The dynamic delay can in most older instruments only be selected at discrete values, and therefore only a discrete set of foci can be obtained. Because of the limited number of foci, the resulting beam has some irregularities, but these are so small that they have no practical effect.

- Composite focus is a special case of dynamic focus, which uses less number of foci. It increases the depth of field by transmitting several pulse sequences, each of which focuses at a different point. The beams with different focal lengths can only be transmitted one by one. The second beam must be transmitted after all echoes produced by the previous one return to the transducer. Images obtained with different transmission focal lengths are cut around their focal lengths and mounted next to each other to form a new frame of image. It is obvious that this process increases the time to form a frame of image. The frame rate is decreasing with the number of focal zones. For instance, the frame rate is N_f for one focal zone, . When the number of focal zones increases to four, the frame rate decreases to $N_f/4$. Therefore, in practice, the trade-off between the quality of images and the frame rate need to be considered.

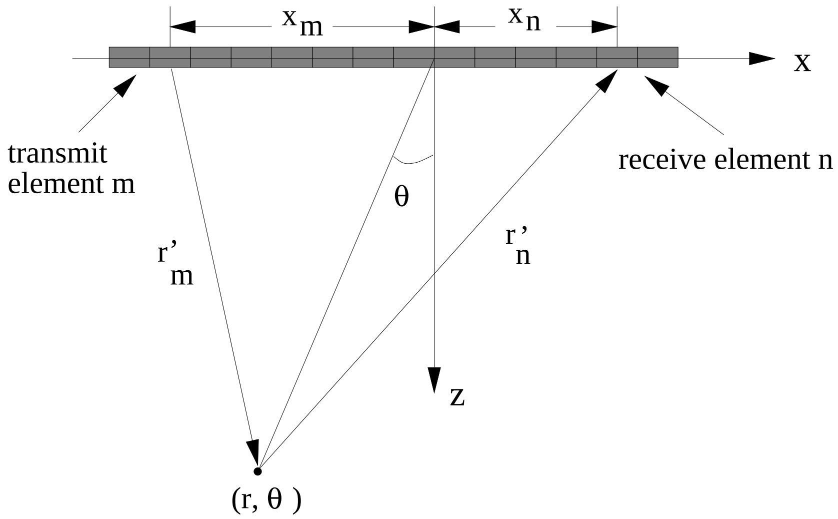


Figure 4.3: *Geometric relationship between the transmit and receive element combination and the focal point.*

4.1.4 Two Way Delay and Sum

When a short pulse is transmitted by element m and the echo signal is received by element n, as shown in Fig. 4.3, a round-trip delay is

$$\tau_{m,n} = \tau_m + \tau_n \tag{4.12}$$

where (m, n) is a transmit and receive element combination, $0 \leq m, n \leq N-1$, τ_m and τ_n can be obtained by Eq. 4.9.

For an N-element phased array, the A-scan signal, $a_{PA}(t)$ is

$$a_{PA}(t) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} y_{m,n}(t - \tau_{m,n}) \tag{4.13}$$

where $y_{m,n}(t)$ is the echo signal and $\tau_{m,n}$ is beamforming delay for the (m, n) receive and transmit element combination given in Eq. 4.12. The first and second summations correspond to transmit and receive beamforming.

For each point in an image plane, the A-scan signal can be expressed as

$$a(r, \theta) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} y_{m,n} \left(\frac{2r}{c} - \tau_{m,n} \right) \tag{4.14}$$

If the wavefield is a monochromatic plane wave with temporal frequency ω^o, the two-way beam pattern with uniform weighting is

$$W_{PA}(\theta) \;=\; \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} e^{-j\omega^o(\tau_m^s + \tau_n^s)} \tag{4.15}$$

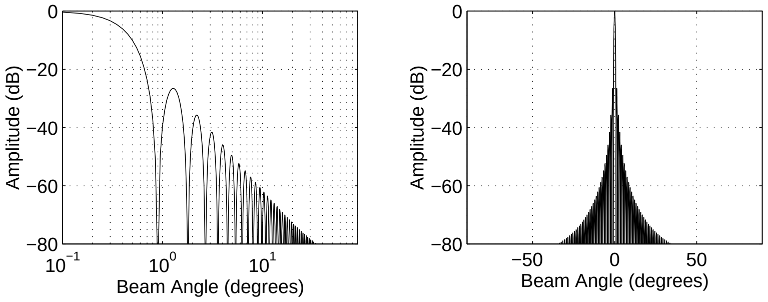


Figure 4.4: *The round trip beam pattern of phased array with uniform weighting.*

By substituting Eq. 4.10 into Eq. 4.15, we obtain

$$\begin{aligned} W_{PA}(\theta) &= \sum_{n=0}^{N-1} e^{j\frac{\omega^o}{c}x_n\sin\theta} \sum_{m=0}^{N-1} e^{j\frac{\omega^o}{c}x_m\sin\theta} \\ &= \left[\frac{\sin(k^o\frac{Nd}{2}\sin\theta)}{\sin(k^o\frac{d}{2}\sin\theta)}\right]^2 \end{aligned} \tag{4.16}$$

This equation is valid in the far field, or at the focal region of the phased array. As expected, the round trip beam pattern due to a uniformly-weighted array is sinc-squared, with the first sidelobe at -26dB, as shown in Fig. 4.4.

4.2 Beam Pattern

Using the beam pattern, we can analyze the output of beamformer for propagating waves for any directions. Beam pattern describes the quality of a beam, measured often by beamwidth and sidelobe levels. In this section, we will see how the beam pattern can be controlled by the array and beamformer design.

As discussed in the previous section, when a linear array steers to propagation direction perpendicular to the array axis, the beam pattern is

$$W(\theta) = \sum_{n=0}^{N-1} w_n e^{-jk^o x_n \sin\theta} \tag{4.17}$$

The one-way far-field beam pattern with uniform weighting is

$$\begin{aligned} W(\theta) &= \sum_{n=0}^{N-1} e^{-jk^o d(-\frac{N-1}{2}+n)\sin\theta} \\ &= e^{jk^o d\frac{N-1}{2}\sin\theta} \sum_{n=0}^{N-1} \left(e^{jk^o d\sin\theta}\right)^{-n} \end{aligned} \tag{4.18}$$

Using the equality

$$\sum_{n=0}^{N-1} a^{-n} = \frac{1-a^{-N}}{1-a} = \frac{a^{-N/2}}{a^{-1/2}} \cdot \frac{a^{N/2}-a^{-N/2}}{a^{1/2}-a^{-1/2}}$$

Eq. 4.18 is reduced to

$$W(\theta) = \frac{\sin\left(k^o \frac{N}{2} d \sin\theta\right)}{\sin\left(k^o \frac{d}{2} \sin\theta\right)} \tag{4.19}$$

when

$$k^o \frac{N}{2} d \sin\theta \neq m\pi \quad \text{for } m = \pm 1, \pm 2, \ldots$$

Otherwise we have

$$W(\theta) = 0.$$

Letting $m = 1$, we have

$$\sin\theta = \frac{\lambda^o}{Nd} \tag{4.20}$$

Eq 4.20 gives the angle of the first zero in the beam pattern of an N-element phased array. For small angles we have $\sin\theta = \theta$, thus

$$\theta_{zero} \approx \frac{\lambda^o}{Nd} \tag{4.21}$$

θ_{zero} is the main lobe which consists the most of the transmit or reflected energy. The -3dB beamwidth, giving the position where the amplitude is half of the maximal amplitude, is another often used beamwidth measure. It is approximately $0.89\theta_{zero}$.

The two way beam pattern may be found by a convolution of the transmit array with the receive array. In the Fourier transform domain it is the multiplication of the transmit beam pattern with the receive beam pattern. Since the same transducer transmits and receives sound, the aperture is convolved against itself and the beam pattern function is thus $W^2(\theta)$.

4.2.1 Sampling in Spatial and Time Domain

Ultrasound waves generated and excited by ultrasound transducers are continuous-time signal, so are backscattered signals. The backscattered signals are received and sampled by the transducer. Under reasonable constraints, continuous-time signals can be adequately represented by samples.

Nyquist sampling theorem [33] states that the sampling frequency must be chosen at least twice as fast as the fastest variation of the signals, if the signals are bandlimited. If the highest frequency component of a bandlimited signal is ω^o, the sampling frequency, ω_s must satisfy

$$\omega_s \geq 2\omega^o$$

If the sampling frequency, $\omega_s < 2\omega^o$ is chosen, the signals are undersampled and aliasing occurs. The original signals can not be recovered from the samples.

If the transducer is a discrete aperture, sampling of signals in the spatial domain has the same constraint. If the signals are bandlimited to wavenumber magnitude below

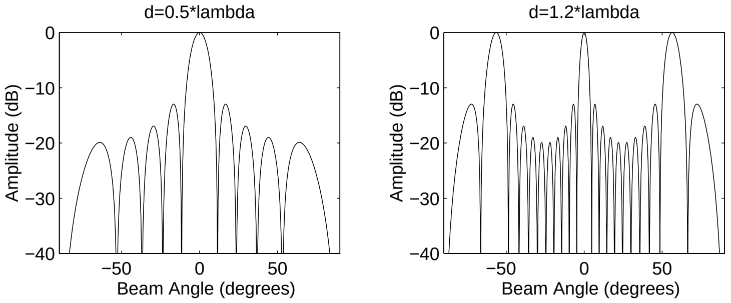


Figure 4.5: *One-way beam pattern of a 10-element linear array with* $d = \frac{\lambda^o}{2}$ *(left) and* $d = 1.2\lambda^o$*(right).*

$|\vec{k}^o| = 2\pi/\lambda^o$, then they can be sampled without loss of information as long as the spatial sampling period d is

$$d \leq \frac{\pi}{|\vec{k}^o|} = \frac{\lambda^o}{2}$$

where λ^o is the wave length. The array elements should then be separated by $\frac{\lambda^o}{2}$ units or less to avoid spatial undersampling. One manifestation of inadequate spatial sampling is the appearance of spurious mainlobe, or grating lobe in the beam pattern, as illustrated in Fig. 4.5.

4.2.2 Beam Control Methods

In general, a desirable beam pattern is the one with a narrow main lobe and sidelobes which are small compared to the main lobe. Beam control is to adjust beam pattern using certain methods. In medical ultrasound imaging, beam control methods such as dynamic focusing, dynamic aperture and apodization, give a possibility for more flexible control over the beam.

In dynamic focusing, the focusing delay varies as a function of range. The dynamic focusing gives a diffraction limited beam.

With dynamic focus, the focal diameter is much smaller for close depths than for far depths. To get a uniform width of the focus over the image field, the number of elements used in the receiver is often reduced at close depths compared to far depths. The variation of aperture with depth is called dynamic aperture. With dynamic aperture the sidelobes for near foci can be kept low at the cost of wider main lobe.

Apodization is to reduce the vibration amplitude towards the edge of the transducer face. It is used to enhance the beam shape and reduce sidelobe levels. The reduction of the amplitude is done by applying various kinds of apodization windows, such as Hamming, Hanning and cosine windows [12]. Fig. 4.6 shows the one way beam pattern of a 128-element phased array with and without apodization window. When applying a window, the sidelobe levels in the beam pattern are reduced, but the main lobe width becomes wider. There is therefore a trade-off between main lobe width and sidelobe levels.

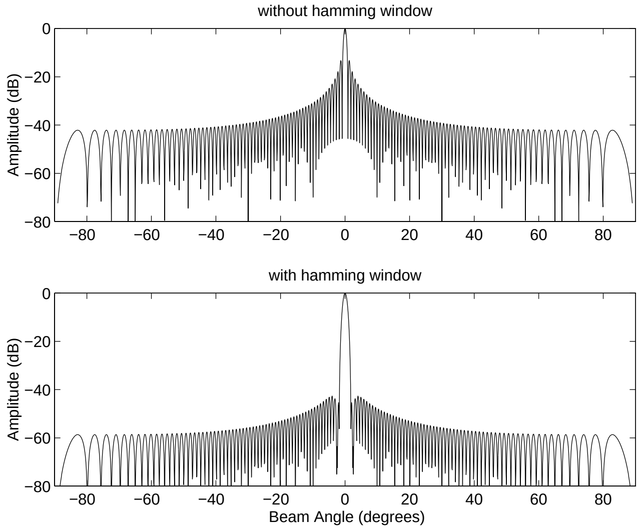


Figure 4.6: *Comparing one way beam pattern of a 128-element phased array with and without Hamming window.*

4.3 Discussion

Beamforming is an important process in many imaging systems. There are many different beamforming methods. Beamforming can be considered as a spectrum analysis process working in spatial domain. Therefore all the spectrum analysis methods developed for time signals can be used as beamforming methods. Classical beamforming methods include Fourier transform based methods and delay and sum beamforming. There are also many adaptive beamforming methods such as MUSIC and ESPRIT which give high resolution under certain assumptions.

Medical ultrasound systems operate at a short range for imaging human organs. In this case focusing is essential. The delay and sum beamforming is the only beamforming method which is suitable for focusing the beam at a close distance. Therefore, in this chapter, we only discuss this type of beamformer.

Electronical beamforming can only be done with an array. We have seen that the beam pattern depends on the array design. To obtain a narrow mainlobe, one will require a large array. However, a large array will make the system very complex. Many synthetic aperture methods have been proposed to synthesize a large aperture without increasing the system complexity. In the next chapter, we will discuss the synthetic aperture methods used for medical ultrasound imaging. In addition, a new synthetic aperture method is described. This method is to increase the imaging frame rate, and thus improve the image quality.

Chapter 5

Synthetic Aperture Beamforming

The concept of synthetic aperture was originally used in radar for high resolution imaging terrain. In this case, the cross-range resolution can be enhanced by narrowing the antenna beam width. With real antennas, this requires enlarging the physical antenna. It is often not possible because of physical constraints. The same effect can be accomplished by using synthetic aperture method. This method synthesizes a large antenna by sequentially examining the volume of interest with a small real antenna moved along a known path.

The idea of synthetic aperture has been used in ultrasound imaging systems. In this case, the benefit of the synthetic aperture has been the reduction of system complexity and cost. Several methods were proposed to form a synthetic aperture for ultrasonic imaging.

- Synthetic aperture focusing technique (SAFT) [32] [43] is a classical synthetic aperture method. At each time only a single array element acts as an active element for transmission and reception. It reduces system complexity, because only a single set of circuit for transmit and receive is needed. However it requires data memory for all data recordings.

- Multi-element synthetic aperture focusing (M-SAF) is an alternate to SAFT [22]. A group of elements transmit and receive signals simultaneously, and transmit beam is defocused to emulate a single element response. The acoustic power and the signal-to-noise ratio are increased compared to SAFT where a single element is used. response. This method requires also memory for data recordings.

- Synthetic focusing method (SF) [44] is an alternate to conventional phased array. At each time one array element transmits a ultrasound pulse and all elements receive the echo signals. The advantage of this approach is that a full dynamic focusing can be applied to the transmit and the receive, giving the highest quality of image. The disadvantage is that a huge data memory is required and motion artifacts may occur.

- Synthetic receive aperture (SRA) [44] was proposed to improve lateral resolution. At the same time, this method enables an imaging system to address a large number of transducer receive elements without the same number of parallel receive channels.

The above methods are described in details in this chapter. We also propose a new synthetic aperture method, synthetic transmit aperture (STA) [16], to increase system frame rate. This is done by splitting transmit aperture into several subapertures. At each time one subaperture transmits a ultrasound pulse and all elements receive the echo signals. The new method increases the frame rate significantly, comparing to the conventional phased array method with composite transmit focusing.

5.1 Aperture Formation

In conventional medical ultrasound imaging, a dense array of elements is used to generate the transmit beam and to form the receive beam, as we discussed in Chap. 4. Ultrasound imaging systems usually use from 64 to 128 transmit/receive channels. In order to reduce the cost of the system, the number of transmit/receive channels should be reduced. Reduced-channel arrays have a small number of transmit/receive channels multiplexed to a conventional dense array. Synthetic aperture is an important method to reduce the number of the channels. In this section, we will use the concept of effective aperture to investigate various synthetic aperture methods with respect to beam pattern. A transmit-receive apodization matrix is used as a tool for aperture synthesis on the reduced-channel array.

5.1.1 Effective Aperture

The concept of effective aperture has been applied to beam pattern analysis (see e.g. [27] and [4]). Let the origin of the coordinate system locate at the first element of an uniformly spaced linear array of N elements with weighting $w(m)$ $(m = 0, \ldots, N-1)$, the one way far field beam pattern is

$$W(\theta) = \sum_{m=0}^{N-1} w(m) e^{-jk^o md\sin\theta} \tag{5.1}$$

where d and k^o are the element spacing and the wavenumber, respectively. This equation can also be described as a discrete Fourier transform (DFT) of the aperture function:

$$\begin{aligned} W(k) &= \mathrm{DFT}[w(m)] \\ &= \sum_{m=0}^{N-1} w(m) e^{-j\frac{2\pi}{N}km}, \qquad k = 0, 1, \ldots, N-1 \end{aligned} \tag{5.2}$$

in which the frequency index k maps into the beam angle θ by

$$\sin\theta = \frac{k\lambda^o}{Nd}$$

where λ^o is wavelength. Since the round-trip beam pattern is the product of the transmit and receive beams

$$W_{RT}(\theta) = W_R(\theta) W_T(\theta) \tag{5.3}$$

using the DFT property, we get

$$W_{RT}(k) = DFT[w_R(m) * w_T(m)] \tag{5.4}$$

where "$*$" denotes convolution. The term inside the brackets of Eq. 5.4 is defined as the effective aperture $e(m)$

$$e(m) = w_R(m) * w_T(m) \tag{5.5}$$

The number of elements, the number of elements used in the effective aperture, n_{TR}, is

$$\begin{aligned} n_{TR} &= n_T + n_R - 1 \\ n_T &= \text{number of transmit elements} \\ n_R &= \text{number of receive elements} \end{aligned}$$

Thus, the round trip beam pattern is determined by the transmit and the receive aperture weightings. Every physical beam can be realized by forming the appropriate effective aperture.

5.1.2 Transmit-Receive Apodization Matrix

A transmit-receive apodization matrix (T/R matrix) [4] is used as a tool for finding effective aperture. T/R matrix is an $N \times N$ matrix of apodization values. The rows of the matrix correspond to the transmit element positions and the columns correspond to the receive element positions. The T/R matrix $\mathbf{M}$ can be factored as

$$\mathbf{M} = \vec{w}_T \vec{w}_R^t$$

where $\vec{w}_T$ and $\vec{w}_R$ are the transmit and the receive apodization vectors, respectively, and the superscript "t" denotes vector transpose. Analogously, the round trip beam can be factored into a transmit and a receive beam.

Each matrix element corresponds to a particular transmit-receive pair of array elements, with the value at that position being the product of the corresponding transmit and receive weightings. From Fig. 5.1, we observe that the n'th cross-diagonal sum of the T/R matrix, $s(n)$, is the effective aperture

$$\begin{aligned} s(n) &= w_T(0)w_R(n) + w_T(1)w_R(n-1) + \cdots + w_T(n)w_R(0) \\ &= \sum_{i=0}^{N-1} w_T(i)w_R(n-i) \\ &= w_T(n) * w_R(n) \\ &= e(n) \end{aligned}$$

where $0 \leq n \leq 2(N-1)$. The effective aperture is related to the T/R matrix through the cross-diagonal sum. For 1-D array of N elements, the effective aperture has length $2N-1$. All matrix elements that fall on a particular cross-diagonal line sum together to form the value of the effective aperture at a single position. The center position of the

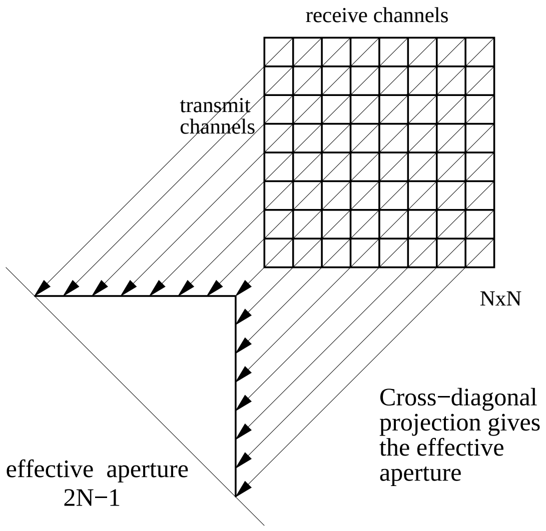
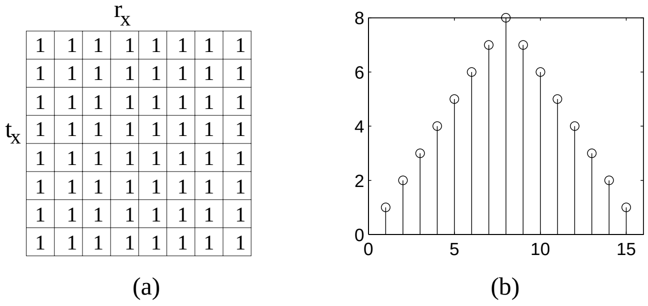


Figure 5.1: *Relationship between T/R matrix and effective aperture.*



1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

(a)

(b)

Figure 5.2: *A PA contains* 8 *elements: (a) the T/R matrix (b) the effective aperture*

effective aperture has the largest number of contributors which is N, with monotonically decreasing number of contributors on both sides until the two end positions each of which have only one contributor.

The T/R matrix with binary entries (on/off) was used in [22] and [4] simply as a table to indicate transmit-receive pairs that are used in imaging. For an N-element PA with uniform weighting, the T/R matrix is uniformly filled and the resulting effective aperture is the triangular function illustrated in Fig 5.2. The triangular weighting function is sometimes called a Bartlett weighting. In this thesis the T/R matrix is modified to analyze synthetic apertures. The value of each matrix element will indicate how many times the transmit-receive pair has been used, e.g. a matrix element 2 means the transmit-receive pairs is used twice, 1 means used once, and 0 (blank) means not used. Obviously, the transmit and the receive elements can be arbitrarily weighted. In this case, the (m, n) elements of the T/R matrix is $bw_t(m)w_r(n)$, where b is the number of times the transmit-receive pair is to be used.

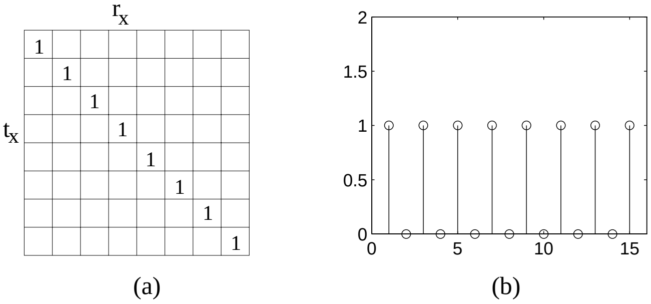


Figure 5.3: *An 8-element SA synthesized by using SAFT (a) its T/R matrix (b) its effective aperture.*

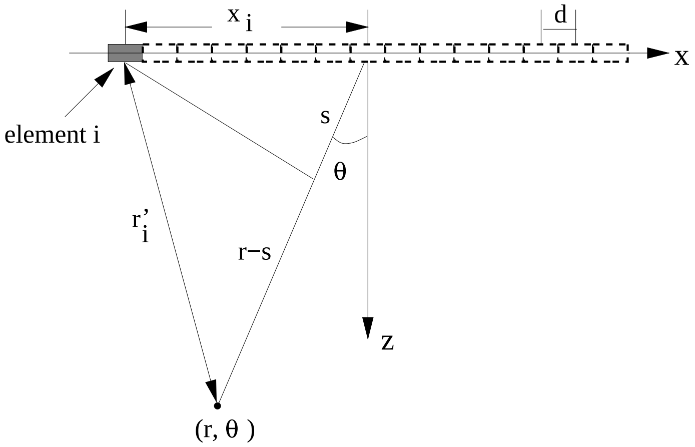


Figure 5.4: *SAFT: a single active element synthesizes an N-element phased array.*

5.2 Synthetic Aperture Focusing

The concept of synthetic aperture focusing technique (SAFT) dates back to the early 1950s when it was used for high resolution terrain imaging by radar [25]. The principle of SAFT has been applied to digitized ultrasonic signals since 1970s [43]. The idea of SAFT is used in [32] to reduce cost and size of ultrasound imaging system.

SAFT is a classical aperture synthesis method, where only one single element acts as both transmitter and receiver on a full array. The active element transmits a short pulse. Signals reflected from an object are received at the same element and stored. The same process repeats at all N elements, and an N-element array is synthesized.

The T/R matrix and the effective aperture of an 8-element synthetic array using SAFT method, is shown in Fig. 5.3.

In Fig. 5.4 we can see that the round-trip delay for element i is

$$\tau_i \quad = \quad 2\frac{r - r'_i}{c} \tag{5.6}$$

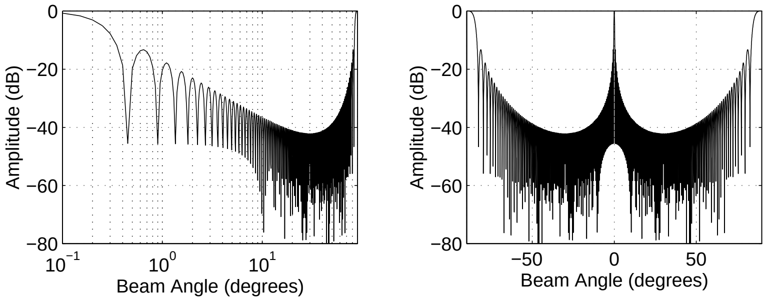


Figure 5.5: *The two-way far field beam pattern of a 128-element synthetic array synthesized by using SAFT with element spacing* $\lambda/2$. *Steering angle is* $0°$.

where

$$r'_i = \sqrt{x_i^2 + r^2 - 2x_i r \sin\theta}$$

The steering delay is

$$\tau_i^s = \frac{1}{c} 2x_i \sin\theta$$

For each image point (r, θ), the A-scan signal is

$$\begin{aligned} a_{SAFT}(r,\theta) &= \sum_{i=0}^{N-1} s_i\left(\frac{2r}{c} - \tau_i\right) \\ &= \sum_{i=0}^{N-1} s_i(t - \tau_i) \end{aligned}$$

where $s_i(t)$ is the echo signal received at the i'th element and τ_i is the beamforming delay for the i'th active element given in Eq. 5.6.

The round-trip far-field beam pattern is

$$\begin{aligned} W_{SAFT}(\theta) &= \sum_{i=0}^{N-1} e^{-j\omega^o \tau_i^s} \\ &= \sum_{i=0}^{N-1} e^{-jk^o 2x_i d \sin\theta} \\ |W_{SAFT}(\theta)| &= \frac{\sin(kNd\sin\theta)}{\sin(kd\sin\theta)} \end{aligned} \tag{5.7}$$

The round-trip beam pattern for steering angle 0° is shown in Fig. 5.5.

From Fig. 5.5, we can observe that the grating lobes appear at $\theta = \pm 90°$. The reason is that the effective aperture (see Fig. 5.3(b)) is interspersed with zeros and the inter-element distance becomes λ which violates the Nyquist sampling theorem. In SAFT, only a single element is used for transmit and receive at each time. A full array aperture is synthesized after all the elements have transmitted and received signals. The advantage of SAFT is that the circuitry for only one single transmit and one single receive is necessary.

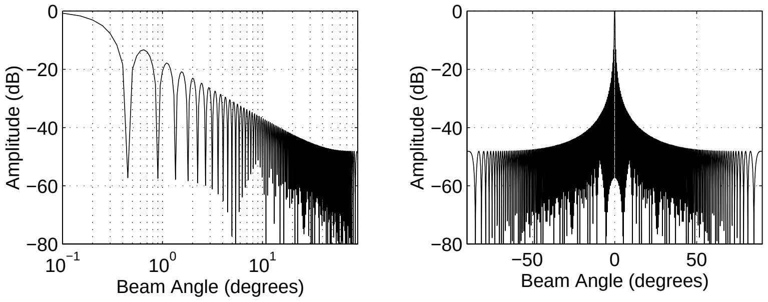


Figure 5.6: *The two-way far field beam pattern of a 255-element synthetic array synthesized by SAFT with element spacing $\lambda/4$. Steering angle is $0°$.*

The cost for the circuitry is thus reduced. Higher spatial resolution is possible since full dynamic focusing is performed on both transmit and receive. Unfortunately, it has also some disadvantages:

1. Low SNR due to the small active aperture (single element).

2. Poor contrast resolution due to higher side lobe levels compared to a conventional PA system and grating lobes occurs because of spatial undersampling.

If $\lambda/4$ element spacing is used instead of $\lambda/2$ spacing, the grating lobes can be completely removed (see Fig. 5.6). Unfortunately, such a system will have about twice number of elements.

5.3 Multi-Element Synthetic Aperture Focusing

Since in SAFT only a single element is fired on each measurement, the acoustic power delivered to the body can be very small. For increasing acoustic power, a group of elements is fired simultaneously. This leads to the multi-element synthetic aperture focusing (M-SAF) method [22].

In M-SAF, a K_t-element transmit subaperture sends an ultrasound pulse and echo signals are recored at a K_r-element receive subaperture. At the next step, one element is dropped and a new element is included to the transmit and receive subaperture, repeating the transmit and receive process. The principle behind this is illustrated in Fig. 5.7.

Elements in the transmit subaperture are fired simultaneously, and a transmit beam is formed. Since the transmit subaperture is used to emulate a high-power, single element transmitter, the transmit beam is similar to a single element response, and a defocusing lens on the transmit subaperture is thus required. A parabolic defocusing lens can be realized using the following delays

$$\delta_m = \frac{1}{c}\frac{x_m^2}{2z_d}$$

where x_m is the distance from the transmit subaperture center to the m'th element, and z_d is the distance from the transmit subaperture to the "defocal" point from the

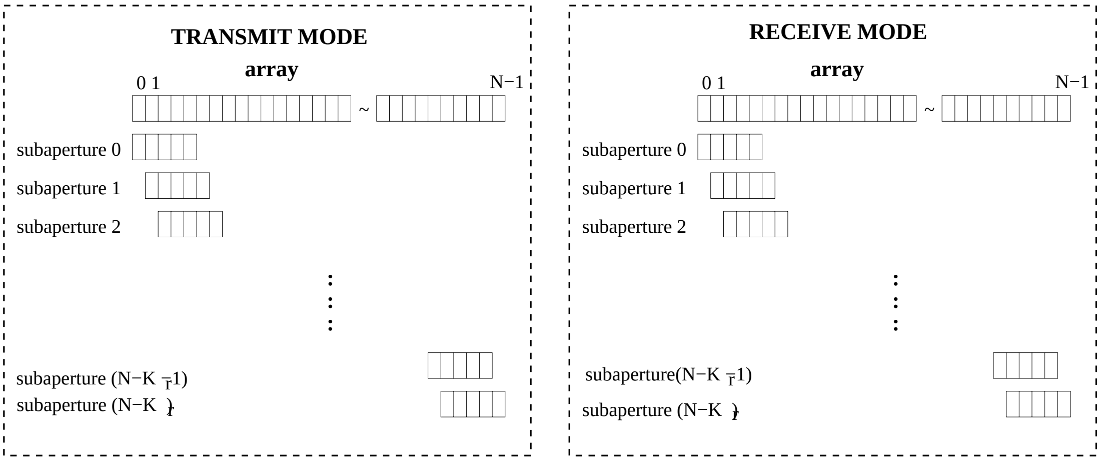
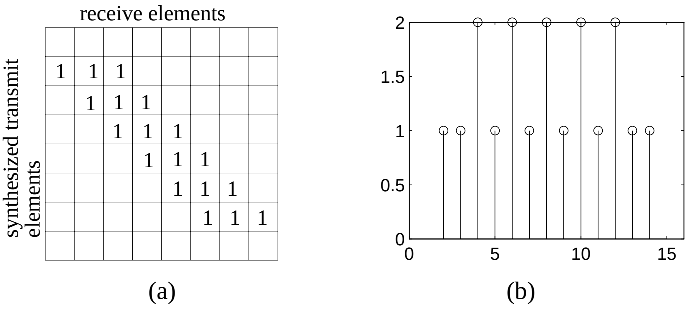


Figure 5.7: *Data acquisition strategy for M-SAF method.* $K_t = K_r = 5$.



1	1	1					
	1	1	1				
		1	1	1			
			1	1	1		
				1	1	1	
					1	1	1

Figure 5.8: *An 8-element SA synthesized by using M-SAF method with* $K_t = K_r = 3$ *(a) its T/R matrix (b) its effective aperture.*

transmit subaperture. For a 90° sector image format, z_d is chosen as $dK_t/2$ to produce a transmit beam as wide as the full sector image. If $K_t \leq 12$, the synthesized response for simple defocusing is very close to the single element response. When an N-element array is synthesized by transmit subapertures, the effective size of the synthetic aperture is $N - K_r + 1$. The T/R matrix and the effective aperture are shown in Fig. 5.8.

For an image point (r, θ), the receive delay for element n on the i'th subaperture is

$$\tau_{n+i} \quad = \quad \frac{1}{c}\left(r - \sqrt{x_{n+i}^2 + r^2 - 2x_{n+i} r \sin\theta}\right)$$

where x_{n+i} is the distance between the element and the aperture center.

We note that the receive delay of the n'th element of the i'th subaperture is the same of that of the i'th element of the n'th subaperture. We therefore can use the notation τ_{n+i} for above cases. Because all elements on the transmit subaperture synthesize a single element located at the subaperture center, the transmit delay for all element on the i'th subaperture is the same and can be calculated as

$$\tau_i = \frac{1}{c}\left(r - \sqrt{x'(i)^2 + r^2 - 2x'(i)r\sin\theta}\right)$$

where $x'(i)$ is the distance from the center of the i'th transmit subaperture to the center of the N-element synthetic aperture, and

$$x'(i) = d\left(-\frac{N-K_t}{2} + i\right)$$

For each image point (r, θ), the A-scan signal is

$$a_{M-SAF}(r,\theta) = \sum_{i=0}^{N-K_r}\sum_{n=0}^{K_r-1}\sum_{m=0}^{K_t-1} s_{i+n,i+m}\left(\frac{2r}{c} - \tau_{n+i,i} - \delta_m\right) \tag{5.8}$$

where the round trip delay, $\tau_{n+i,i}$ is $\tau_{n+i} + \tau_i$.

The two-way far-field beam pattern is

$$\begin{aligned} W_{M-SAF}(\theta) &= \sum_{i=0}^{N-K_r}\sum_{n=0}^{K_r-1} e^{-j\omega^o(\tau^s_{n+i}+\tau^s_i)} \\ &= e^{jk^o\frac{N-1}{2}d\sin\theta} e^{jk^o\frac{N-K_r}{2}d\sin\theta}\sum_{i=0}^{N-K_r} e^{-jk^o i2d\sin\theta}\sum_{n=0}^{K_r-1} e^{-jk^o nd\sin\theta} \qquad (5.9) \end{aligned}$$

Note that the third summation in Eq. 5.8 is dropped in computing the beam pattern, because the transmit subaperture is defocused to emulate a single element and thus can be seen as one single element. Amplitude of the beam pattern is

$$|W_{M-SAF}(\theta)| = \frac{\sin(k^o(N-K_r+1)d\sin\theta)}{\sin(k^o d\sin\theta)} \cdot \frac{sin(k^o K_r\frac{d}{2}\sin\theta)}{\sin(k^o\frac{d}{2}\sin\theta)} \tag{5.10}$$

The first term in W_{M-SAF} is the same as the beam pattern of SAFT with $(N - K_r + 1)$ elements. The second term is the result of a K_r element receive subaperture.

From Fig 5.9, we can see that the grating lobes are suppressed and reduced to about -45 dB. This is much low than that of the synthetic aperture synthesized by using SAFT method. Compared to SAFT, M-SAF method not only increases acoustic power, but also suppresses the grating lobes.

5.4 Synthetic Receive Aperture

The lateral resolution of phased array ultrasonic imaging systems can be improved by increasing the transducer array size, which necessitates an increase in the number of coherent transmit and receive elements in the array. It is relatively simple and inexpensive to increase the number of transmit channels, but increasing the number of parallel receive channels will significantly increase the complexity, cost and power consumption of

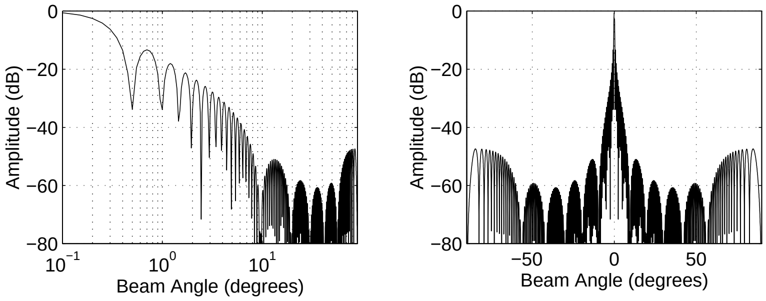


Figure 5.9: *The two-way far-field beam pattern of an SA synthesized by using M-SAF method with* $N = 128$ *and* $K = 12$.

an imaging system. Trahey *et al* [44] proposed a synthetic receive aperture method (SRA) which increase the number of effective receive elements without increasing the number of parallel receive channels. A real-time SRA imaging system was later constructed [48].

In the receive mode the aperture is split into two or more subapertures. Each subaperture contains a different subset of the available transducer elements. In order to form each line of image data in the SRA system, the transmitters must be fired once for each receive subaperture. For a single transmit pulse (from all transmit elements), the RF sum for one receive subaperture is formed and stored in memory. Then a second identical pulse is transmitted in the same direction and the RF sum for another subaperture is formed and stored. After the RF signals have been acquired from all receive subapertures, the total RF sum is formed by coherently adding together the sums from various subapertures. This algorithm enables an imaging system to address a large number of transducer receive elements without a corresponding large number of parallel receive channels.

For an N-element linear array, receive aperture is split into $N_s = N/K_r$ subapertures, and each subaperture contains K_r elements. The transmit delay for the m'th element is given in Eq. 4.7, and the receive delay for the n'th element in the i'th receive subaperture is

$$\tau_{iK_r+n} = \frac{(r - r'_{i,n})}{c}$$

where

$$r'_{i,n} = \sqrt{x^2_{iK_r+n} + r^2 - 2x_{iK_r+n} r \sin\theta}$$

For each image point (r, θ), the A-scan signal is

$$\begin{aligned} a_{SRA}(r,\theta) &= \sum_{m=0}^{N-1} \sum_{i=0}^{N_s-1} \sum_{n=0}^{K_r-1} s_{iK_r+n,m}\left(\frac{2r}{c} - \tau_{iK_r+n} + \tau_m\right) \\ &= \sum_{m=0}^{N-1} \sum_{i=0}^{N_s-1} \sum_{n=0}^{K_r-1} s_{iK_r+n,m}(t - \tau_{iK_r+n} + \tau_m) \end{aligned}$$

where $s_{iK_r+n,m}(t)$ is the echo signal.

It is known that the lateral resolution can be improved by increasing array length. In practice, it is not very expensive to build a large transmit aperture, but is very complex to form a large receive aperture. This method uses a large transmit aperture and enables us to obtain a large receive aperture using synthetic aperture method.

5.5 Synthetic Focusing

Conventional PA imaging system fires all elements on every excitation. Transmit beam is prefocused and the depth of field is hence limited. Therefore, images obtained blur out of focus. As an alternate to conventional PA imaging, one can use a synthetic focusing method (SF) [44] to extend the depth of field. In this method, an aperture is synthesized by using multiple firings. On every firing, a single element acts as a transmitter and all elements as receivers. For an N-element array, $N \times N$ independent recordings are required to synthesize an N-element PA on both transmit and receive. All data recordings must be stored in memory and then focused synthetically by a computer.

Conceptually, a SF system has all the characteristics of a conventional PA. The T/R matrix and the effective aperture of SF system is the same as those of PA. Practically, it has some advantages and disadvantages compared to a standard PA. One advantage is that the independent transmit multiplexers are used to replace complex, parallel transmit electronics of a conventional PA. Another major advantage is that both the transmit and the receive apertures can be dynamically focused. The depth of field is extended without any reduction in frame rate. However, there are several problems associated with the synthetic focusing approach. First, the image reconstruction is computationally complex, a difficulty for real time imaging. This complexity increases with the number of elements, because the RF data must be stored for every combination of transmit and receive elements, and later recombined. Also, the motion of the tissue and of the transducer during the data acquisition will degrade the image quality. Accurate image formation requires that there is no target motion during the data acquisition, but it is practically impossible. Motion compensation is needed when N is large.

5.6 Synthetic Transmit Aperture

Most synthetic aperture ultrasound imaging techniques are used to reduce the system complexity and cost. In this section, we will describe a new synthetic transmit aperture method (STA) which increases the frame rate of a ultrasonic imaging system.

5.6.1 Method

In conventional abdominal ultrasound imaging system, real aperture beamforming is used. Since abdominal organs don't move very fast, requirement for the frame rate is not high. Therefore, we can trade frame rate for image quality. This is done by using composite focusing to increase the depth of field. In the composite focusing imaging, the whole image is divided into several focal zones. One transmission is done for each focal zone. A subimage is formed for each focal zone by focusing the transmit beam to this zone. The whole image is then obtained by combining all the subimages.

The size of each zone L_F is related to the depth of focus given in Eq. 4.6. If M zones are used, the useful range is

$$L_{FPA} = ML_F$$

As an alternative we consider a synthetic aperture imaging system where the real aperture D' is split into N_s subapertures of size $D = D'/N_s$. Each of the subapertures is prefocused and the useful image range is

$$L_{FSA} = 6\lambda \left(\frac{R}{D}\right)^2 = N_s^2 6\lambda \left(\frac{R}{D'}\right) = N_s^2 L_F \tag{5.11}$$

Thus N_s pulses in the synthetic mode give the same depth of focus as $M = N_s^2$ pulses in the composite mode. The increase in frame rate is

$$f_r = \frac{N_s^2}{N_s} = N_s$$

Fig. 5.10 shows the data acquisition strategies for the composite PA and the STA systems. STA system has several advantages compared to the composite PA system. Using the STA, dynamic focusing can be performed on the synthetic transmit aperture and the receive aperture. The images reconstructed from the STA system give the same image resolution as that from the composite PA system, with an increased frame rate. The STA system offers higher image resolution than that of the composite PA system, if the frame rate is the same. The disadvantage of the STA system is that its electronic complexity is increased.

5.6.2 Algorithm

For an N-element aperture, the transmit aperture is split into N_s subapertures, and each subaperture contains $K_t = N/N_s$ elements. The STA with $K_t = 1$ is identical to a SF system. Each subaperture transmits a prefocused beam with focus point near the middle of the region of interest. Echo signals reflected from objects are received at full receive aperture and stored in the memory. All data recordings must then be combined with dynamic focusing.

Fig. 5.11 shows the geometry of the transmission and the reception for a STA system, where (r, θ) is the point of focus. The transmit delay has two parts. The first part is the delay to focus all the elements in a subaperture towards the focusing point. The second part is the delay to focus all the subaperture dynamically. The first part of the delay for the m'th element in a subaperture is

$$\tau_m = \frac{r_s - r_t'(m)}{c}$$

where

$$\begin{aligned} r_t'(m) &= \sqrt{x_t(m)^2 + r_s^2 - 2x_t(m) r_s \sin\alpha} \\ x_t(m) &= d\left(-\frac{K_t - 1}{2} + m\right) \end{aligned}$$

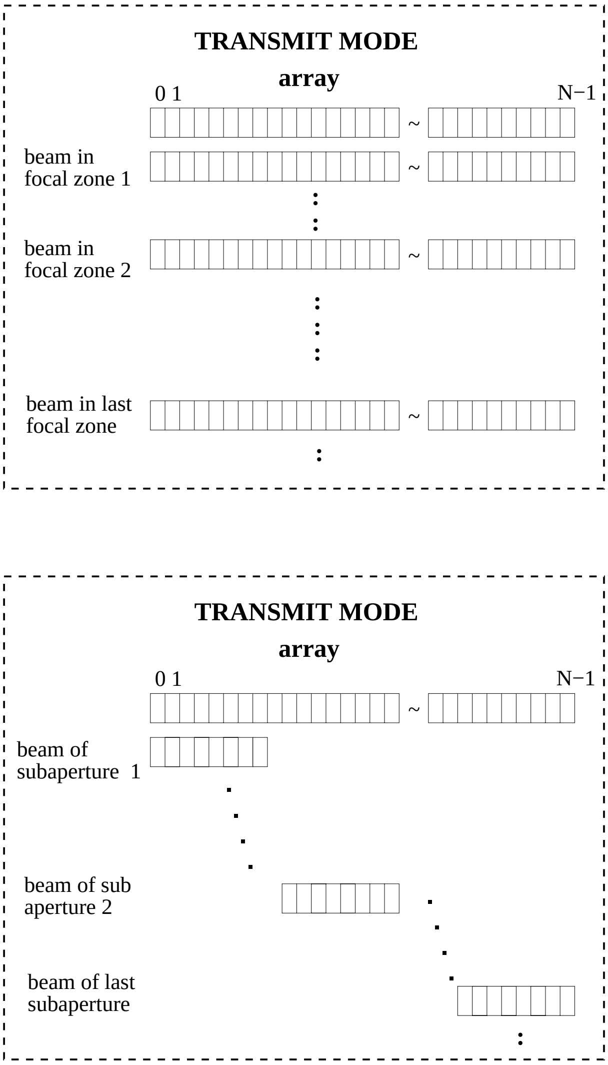


Figure 5.10: *Data acquisition strategies for composite PA (upper) and STA (bottom) systems*

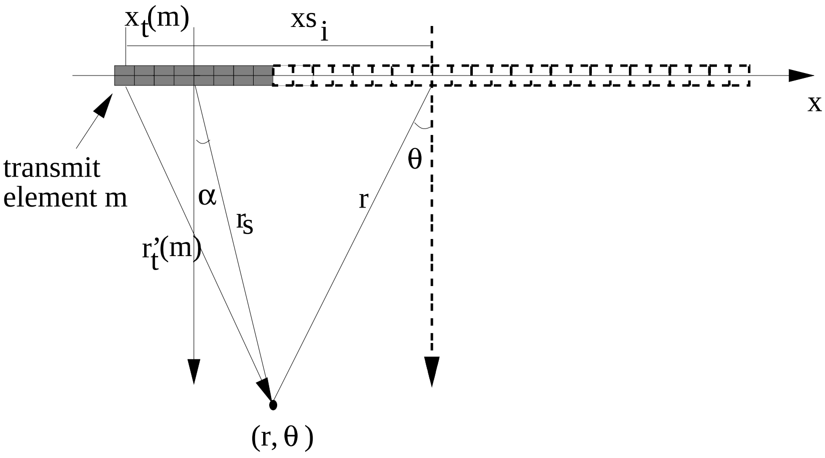


Figure 5.11: *The geometry of the transmission and the reception for STA*

and

$$\alpha = \cos^{-1}\left(\frac{r\cos\theta}{r_s}\right)$$

The second part of the delay for the i'th subaperture is

$$\begin{aligned}\tau_i &= \frac{1}{c}(r - r_s) \\ &= \frac{1}{c}\left(r - \sqrt{(r\cos\theta)^2 + (xs_i - r\sin\theta)^2}\right) \\ &= \frac{1}{c}\left(r - \sqrt{xs_i^2 + r^2 - 2xs_i r\sin\theta}\right)\end{aligned}$$

where we have

$$xs_i = K_t d\left(-\frac{N_s - 1}{2} + i\right)$$

The receive delay τ_n for the STA can be obtained in a similar way as for the PA (see Chap. 4). Thus, the total delay is $\tau_n + \tau_i + \tau_m$.

For each image point (r, θ), the A-scan signal is

$$a_{STA}(t) = \sum_{n=0}^{N-1}\sum_{i=0}^{N_s-1}\sum_{m=0}^{K_t-1} s_{K_t i+m,n}(t - \tau_n - \tau_i - \tau_m) \tag{5.12}$$

where $s_{m,n}(t)$ is the echo signal. The first summation corresponds to the receive beamforming, while the second and the third summations corresponds to transmit synthetic aperture beamforming.

The STA imaging technique assumes that targets do not move significantly between the acquisitions of data from different subapertures. If the tissue does move, data acquired from different subapertures will be out of phase and image degradation may occur. The method is obviously not suitable for imaging of moving objects such as heart.

5.6.3 Beam Pattern Analysis

The two-way far field beam pattern of a STA system is

$$\begin{aligned} W_{STA}(\theta) &= \sum_{n=0}^{N-1}\sum_{i=0}^{N_s-1}\sum_{m=0}^{K_t-1} e^{-j\omega^o(\tau_n^s+\tau_i^s+\tau_m^s)} \\ &= \sum_{n=0}^{N-1} e^{-jk^o x_n \sin\theta} \sum_{i=0}^{N_s-1} e^{-jk^o xs_i \sin\theta} \sum_{m=0}^{K_t-1} e^{-jk^o x_t(m)\sin\alpha} \end{aligned} \tag{5.13}$$

When in far field, we have $xs_i \ll r$, and $xs_i/r \approx 0$, therefore

$$r_s = \sqrt{(r\cos\theta)^2 + r^2\left(\frac{xs_i}{r} - \sin\theta\right)^2} \approx r$$

and

$$\alpha = \cos^{-1}\left(\frac{r\cos\theta}{r_s}\right) \approx \theta$$

Eq. 5.13 can then be approximated as

$$\begin{aligned} W_{STA}(\theta) &\approx \sum_{n=0}^{N-1} e^{-jk^o nd\sin\theta} \sum_{i=0}^{N_s-1} e^{-jk^o iK_t d\sin\theta} \sum_{m=0}^{K_t-1} e^{-jk^o md\sin\theta} \\ |W_{STA}(\theta)| &= \frac{\sin\left(k^o\frac{Nd}{2}\sin\theta\right)}{\sin\left(k^o\frac{d}{2}\sin\theta\right)} \cdot \frac{\sin\left(k^o\frac{N_sK_td}{2}\sin\theta\right)}{\sin\left(k^o\frac{K_td}{2}\sin\theta\right)} \cdot \frac{\sin\left(k^o\frac{K_td}{2}\sin\theta\right)}{\sin\left(k^o\frac{d}{2}\sin\theta\right)} \\ &= \left[\frac{\sin\left(k^o\frac{Nd}{2}\sin\theta\right)}{\sin\left(k^o\frac{d}{2}\sin\theta\right)}\right]^2 \end{aligned} \tag{5.14}$$

Note that the above expression of the two-way far field beam pattern is of the same form for a full phased array. Therefore, it is not necessary to have overlap between two neighboring subapertures. The overlap is used in many other synthetic aperture methods for removing grating lobes.

5.7 Discussion

In this chapter, various synthetic aperture methods for ultrasound imaging have been examined. The synthetic aperture method was first used in side scan radar imaging to form a physically impossible large aperture. A large aperture gives a narrow main lobe which again gives a high spatial resolution. The purpose of using synthetic aperture methods in medical ultrasound is different from that in radar imaging. In ultrasound imaging, a sufficiently large aperture is not difficult to obtain. However, using a large number of transmit-receive channels will increase the system complexity and the system cost. Therefore, many synthetic aperture methods have been proposed to reduce the system complexity. The system complexity reduction is achieved by using only some of the transmit-receive channels at one time and then synthesizing a full array by combining the information obtained from different transmit-receive channels.

We have also proposed a new synthetic aperture method for ultrasound imaging. This method is called the synthetic transmit aperture (STA) method which increases the frame rate and gives the same image quality as that of the composite focusing phased array system.

In this section, some important issues associated with the synthetic aperture ultrasound imaging will be discussed. Various synthetic aperture methods for ultrasound imaging will be compared. A general formula of two-way far-field beam pattern will be given.

5.7.1 Motion Compensation

In synthetic aperture radar and sonar, motion compensation is often a key step in image formation when the targets are moving or the platform is not stable [29] [6]. In ultrasound imaging, images from an synthetic aperture imaging system can be degraded by any misalignment of the received signals at different subapertures [45]. The application of synthetic aperture techniques is limited due to the tissue motion. In order to obtain good results for moving targets, motion compensation is needed. Here we will show what is the maximum number of subapertures we can use without having significant motion artifacts.

As discussed in previous sections, a synthetic aperture is synthesized by using multiple firings. The pulses from each subaperture can only be transmitted one by one, and the transmission of the pulse from one subaperture must be done after all the echoes produced by the previous transmission have returned to the transducer. Time between successive pulses (the pulse repetition time) is

$$T = \frac{2R_{max}}{c} \tag{5.15}$$

where R_{max} and c are the maximum range and the velocity of propagation, respectively.

For target motion of speed v, the distance which the target has moved during multiple firings is

$$d_t = \frac{v(s-1)}{\mathrm{PRF}} \tag{5.16}$$

where PRF is the number of pulses transmitted per second (PRF $= 1/T$), and s is the number of subapertures.

The maximum distance that the targets may move during multiple firings is equivalent to the allowable delay error in beamforming, and a rule of thumb is [14]

$$d_t \leq 0.1\lambda \tag{5.17}$$

where λ is the wavelength.

Substituting Eq. 5.16 into Eq. 5.17 and rearranging the equation, we have

$$s \leq \frac{0.1\lambda \cdot \mathrm{PRF}}{v} + 1 \tag{5.18}$$

We assume that the velocity of the tissue motion is about 20 mm/s in abdominal organs. For $\lambda = 0.44$ mm, PRF $= 5000$ Hz, the maximum number of subapertures is 12. A motion compensation is necessary if the number of subapertures is larger than this limit.

5.7.2 Sampling Requirement

In real aperture imaging, the sampling requirement to avoid grating lobes is simply that the inter-element spacing must be equal or less than half of the wave length. Similar requirement applies to the synthetic aperture imaging. However, in the case of synthetic aperture, we often need to have overlap between the neighboring subapertures.

Before we study the sampling requirement for synthetic aperture ultrasound imaging, we will have a look at the sampling requirement for synthetic aperture radar (SAR) which is a classic variant of the synthetic aperture applications.

It is common in SAR to allow the platform to travel at a known constant speed V_o along a known path. The distance between two neighboring pulses is

$$D_{pulse} = \frac{V_o}{\mathrm{PRF}}$$

where PRF is the pulse repetition frequency. Since SAR collects data when in motion, there will be Doppler shifts in the targets echoes. The total Doppler shift (in Hertz) is

$$\text{Doppler shift} = f\frac{2V_o \sin(\frac{\theta}{2})}{c} = \frac{2V_o \sin(\frac{\theta}{2})}{\lambda} \tag{5.19}$$

where the $\theta/2$ is the -3 dB half width. Assuming a small θ we have $\theta = \lambda/D$, where D is the length of the real aperture. Substitute into the above equation we have

$$\text{Doppler shift} = \frac{V_o}{D} \tag{5.20}$$

Radars which recover all the information from the Doppler shift must sample at a rate equal to or larger than twice the highest Doppler frequency presented. In other words, the Doppler shift should be equal to or less than one half of the pulse repetition frequency, i.e.

$$\text{Doppler shift} \leq \frac{PRF}{2} = \frac{V_o}{2D_{pulse}}$$

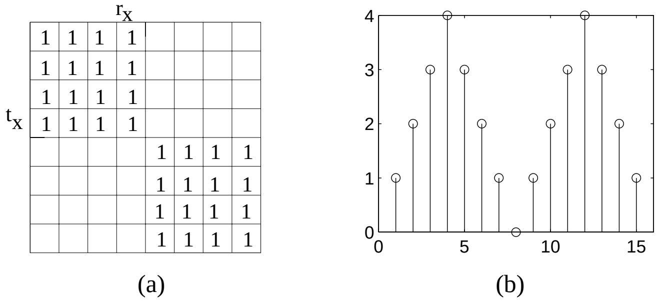
Therefore we have

$$D_{pulse} \leq \frac{D}{2} \tag{5.21}$$

It means that a pulse must be transmitted when the platform has moved a half of the real aperture length D. So the distance necessary to satisfy spatial sampling in SAR is less than, or equal to $D/2$.

Now we will examine the sampling requirement for the synthetic aperture ultrasound. The beam patterns are analyzed to see if it is necessary to have overlap between the subapertures. In order to do so, we divide the synthetic aperture methods in three groups.

Group 1
An N-element array is split into N_s subapertures. Each subaperture with $K = N/N_s$ elements acts as an active aperture. The first subaperture transmits pulses. Echo signals are received at the same subaperture and stored in memory. The same process is repeated



1	1	1	1				
1	1	1	1				
1	1	1	1				
1	1	1	1				
				1	1	1	1
				1	1	1	1
				1	1	1	1
				1	1	1	1

Figure 5.12: *An 8-element synthetic aperture with $N_s = 2$ and $K = 4$ in group 1. A synthetic aperture method in group 1 is used. (a) its T/R matrix (b) its effective aperture.*

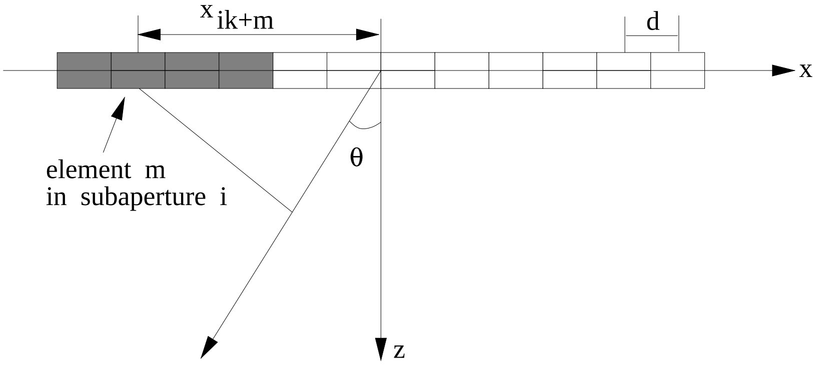


Figure 5.13: *The geometry of the synthetic aperture and the subapertures for methods in group 1.*

at all the other subapertures. A large array is then synthesized. Fig. 5.12 shows the T/R matrix of this case, using $N = 8, N_s = 2$, and $K = 4$. Synthetic aperture focusing technique ($K = 1$) and multi-element synthetic aperture focusing are two examples in this group.

As shown in Fig. 5.13, the steering delay of element m in the i'th subaperture is

$$\tau^s_{i,m} = \frac{1}{c} x_{i,m} \sin\theta x_{i,m} = \left(m - \frac{N-1}{c} \right) d + D_p$$

where D_p is the distance between two adjacent subapertures.

The two-way far-field beam pattern is

$$\begin{aligned} W(\theta) &= \sum_{i=0}^{N_s-1} \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} e^{-j\omega^o(\tau^s_{i,m}+\tau^s_{i,n})} \\ &= e^{jk^o(N-1)d\sin\theta} \sum_{i=0}^{N_s-1} e^{-jk^o i 2D_p \sin\theta} \sum_{m=0}^{K-1} e^{-jk^o md\sin\theta} \sum_{n=0}^{K-1} e^{-jk^o nd\sin\theta} \end{aligned}$$

$$|W(\theta)| = \frac{\sin(k^o N_s D_p \sin\theta)}{\sin(k^o D_p \sin\theta)} \left[\frac{\sin(k^o \frac{Kd}{2} \sin\theta)}{\sin(k^o \frac{d}{2} \sin\theta)} \right]^2 \tag{5.22}$$

where Kd is the size of the subaperture and is denoted as D. The beam pattern in Eq. 5.22 is a multiplication of two parts

$$|W(\theta)| = |W_s(\theta)||W_r(\theta)| \tag{5.23}$$

where

$$\begin{aligned} |W_s(\theta)| &= \frac{\sin(k^o L \sin\theta)}{\sin(k^o D_p \sin\theta)} && (5.24) \\ |W_r(\theta)| &= \left[\frac{\sin(k^o \frac{D}{2} \sin\theta)}{\sin(k^o \frac{d}{2} \sin\theta)}\right]^2 && (5.25) \end{aligned}$$

and $L = N_s D_p$.

When $D_p = D$, the beam patterns given by Eq. 5.24 and 5.25 are shown in Fig. 5.14(a) and 5.14(b), respectively. Fig. 5.14(c) shows the two-way beam pattern obtained by multiplying these two beam patterns. The train of the sinc pulses in Fig. 5.14(a) are the main and the grating lobes of a discrete array of length L in the visible region ±90°. Fig. 5.14(a) and 5.14(b) shows that if the grating lobes of the beam pattern in Fig. 5.14(a) are placed at the zeros of the beam pattern in Fig. 5.14(b), the grating lobes will disappear after multiplying the two beam patterns. This can be done by letting the denominator of Eq. 5.24 equal to the numerator of Eq. 5.25, i.e.

$$\sin(k^o D_p \sin\theta) = \sin\left(k^o \frac{D}{2} \sin\theta\right) \tag{5.26}$$

Simplifying the equation, we get

$$D_p = \frac{D}{2} \tag{5.27}$$

which means that a 50% overlap between the subapertures will remove the grating lobes. We can also have

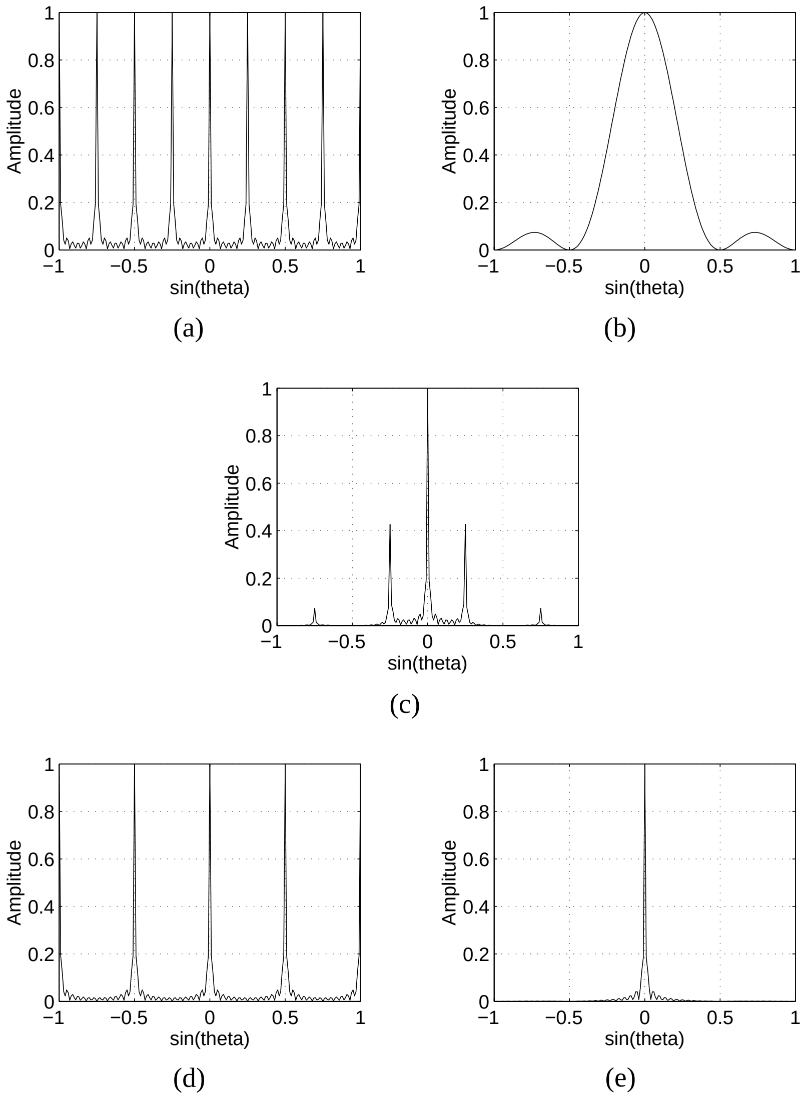
$$D_p < \frac{D}{2} \tag{5.28}$$

In this case, the grating lobes will not be zeroed, but will be suppressed by the side lobes of the real aperture beam pattern. From the above discussion, we conclude that the requirement for removing the grating lobes from the synthetic aperture beam pattern is

$$D_p \leq \frac{D}{2} \tag{5.29}$$

which is the same as the sampling requirement in SAR. When a 50% overlap between the subapertures is used, the number of the subaperture is $N_s' = 2N/K - 1$. In this case, the two-way far-field beam pattern can be found by substituting $N_s' = 2N/K - 1$ and $D_p = D/2$ into Eq. 5.23. It gives

$$|W(\theta)| = \frac{\sin(k^o N_s' \frac{Kd}{2} \sin\theta)}{\sin(k^o \frac{Kd}{2} \sin\theta)} \left[\frac{\sin(k^o \frac{Kd}{2} \sin\theta)}{\sin(k^o \frac{d}{2} \sin\theta)}\right]^2 \tag{5.30}$$



(a)

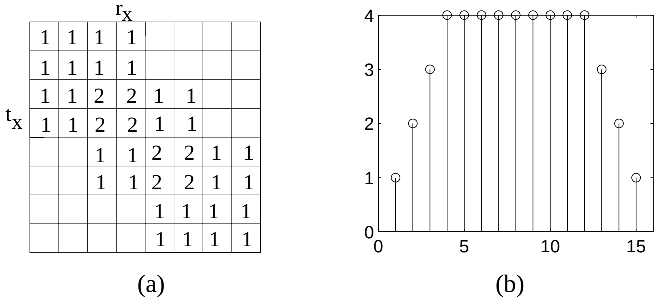
(b)

(c)

(d)

(e)

Figure 5.14: *Beam patterns of synthetic apertures and a real aperture with* $N = 128$*,* $K = 4$*,* $N_s = 32$ *and* $N_s' = 2N/K - 1$*. (a) beam pattern computed from Eq. 5.24 with* $D_p = D$*. (b) beam pattern computed from Eq. 5.25. (c) result of multiplying (a) and (b). (d) beam pattern computed from Eq. 5.30. (e)result of multiplying (d) and (b) .*



1	1	1	1				
1	1	1	1				
1	1	2	2	1	1		
1	1	2	2	1	1		
		1	1	2	2	1	1
		1	1	2	2	1	1
				1	1	1	1
				1	1	1	1

Figure 5.15: *An 8-element synthetic aperture with $N_s' = 3$ and $K = 4$, using 50% overlap between subapertures. (a) its T/R matrix (b) its effective aperture*

Fig. 5.14(d) shows the first term in Eq. 5.30 and Fig. 5.14(e) shows the two-way beam pattern when a 50% overlap is applied. Clearly, the grating lobes are removed with the overlap.

Fig. 5.15(a) shows the T/R matrix of an 8-element synthetic aperture using a 50% overlap between the subapertures. Its corresponding effective aperture (see Fig. 5.15(b)) is not interspersed with zeros and therefore removes grating lobes in the synthetic aperture beam pattern.

Group 2

A receive aperture with N elements is split into N_s subapertures. Each receive subaperture consists of $K_r = N/N_s$ elements. A transmit pulse (from all transmit elements) is excited at one direction (the first beam) for the first receive subaperture. After the echo reception, an identical pulse (the first beam) is excited for the second receive subaperture. This process is repeated for all the receive subaperture and so on. After the data acquisition from all receive subapertures, a pulse is excited at a new direction (the second beam) for the first receive subaperture and a similar process as for the first beam again. After the last beam is excited for all the receive subapertures, the whole data set is collected. The principle of data acquisition strategy is illustrated in Fig. 5.16. The T/R matrix for this case using $N = 8, N_s = 2$, and $K_r = 4$ is illustrated in Fig. 5.17. The sum of the two T/R matrixes is the same as that of a full phased array with uniform weightings. Synthetic receive aperture method belongs to this group.

In this case, the steering delay for transmit element m is

$$\tau_m^s = \frac{x_m \sin\theta}{c}$$

and the steering delay for receive element n in subaperture i is

$$\tau_{iK_r+n}^s = \frac{x_{iK_r+n}}{c}$$

where the number of receive element in each subaperture is $K_r = N/N_s$ and x_m and x_{iK_r+n} are the distance from an element to the array center. The two-way far-field beam

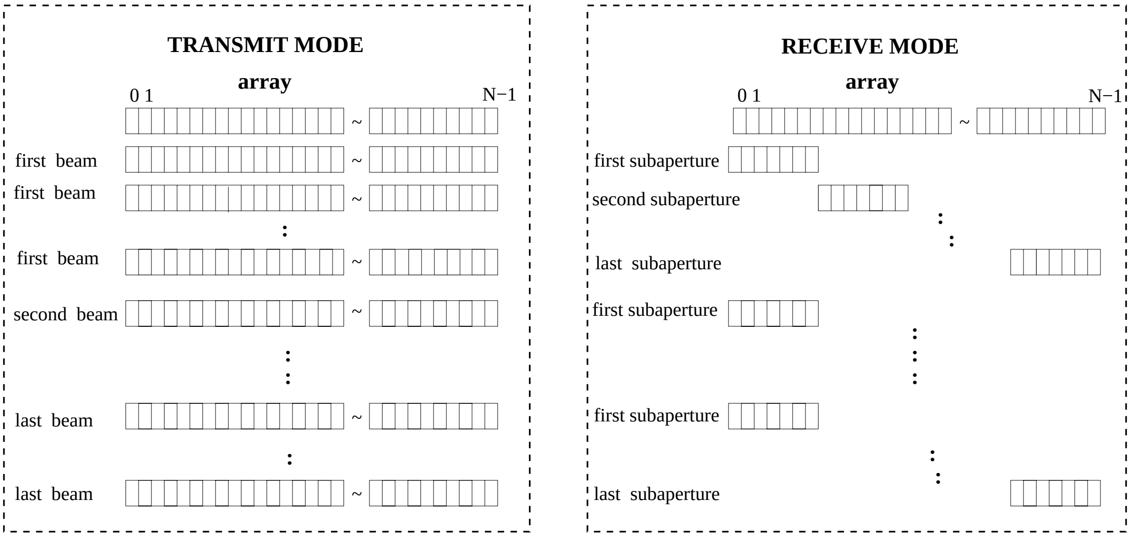
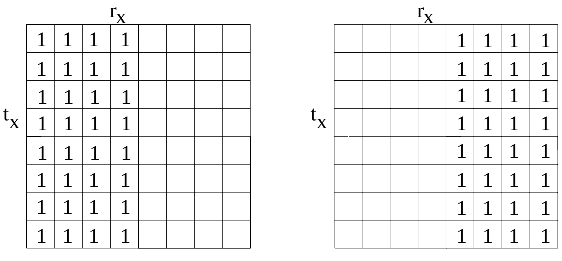


Figure 5.16: *Data acquisition strategy for synthetic aperture methods in group 2.*



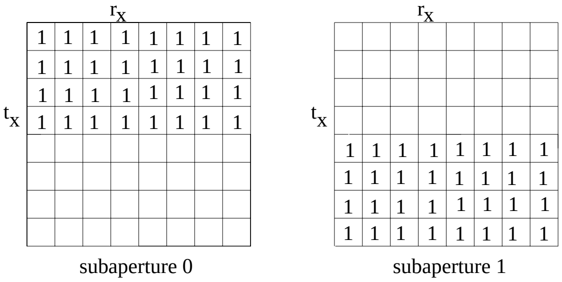
	r_x							
	1	1	1	1				
	1	1	1	1				
	1	1	1	1				
t_x	1	1	1	1				
	1	1	1	1				
	1	1	1	1				
	1	1	1	1				
	1	1	1	1				

subaperture 0

	r_x							
					1	1	1	1
					1	1	1	1
					1	1	1	1
t_x					1	1	1	1
					1	1	1	1
					1	1	1	1
					1	1	1	1
					1	1	1	1

subaperture 1

Figure 5.17: *T/R matrix of an 8-element synthetic aperture with $N_s = 2$ and $K_r = 4$, using methods in group 2.*


r_X
t_X
r_X
t_X
subaperture 0
subaperture 1

subaperture 0

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

subaperture 1

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Figure 5.18: *T/R matrix of an 8-element synthetic aperture with $N_s = 2$ and $K_t = 4$, using methods in group 3*

pattern is

$$\begin{aligned} W(\theta) &= \sum_{m=0}^{N-1}\sum_{i=0}^{N_s-1}\sum_{n=0}^{K_r-1} e^{-j\omega^o(\tau^s_{iK_r+n}+\tau^s_m)} \\ &= e^{jk^o(N-1)d\sin\theta}\sum_{m=0}^{N-1} e^{-jk^omd\sin\theta}\sum_{i=0}^{N_s-1} e^{-jk^oiK_rd\sin\theta}\sum_{n=0}^{K_r-1} e^{-jk^ond\sin\theta} \\ |W(\theta)| &= \frac{\sin(k^o\frac{Nd}{2}\sin\theta)}{\sin(k^o\frac{d}{2}\sin\theta)}\cdot\frac{\sin(k^oN_s\frac{K_rd}{2}\sin\theta)}{\sin(k^o\frac{K_rd}{2}\sin\theta)}\cdot\frac{\sin(k^o\frac{K_rd}{2}\sin\theta)}{\sin(k^o\frac{d}{2}\sin\theta)} \end{aligned}$$

Substituting $K_r = N/N_s$ into the above equation we have

$$|W(\theta)| = \left[\frac{\sin(k^o\frac{Nd}{2}\sin\theta)}{\sin(k^o\frac{d}{2}\sin\theta)}\right]^2 \tag{5.31}$$

Clearly, the two-way far-field beam pattern is the same as that of a conventional phased array. Consequently, there is no need for overlap between the receive subapertures.

Group 3

A transmit aperture with N elements is split into N_s subapertures. Each transmit subaperture consists of $K_t = N/N_s$ elements. Each subaperture transmits pulses, and the signals reflected from objects are received at all the N elements and stored in memory. This process is repeated for all the transmit subapertures, to synthesize a full array. The T/R matrix for this case using $N = 8$, $N_s = 2$, and $K_t = 4$ is shown in Fig. 5.18. Synthetic focusing ($K_t = 1$) and synthetic transmit aperture ($K_t > 1$) methods are two samples. See Sec. 5.6.3 for the beam pattern analysis of the STA method. In this case, the two-way far field beam pattern is identical to that of a full phased array. It is therefore not necessary to have overlap between the neighboring subapertures.

5.7.3 A General Formula for Beam Pattern

In the previous sections, we have discussed various aperture synthesis methods for ultrasound imaging. We know that by moving a small real aperture along a path in transmit

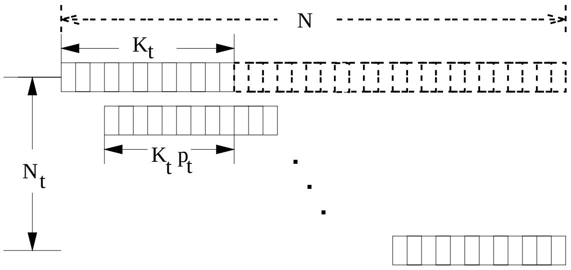


Figure 5.19: *An N-element synthetic transmit aperture is synthesized by moving a K_t-element real aperture N_t times with p_t overlap between two adjacent subapertures.*

mode, receive mode or both, a synthetic aperture can be synthesized. Here, we will give a general formula of the steering delay and the two-way beam pattern for different aperture synthesis methods. We first give the steering delay and the one way beam pattern for the transmit and the receive mode, and then examine the two way beam pattern.

Transmit Mode

A real aperture with K_t elements and inter-element distance d units in N_t locations can synthesize an N-element synthetic transmit aperture. Each time the real aperture moves $d_t = K_t(1 - p_t)d$ units distance, i.e., two neighouring subapertures have p_tK_t elements overlap, as shown in Fig. 5.19. The steering delay for the m'th element at each transmit subaperture is

$$\tau_m^s = \frac{1}{c} x_t(m) \sin\alpha$$

where $x_t(m) = d(-(K_t - 1)/2 + m)$ is the distance from the center of the subaperture to the m'th element. (See Fig. 5.11 for the definition of the angle α.)

The steering delay for the i'th synthetic transmit aperture is

$$\tau_{st}^s(i) = \frac{1}{c} x_{st}(i) \sin\theta$$

where $x_{st}(i) = d_t(-(N_t - 1)/2 + i)$ is the distance from the center of the synthetic aperture to the center of the i'th subaperture. (θ is the steering angle shown in Fig. 5.11.)

Assume that $\alpha \approx \theta$. The transmit far field beam pattern is

$$\begin{aligned} W_t(\theta) &= \sum_{i=0}^{N_t-1} \sum_{m=0}^{K_t-1} e^{-j\omega^o(\tau_m^s + \tau_{st}^s(i))} \\ &\approx \sum_{i=0}^{N_t-1} e^{-jk^o x_{st}(i)\sin\theta} \sum_{m=0}^{K_t-1} e^{-jk^o x_t(m)\sin\theta} \end{aligned} \tag{5.32}$$

where N_t is the number of subapertures given by

$$N_t = \frac{N - K_t p_t}{K_t - K_t p_t}$$

Receive Mode
A real aperture with K_r elements and inter-element distance d units in N_r different locations can synthesize an N-element synthetic receive aperture. Each time the real aperture moves $d_r = k_r(1-p_r)d$ units distance, i.e., two neighouring subapertures have $p_r K_r$ elements overlap.

The steering delay for the n'th element in the k'th subaperture is

$$\tau^s_{k,n} = \frac{1}{c} x_{k,n} \sin\theta$$

where $x_{k,n} = d(-(N-1)/2+n) + id_r$ is the distance from center of the synthetic receive aperture to the n'th element in subaperture k.

Receive far field beam pattern is

$$\begin{aligned} W_r(\theta) &= \sum_{k=0}^{N_r-1}\sum_{n=0}^{K_r-1} e^{-j\omega^o \tau^s_{k,n}} \\ &= \sum_{k=0}^{N_r-1}\sum_{n=0}^{K_r-1} e^{-jk^o x_{k,n}\sin\theta} \end{aligned} \qquad (5.33)$$

where N_r is the number of subapertures given by

$$N_r = \frac{N - K_r p_r}{K_r - K_r p_r}$$

There are two different cases, In case 1, each transmit subaperture sends a ultrasound pulse which is received by all the receive subapertures. Therefore, the index i in Eq. 5.32 and k in Eq. 5.33 are independent.

In the case, the two-way far field beam pattern is obtained by multiplying Eq. 5.32 and 5.33,

$$W_{rt}(\theta) = \sum_{i=0}^{N_t-1} e^{-jk^o x_{st}(i)\sin\theta} \sum_{m=0}^{K_t-1} e^{-jk^o x_t(m)\sin\theta} \sum_{k=0}^{N_r-1}\sum_{n=0}^{K_r-1} e^{-jk^o x_{k,n}\sin\theta}$$

and

$$\begin{aligned} |W_{rt}(\theta)| &= \sum_{i=0}^{N_t-1} e^{-jk^o i d_t \sin\theta} \sum_{m=0}^{K_t-1} e^{-jk^o m d \sin\theta} \sum_{k=0}^{N_r-1} e^{-jk^o k d_r \sin\theta} \sum_{n=0}^{K_r-1} e^{-jk^o n d \sin\theta} \\ &= \frac{\sin(k^o N_t \frac{d_t}{2}\sin\theta)}{\sin(k^o \frac{d_t}{2}\sin\theta)} \cdot \frac{\sin(k^o K_t \frac{d}{2}\sin\theta)}{\sin(k^o \frac{d}{2}\sin\theta)} \\ &\quad \cdot \frac{\sin(k^o N_r \frac{d_r}{2}\sin\theta)}{\sin(k^o \frac{d_r}{2}\sin\theta)} \cdot \frac{\sin(k^o K_r \frac{d}{2}\sin\theta)}{\sin(k^o \frac{d}{2}\sin\theta)} \end{aligned} \qquad (5.34)$$

The PA, SRA, SPA and STA methods belong to this case. Tab. 5.1 shows the important parameters of these methods and the two way beam patterns which are the special cases of that given by Eq. 5.34.

	PA	SRA	SF	STA
K_t	N	N	1	K_t
p_t	0	0	0	0
d_t	Nd	Nd	d	$K_t d$
N_t	1	1	N	N/K_t
K_r	N	K_r	N	N
p_r	0	0	0	0
d_r	Nd	$K_r d$	Nd	Nd
N_r	1	N/K_r	1	1
Two way beam pattern	Eq. 4.16	Eq. 5.31	Eq. 4.16	Eq. 5.14

Table 5.1: *Important parameters and the two way beam patterns of synthetic aperture methods of case 1.*

	PA	SAFT	M-SAF
K_t	N	1	1*
p_t	0	0	0
d_t	Nd	d	d
K_r	N	1	K_r
p_r	0	0	$(K_r - 1)/K_r$
d_r	Nd	d	d
N_s	1	N	$N_r = N - K_r + 1$
Two way beam pattern	Eq. 4.16	Eq. 5.7	Eq. 5.10

Table 5.2: *Important parameters and the two way beam patterns of synthetic aperture methods of case 2. In case of M-SAF, a single transmit element is emulated, and therefore we use $K_t = 1$ instead for the number of transmit channels.*

In case 2, the ultrasound pulse sent by each transmit subaperture is received by one corresponding receive subaperture. In index i in Eq. 5.32 and k in Eq. 5.33 are therefore dependent. In this case, the two-way far field beam pattern is

$$W_{rt}(\theta) \approx \sum_{i=0}^{N_s-1} e^{-jk^o x_{st}(i)\sin\theta} \sum_{m=0}^{K_t-1} e^{-jk^o x_t(m)\sin\theta} \sum_{n=0}^{K_r-1} e^{-jk^o x_{i,n}\sin\theta}$$

and

$$\begin{aligned} |W_{rt}(\theta)| &= \sum_{i=0}^{N_s-1} e^{-jk^o i(d_t+d_r)\sin\theta} \sum_{m=0}^{K_t-1} e^{-jk^o md\sin\theta} \sum_{n=0}^{K_r-1} e^{-jk^o nd\sin\theta} \\ &= \frac{\sin(k^o N_s \frac{d_t+d_r}{2}\sin\theta)}{\sin(k^o \frac{d_t+d_r}{2}\sin\theta)} \cdot \frac{\sin(k^o K_t \frac{d}{2}\sin\theta)}{\sin(k^o \frac{d}{2}\sin\theta)} \cdot \frac{\sin(k^o K_r \frac{d}{2}\sin\theta)}{\sin(k^o \frac{d}{2}\sin\theta)} \end{aligned} \quad (5.35)$$

where

$$N_s = \begin{cases} N_t & N_t \le N_r \\ N_r & \text{otherwise} \end{cases}$$

The SAFT and M-SAF methods belong to this case. By substituting the parameters associated to SAFT or M-SAF into Eq. 5.35, the resulting two way beam patterns are the same as those presented in the previous sections. See Tab. 5.2.

Chapter 6

Experiments and Results

In this chapter, the proposed synthetic aperture method STA, as well as the other synthetic aperture methods are investigated experimentally using RF data acquired from three different phantoms. The wire phantom, placed in a water tank, consists of six wires which lie at ranges of 34, 48, 65, 83, 101 and 121 mm. Another phantom contains diffuse scaterers and four cyst regions. They are used to test contrast resolution. Additionally, a tissue equivalent phantom, which contains wire and cyst objects as well as diffuse scatterers, is used to test image quality in general. In the rest of the thesis, these phantoms are referred to as: Wire Phantom, Cyst Phantom and Tissue Phantom.

Experimental results for the different synthetic aperture imaging techniques will be given. All the RF data sets used in the experiments were acquired by the researchers at the Biomedical Ultrasonics Laboratory, University of Michigan. A digital bandpass filter was designed. The sampled RF data are filtered with the digital bandpass filter for additional noise removal. All the beamforming and image reconstruction are carried out digitally to emulate a digital imaging system.

6.1 Data Acquisition

We used complex RF data sets acquired by the researchers at the Biomedical Ultrasonics Laboratory, University of Michigan. The data were acquired by using an experimental system with a 3.5 MHz, 128-element transducer array with 0.22 mm inter-element spacing. The inter-element space is about $\lambda/2$. All elements can be used for both transmit and receive. Each time only a single element transmits signal and all elements receive the echoes. The transmit and receive element combination gives a total of 128×128 possible RF A-scans. All these possible A-scans were recorded. Each A-scan was sampled at 13.89 MHz, and had 2048 samples corresponding to a spatial range of about 130 mm. To reduce noise imposed by the data acquisition system, signal averaging over successive acquisitions was employed. The final data were stored by using 16-bit words.

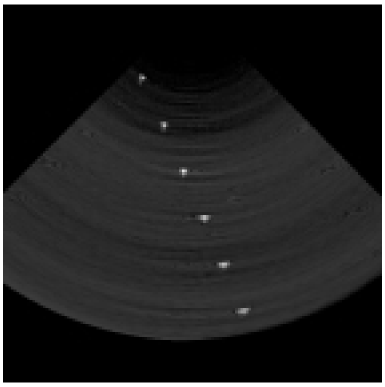


Figure 6.1: *Image reconstructed from the recorded RF data of the Wire Phantom by using a phased array beamforming with a composite transmit focusing of 16 focus zones and a dynamic receive focusing. The display dynamic range is 55 dB.*

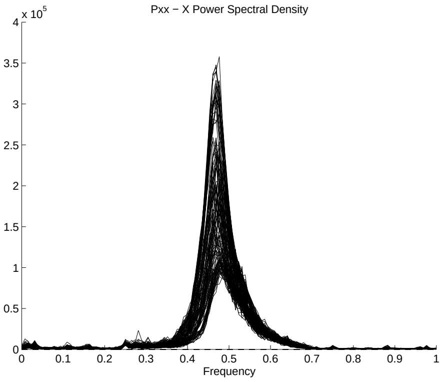


Figure 6.2: *Power spectral densities of one set of the raw data recorded from the Wire Phantom. The frequency axis is from 0 to 1. The normalized frequency 1 corresponds to half of the sampling frequency, which is 13.89 MHz in real frequency.*

6.2 Data Analysis

Using the recorded RF data, we first reconstructed images using a real aperture beamforming with a composite transmit focusing of 16 focus zones and a dynamic receive focusing. The reconstructed image from the Wire Phantom is shown in Fig. 6.1. From the figure we can see that the RF data contains considerable noise. Therefore, an additional noise removal is necessary.

The design of a proper filter to remove noise is based on the spectral distribution of the signals. The power spectral densities of a representative data set recorded from the Wire Phantom are shown in Fig. 6.2. In this figure, the frequency axis is from 0 to 1. The normalized frequency 1 corresponds to half of the sampling frequency, which is 13.89 MHz in real frequency.

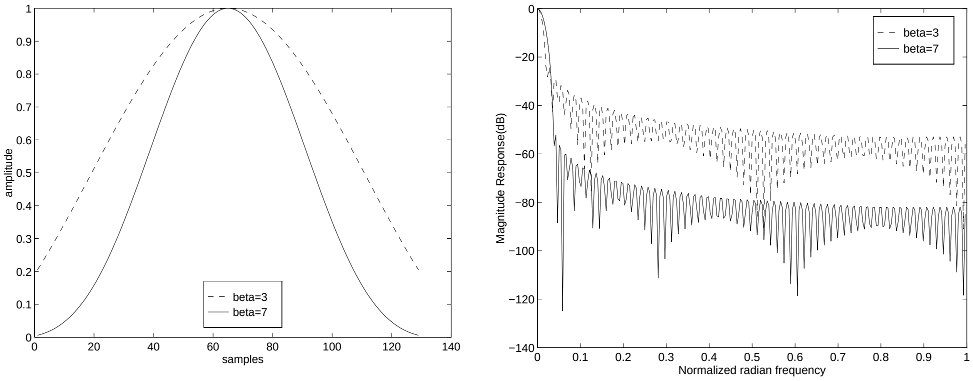


Figure 6.3: *Left: Kaiser windows with $\beta = 3$ and 7. Window size $M = 128$. Right: the corresponding Fourier transform magnitudes. The normalized frequency is from 0 to 1.*

6.2.1 Filter Design

A bandpass finite-duration impulse response (FIR) filter was implemented. The simplest method of FIR filter design is called the window method. The basic principle of the window design method is to truncate the ideal impulse response to a finite-length window. Some commonly used windows are Hamming, Hanning, Blackman, Bartlet and Kaiser window (see e.g. [33]).

The Kaiser window, used to design the FIR filter in this thesis, is defined as

$$w[n] = \begin{cases} \frac{I_0\left[\beta\sqrt{1-[(n-\alpha)/\alpha]^2}\right]}{I_0(\beta)}, & 0 \leq n \leq M \\ 0 & \text{otherwise} \end{cases}$$

where M is the window size, $\alpha = M/2$, and $I_0(\cdot)$ is the zeroth-order modified Bessel function of the first kind. By varying M and the parameter β, the window size and shape can be adjusted to trade sidelobe amplitude for mainlobe width. Fig. 6.3 shows the properties of two Kaiser windows with $\beta = 3$ and $\beta = 7$. Since low sidelobe level is preferred in ultrasound imaging, $\beta = 7$ is chosen for the FIR filter design.

From Fig. 6.2, we observe that the main energy is distributed between 0.23 and 0.67 in the normalized frequency domain, corresponding to 1.5974 MHz and 4.6532 MHz in real frequency. Thus, a digital bandpass filter with normalized cutoff frequency 0.23 and 0.67 is needed to remove noise. The resulting bandpass filter, using Kaiser window with $M = 128$ and $\beta = 7$, is illustrated in Fig. 6.4.

The spectra of the bandpass filtered signals are shown in Fig. 6.5. Clearly, the noise energy outside the desired cutoff frequency interval are suppressed. Similar bandpass filters were also used for sampled data from the other phantoms. Appendix B.2.1 gives the C-program used for the filtering.

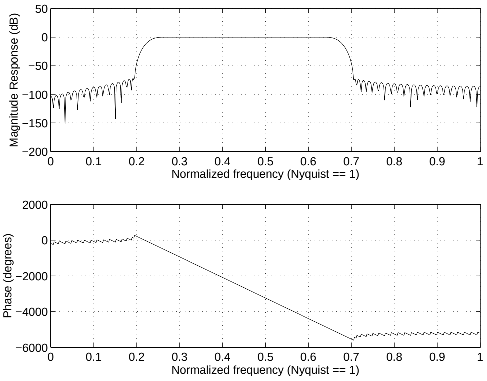


Figure 6.4: *The magnitude and the phase of the frequency response of a digital bandpass filter, generated by using* matlab *code:* `fir1(128, [.23 0.67], kaiser(129,7))`*.*

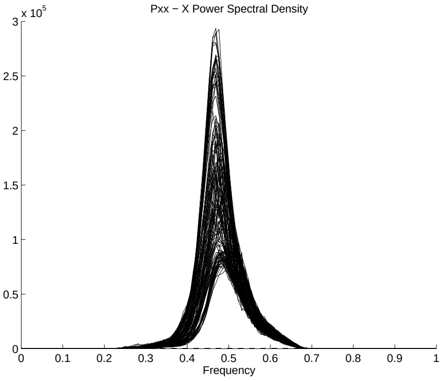


Figure 6.5: *Power spectral densities of one set of the raw data recorded from the Wire Phantom. A bandpass filtering with a Kaiser window was applied.*

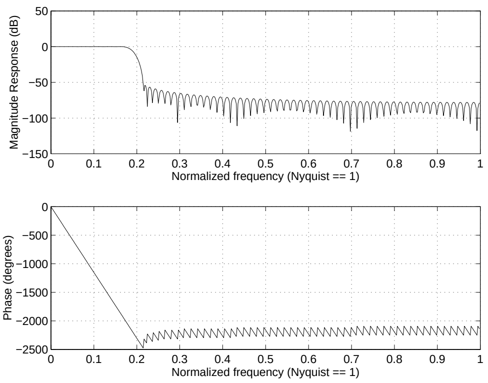


Figure 6.6: *The magnitude and the phase of the frequency response of a digital lowpass filter generated by using* matlab *code:* `fir1(128, .19, kaiser(129,5))`. *The filter is used after signal upsampling.*

6.2.2 Data Upsampling

Due to the rapid development of high speed, large bit density analog to digital converters, modern beamforming systems exploit the advantages of digital implementations. When developing a delay and sum beamformer for sampled data, one problem that might arise with a digital imaging system is the sidelobes due to the phase or timing errors [31] [2]. A solution to this problem is to greatly oversample the signal so that accurate time delay can be produced [15]. Typical interpolation factors are 2, 4 and 8.

To reduce the aberrations introduced by delay quantization, we can interpolate between the samples of the echo signals. In the conventional approach, all the echo signals are interpolated by a factor I to obtain oversampled signals. Usually it takes the following two steps:

1. Upsample the original echo signals by interspersing $I-1$ zero-valued samples between the echo samples.

2. Pass the upsampled signal through a lowpass interpolation filter having unit-sample response.

Interspersing $I-1$ zeros between the samples in the time domain causes a periodic replication of the signal spectrum in the frequency domain. Lowpass filtering ideally eliminates all of the replications except the one at baseband ($-\pi/I < \omega < \pi/I$), resulting in a spectrum that is simply a compressed version of the original. The signals now have an effective sampling rate that is I times higher than the original.

A simple delay and sum algorithm requires a fairly high sampling rate. In the experiments presented in this thesis, the sampled signals were interpolated with an interpolation factor 4. The corresponding lowpass filter had a normalized cutoff frequency 0.25π, shown in Fig. 6.6. A C-program used to oversample data is given in Appendix B.2.2.

6.3 Results and Discussions

Experiments have been done to demonstrate the performance of various synthetic aperture ultrasound imaging systems. Particularly, the new proposed method STA has been compared to the composite PA imaging systems.

To form B-scan images, beams were first formed by delay and sum beamforming algorithms with a transmit f/number of 2 and receive f/number of 1.5. The B-scan images were produced from these beams after a gain compensation, a scan conversion, and a logarithmic compression.

The beam width between two first zeros is

$$\theta_w = \frac{\lambda}{Nd}$$

where λ is the wave length, N is the array size, and d is the inter-element space. According to the Nyquist sampling theorem, the inter-element space d must be less or equal to $\lambda/2$. Let $d = \lambda/2$. We have $\theta_w = 2/N$. Again, according to the Nyquist theorem, we need to have at least two beams within a beamwidth in order to preserve all the information. Therefore, the between-beam angle is $\theta_s = \theta_w/2 = 1/N$. For $N = 128$ and a 90° sector image, the number of beam lines should be at least

$$\frac{\pi/2}{\theta_s} = \frac{\pi N}{2} = 201$$

All images presented in this thesis were reconstructed with 201 beam lines at constant increments of $\sin\theta$ over a 90° sector.

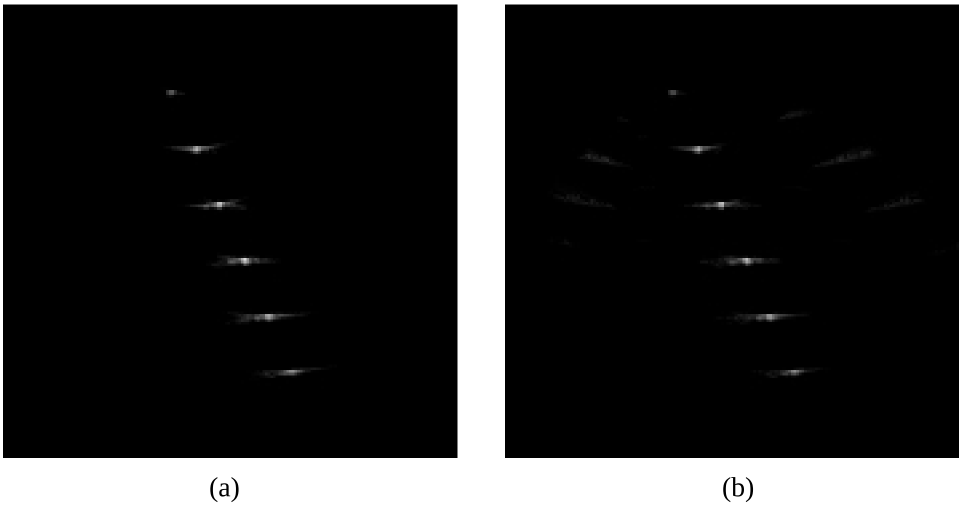
6.3.1 Effect of Data Upsampling

For signals sampled at near Nyquist rate, it is difficult to form high quality beams with low sidelobe levels using a delay and sum beamforming. One way to deal with this kind of signals is to interpolate the sampled signals by an upsampling. Fig. 6.7 shows the effect of upsampling. Image reconstructed from the original signals contains sidelobes which are visible in a 50 dB dynamic range. These sidelobes are due to the quantization errors. After an upsampling, the sidelobes are suppressed and become invisible in this dynamic range. The images shown in the figures were reconstructed by using the synthetic aperture method M-SAF with $K_r = K_t = 10$, because the sidelobes were the most clear for this method when the upsampling was not applied. Image reconstructions were also done by using other synthetic aperture and real aperture beamforming methods. The sidelobe reductions by upsampling were significant for all these methods.

The data upsampling minimizes the sidelobes due to the quantization errors. Therefore, in the rest of the experiments, echo signals were upsampled before a delay and sum beamforming algorithm was applied.

6.3.2 Effect of Focus Modes

The delay and sum beamforming can be done in different ways regarding to the focus modes. It can have a fixed transmit and receive focus (FTR), a fixed focus for transmit and



(a) (b)

Figure 6.7: *B-scan images of the Wire Phantom reconstructed by using the synthetic aperture method M-SAF with $K_r = K_t = 10$. The images are displayed over a 50 dB display dynamic range. (a) The data were upsampled with interpolation factor 4. (b) No upsampling applied.*

a dynamic focus for receive (FTDR), a dynamic focus for both transmit and receive (DTR), or a composite transmit focus and dynamic focus for receive (CTDR). Some experiments were done to shown the effect of different focus modes. When a fixed transmit focus was used, the focus point was at the depth of 65 mm from the array. When composite transmit focus mode was used, the whole image was split into 4 focal zones. Fig. 6.8 shows the B-scan sector images of the Wire Phantom displayed over a 55 dB dynamic range for different focus modes. The DTR mode is specially associated with the synthetic focusing (SF) method. Therefore, for the DTR mode, the SF method was used in the experiments. For the other modes, a real aperture phased array imaging was used. From the figure we can see that the DTR mode SF method gives the best image quality (Fig. 6.8(a)). However, it is difficult to implement this method in a real-time system. Objects in Fig. 6.8(b) were blurred out of focus point due to the limited depth of field for the FTR mode. A better image quality was achieved by using the FTDR mode. In this case, the image was degraded at the region away from the focal zone of the fixed transmit lens (Fig. 6.8(c)). Composite transmit focus can be used to reduce the effects of the fixed transmit focus. This is shown in Fig. 6.8(d) for the CTDR mode. The image quality of the CTDR mode is similar to that of the DTR mode. The advantage of the CTDR mode is that the imaging frame rate is significantly increased, compared to the DTR mode.

6.3.3 Performance of Synthetic Aperture Methods

Experiments have been done to evaluate the performance of different synthetic aperture imaging systems. The results of real aperture PA imaging are also included for comparison. As we have seen, the dynamic focus mode gives the best image quality. Therefore, the dynamic focusing receive mode were used in testing all the synthetic aperture imaging

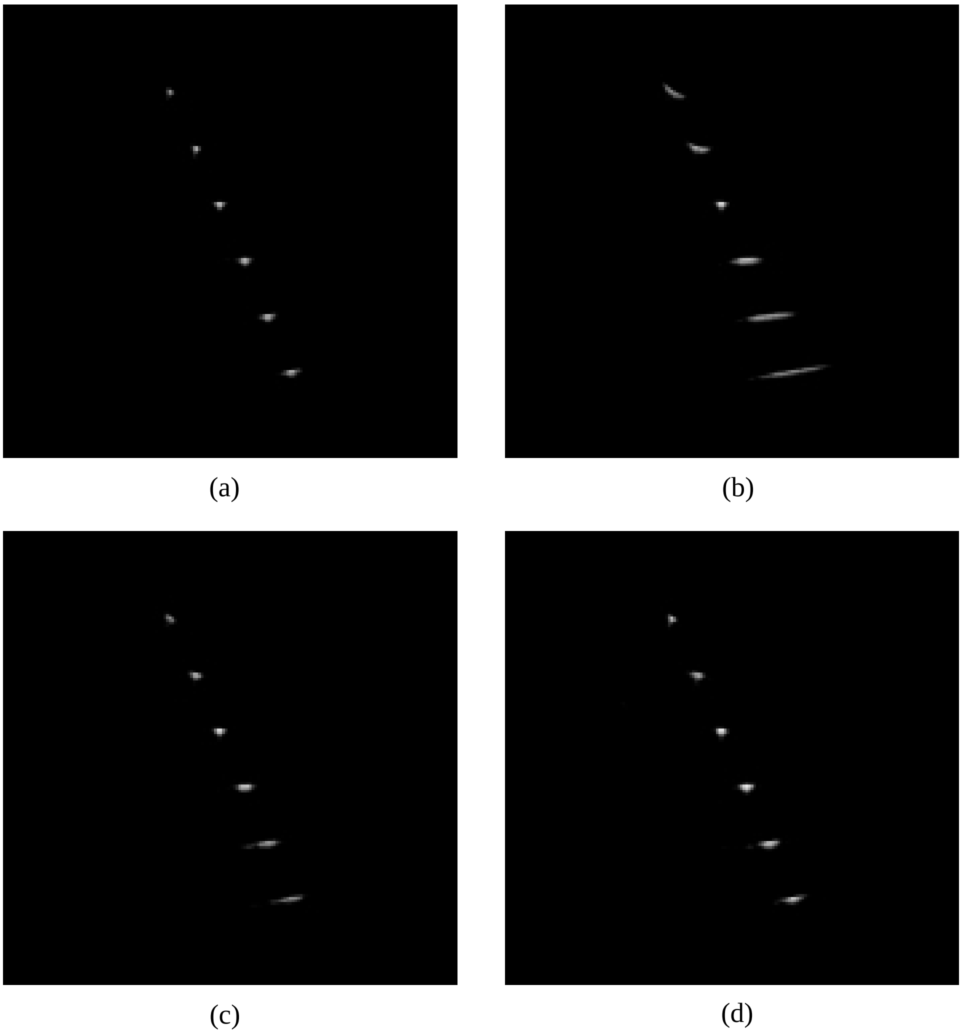


Figure 6.8: *B-scan images of the Wire Phantom formed by using different focus modes, displayed over a 55 dB display dynamic range. (a) DTR (SF) mode (b) FTR mode (c) FTDR mode (d) CTDR mode (4 focal zones)*

systems. The multiple foci extend the depth of field without any reduction in the frame rate. In the STA imaging system two transmit subapertures were used. Fig. 6.9 shows the B-scan sector images of the Wire Phantom displayed over a 45 dB display dynamic range for different synthetic aperture methods. We can see that grating lobe artifacts are presented in the SAFT image due to the undersampling in the effective aperture. As discussed earlier, the grating lobes can be removed if the array is oversampled, i.e. the inter-element space is reduced to $\lambda/4$. These grating lobes did not appear in the images of other synthetic aperture methods. The M-SAF image also exhibits a reduced sidelobe level compared to the SAFT. The sidelobes are further reduced in the SF and the STA images. A reduction in sidelobe level will increase the contrast resolution. However, the M-SAF system has a lower system complexity and cost. The SF system gives obviously the best image quality. The STA system gives the similar quality with an increased frame rate.

The performance of the new STA method was compared with the composite transmission PA system in details. Fig. 6.10 compares the B-scan sector images of the Wire Phantom obtained by using a STA system and a real aperture composite transmission PA system. In the STA system, the pulses were prefocused at 65 mm range in transmission. Although the number of transmit pulses of the STA is one half of that of the PA with the composite transmit focus, the image quality of the STA is very similar to that of the PA because of a dynamic focusing on the synthetic aperture for received data. The STA imaging system increases the image frame rate significantly, and gives a compatible image quality.

For a further comparison of these two techniques, -6 dB and -30 dB contour plots of the six wire targets are illustrated in Fig. 6.11. The figure indicates that the proposed STA method exhibits satisfactory image quality compared to the conventional PA with composite transmit focus. In this experiment, the frame rate of the STA system was four times that of the composite PA system.

When we changed the parameters so that these two systems gave the same frame rate, the STA system gave a better image quality than that of the PA system, as shown in Fig. 6.12. The reason is that the number of the focal zones for the PA system is reduced in order to have the same frame rate.

The contour plots in Fig. 6.13 confirms that the resolution increases with the number of the transmit subapertures. The STA imaging can achieve the best image quality when the number of transmit subapertures is the number of element of the full array. (In this case, the subaperture size is 1.)

Fig. 6.14 shows the B-scan images of the Cyst Phantom over a 60 dB display dynamic range. The phantom was constructed from gelatin with nylon particles used as ultrasonic scatterers at a concentration sufficient to produce fully developed speckle. Each cyst region does not contain any scatterers. The ringlike structure around each cyst has slightly different scatterer density than the rest of the phantom. The slight concentration around the border was purposely produced to highlight potential artifacts near the cyst border. The diameter of each cyst is about 20 mm. In the STA system, pulses from each transmit subaperture were prefocused at 70 mm range. Here we compared the results of SF method, STA method, and the real aperture PA method. The SF method gave the best result due

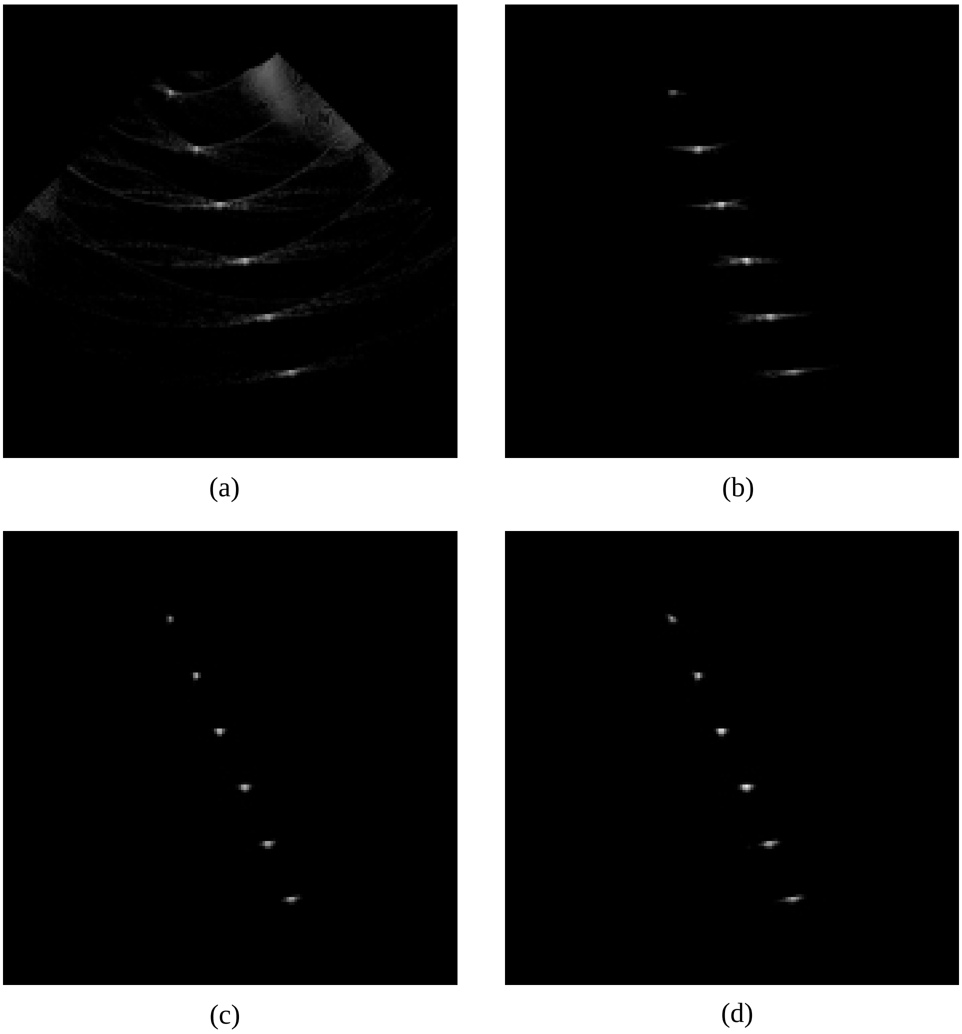


Figure 6.9: *The B-scan images of the Wire Phantom formed by different synthetic aperture imaging methods, displayed over a 50 dB display dynamic range. (a) SAFT (b) M-SAF (c) SF (d) STA (2 transmit subapertures).*

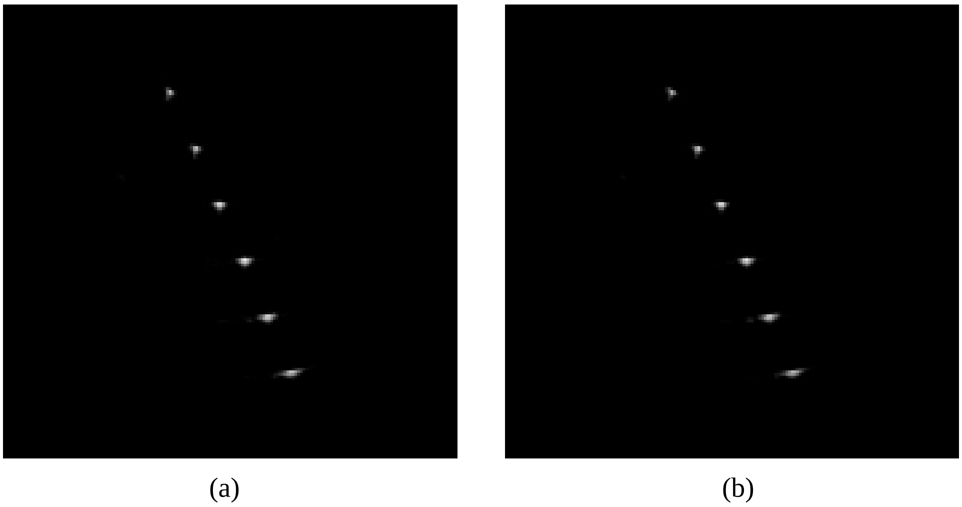


Figure 6.10: *The B-scan images of the Wire Phantom over a 60 dB display dynamic range: (a) PA (sixteen focal zones composite transmission) (b) STA (four transmit subapertures).*

to a DTR focusing mode. The PA method, using a CTDR focusing mode, also gave good image quality. The STA method with four transmit subapertures gave a compatible image quality. However, this method reduced the image frame rate. The STA method using one of the four transmit subapertures gave a slightly worse result. The noise and sidelobe levels were increased with the reduction of the transmission channels.

The third phantom used for the experiments is called the Tissue Phantom. It was used to test the image quality in general. This is a tissue mimicking phantom with sound speed of 1540 m/s and an attenuation rate of 0.5 dB/cm/MHz. The phantom is a tissue mimicking ultrasound resolution phantom containing wire targets, cysts and distributed scatterers. A portion of the sector at the left side is black because one wall of the phantom is within the sector. Fig. 6.15(a) shows an image of this phantom using the STA system without time gain compensation (TGC), and an image with time gain compensation (Fig. 6.15(b)) is also shown for comparison. Beam from each transmit subaperture is prefocused at 60 mm range. Clearly we see that the TGC plays an important role for the STA imaging system in compensating the attenuation effect. Similar experiments were done for the other imaging systems, and the results showed that the TGC was important for the other systems as well.

The B-scan images of the Tissue Phantom obtained by the SF, the STA and the real aperture PA systems are compared in Fig. 6.16. Again we see that the image quality of the STA system is similar to those of the SF and the PA systems.

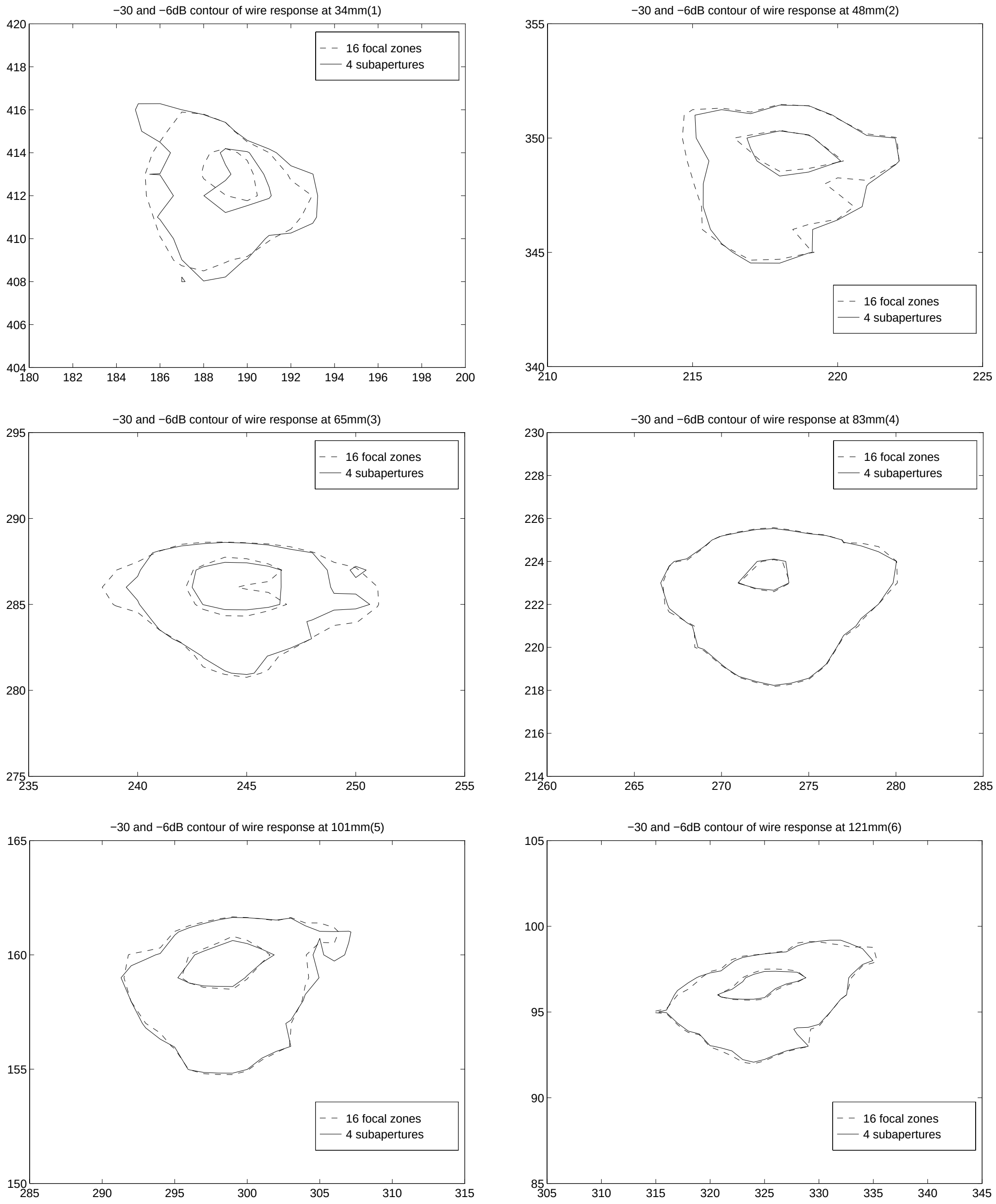
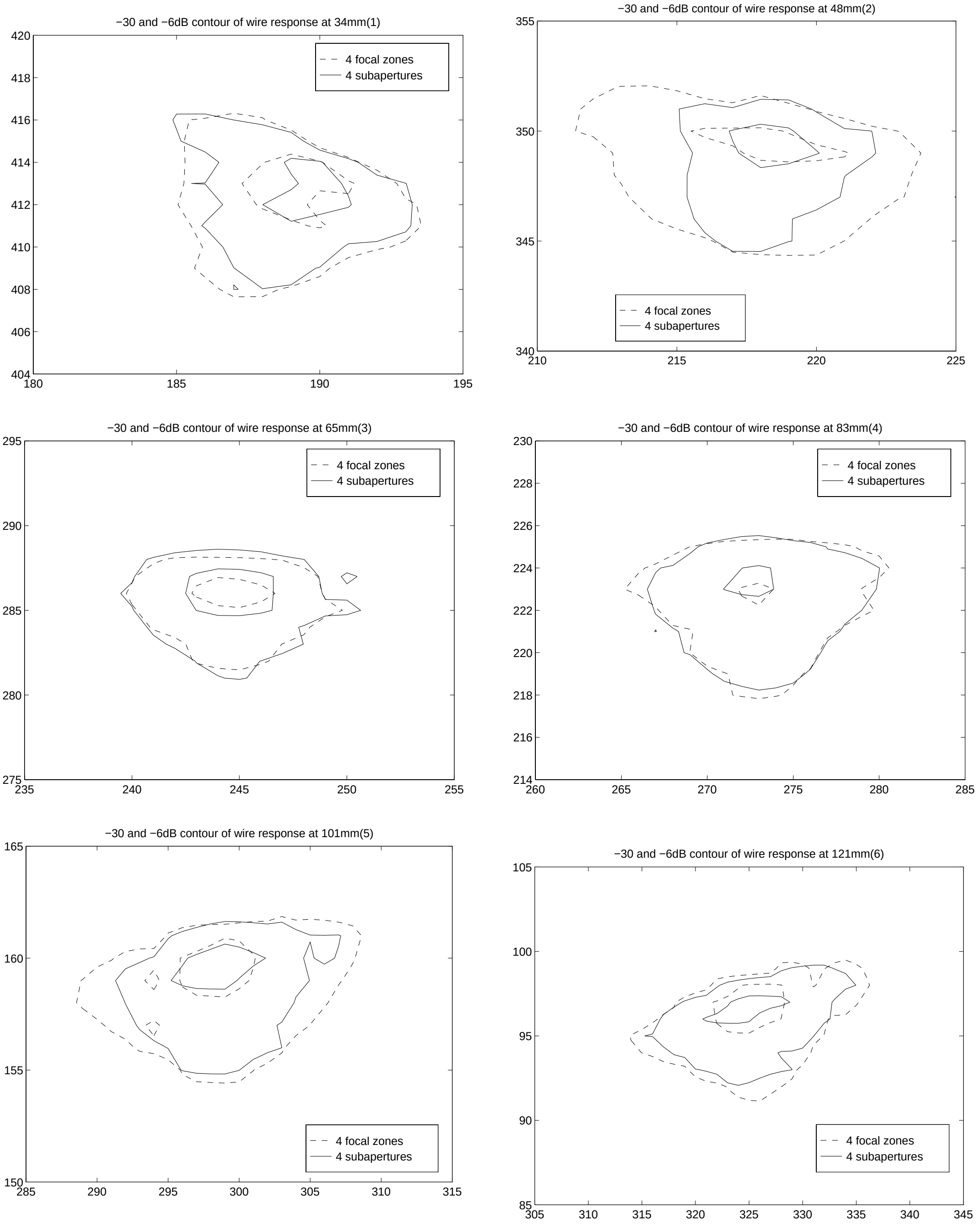


Figure 6.11: *Wire target amplitudes presented by contour plots. A STA system with four transmit subapertures (solid lines) and a PS system with sixteen focal zones (dashed lines) are compared.*



−30 and −6dB contour of wire response at 34mm(1)

−30 and −6dB contour of wire response at 48mm(2)

−30 and −6dB contour of wire response at 65mm(3)

−30 and −6dB contour of wire response at 83mm(4)

−30 and −6dB contour of wire response at 101mm(5)

−30 and −6dB contour of wire response at 121mm(6)

Figure 6.12: *Wire target amplitudes presented by contour plots. A STA system with four transmit subapertures (solid lines) and a PS system with four focal zones (dashed lines) are compared.*

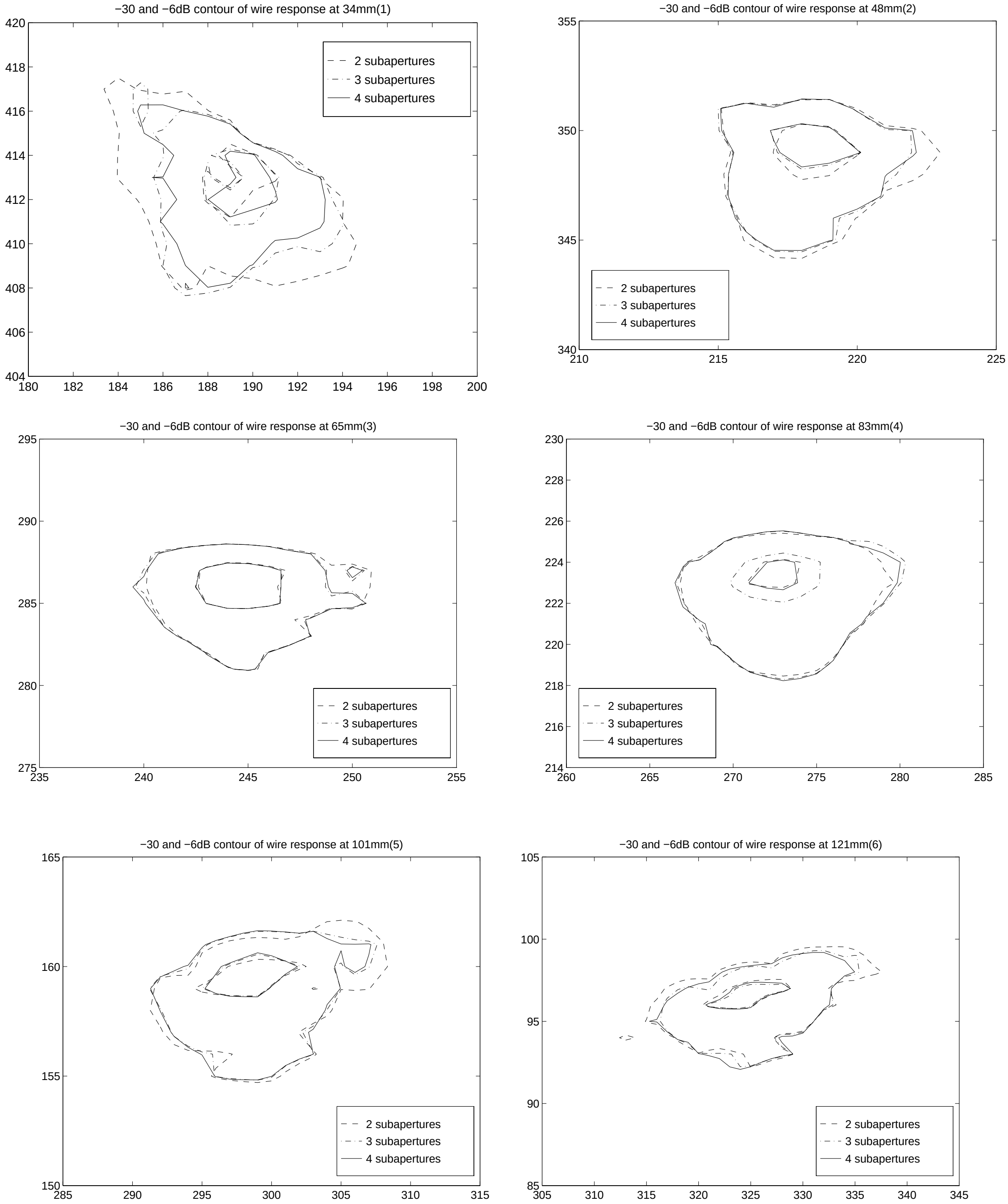
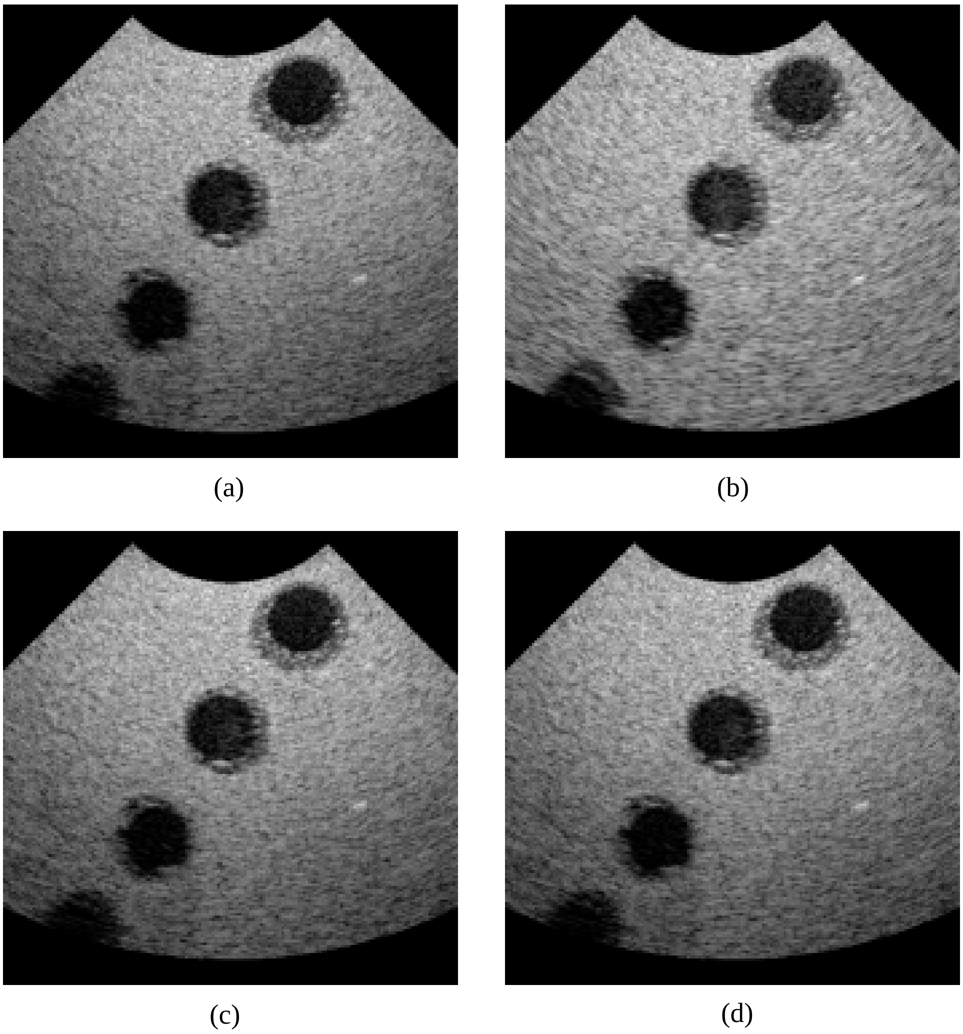


Figure 6.13: *A comparison of the STA systems with different number of transmit subapertures using wire target amplitudes presented by contour plots.*



(a) (b) (c) (d)

Figure 6.14: *B-scan images of the Cyst Phantom displayed over a 60 dB display dynamic range. (a) SF (b) STA with four transmit subapertures, but only the second transmit subaperture was used for image reconstruction (c) STA (four transmit subapertures) (d) PA (sixteen focal zones).*

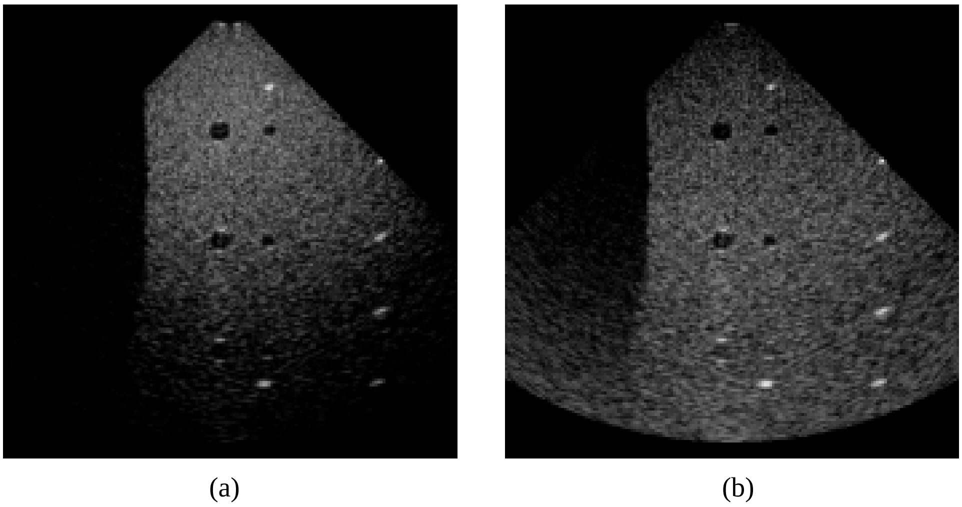
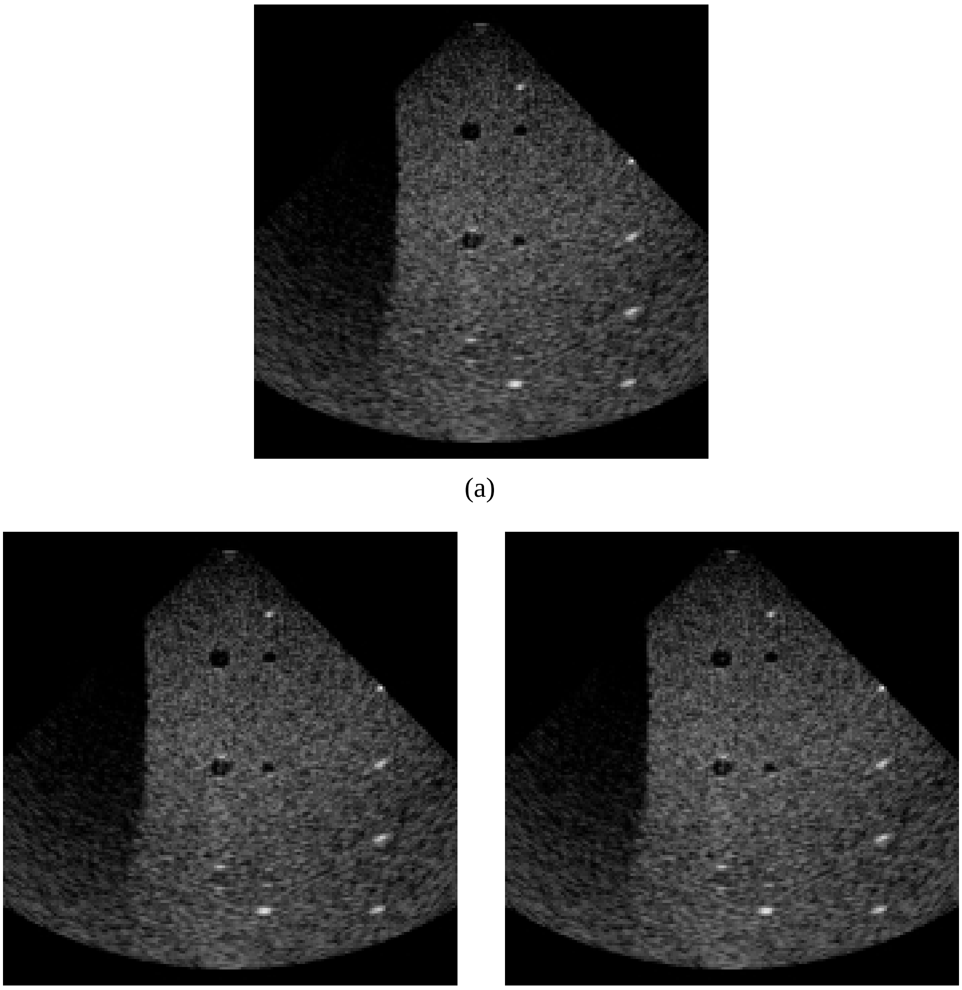


Figure 6.15: *Comparison of images before and after time gain compensation (TGC). B-scan images of the Tissue Phantom over a 60 dB display dynamic range using the STA system (four transmit subapertures). (a) before TGC (b) after TGC.*



(a)

(b)

(c)

Figure 6.16: *B-scan images of the Tissue Phantom over a 60 dB display dynamic range. (a) SF (b) STA (four transmit subapertures) (c) PA (sixteen focal zones).*

Chapter 7

Discussion and Conclusion

Synthetic aperture methods for medical ultrasound imaging are studied in this thesis. Experiments have been done to compare various synthetic aperture methods and to examine the properties of these methods. A new synthetic aperture ultrasound image method, called the synthetic transmit aperture (STA) method, is presented in this thesis (the method was also presented in [19]). The main advantage of this method is the higher system frame rate compared to a composite focusing PA system.

Different synthetic aperture ultrasound methods have different purposes. Tab. 7.1 outlines some of the factors affecting the overall performance for the synthetic aperture methods. The parameters used for the comparison are the number of transmit channels, the number of receive channels, the number of subapertures, the signal-to-noise ratio (SNR), and the transmit focusing mode. The SNR is computed assuming uncorrelated additive electronic noise on receive. SNR_0 is the SNR of a single element synthetic aperture recording. The system complexity is an important measurement to evaluate an imaging system. It can be defined as the number of active channels. For the STA imaging, the system complexity is K_t. The SAFT imaging has the lowest system complexity. However, it is not practical due to its bad beam characteristic and low SNR.

All the SA methods assume that the internal organs are reasonably stationary under data acquisition. Otherwise, motion artifacts will be presented in the images. For tissue

	# Transmit Channel	# Receive Channel	# Subapertures N_s	$\mathrm{SNR}/\mathrm{SNR}_0$ (dB)	Transmit Focusing
PA	N	N	1	$10log_{10}(N^3)$	composite
SAFT	1	1	N	$10log_{10}(N)$	dynamic
M-SAF	K_t	K_r	N_ϵ	$10log_{10}(K_tK_rN_\epsilon)$	dynamic
SRA	N	K_r	N/K_r	$10log_{10}(N^2K_r)$	fixed
SF	1	N	N	$10log_{10}(N^2)$	dynamic
STA	K_t	N	N/K_t	$10log_{10}(N^2K_t)$	fixed

Table 7.1: *A comparison of various synthetic aperture methods. N is the size of the array, $N_\epsilon = N-K_r+1$ and $1 \leq K_r \leq 12$.*

motion $v = 20\text{mm/s}$, wave length $\lambda = 0.44$ mm and pulse repetition frequency PRF = 5000 Hz, the maximum number of subapertures is 12. Within this limit, motion compensation is not necessary. When a 128-element phased array is to be synthesized, the SAFT, M-SAF and SF imaging methods will need a motion compensation because in this case N_s is larger than the maximum number of subapertures. However, using the SRA or the STA methods, the motion compensation is not necessary if proper values of K_t and K_r are chosen. In the STA method, practical number of subapertures is in range of 2–4. Therefore, motion compensation is not required.

In advanced composite focusing systems, additional improvements in image quality are achieved by varying transmit parameters such as aperture diameter, transducer center frequency and pulse repetition frequency in different focusing zones [20][35]. At shallow depths, where attenuation is not a problem, the transmit frequency can be set higher in order to give a high spatial resolution, while the frequency is set lower for deeper zones. The optimization of transmit frequency within each transmit focal zone allows for increased resolution in near field and faster updating rate. A hybrid of the composite focusing and the STA will combine the advantages of both methods. Assume that $N_s = 2$ subapertures are used with $M_f = 4$ different transmit foci. The number of transmit required are $N_s M_f = 8$, and the equivalent number of zones are $M = N_s^2 M_f = 16$. The approximate frame rate increase is still $N_s = 2$, which in this case is reduced to half of what it could have been ($\sqrt{M} = 4$).

Comparison of different forms of focusing schemes by using real full phased array has been made. Fixed focusing on transmit and receive gives noticeable blurring away from the focal point. With full dynamic focusing on transmit and receive, resulting images exhibit the best image quality. However, this focusing mode is not practical in real-time application since imaging frame rate approaches zero. An alternate to the focusing mode is composite transmit focus with N_c focal zones and dynamic receive. This gives comparable image quality at a cost of reduction in frame rate by a factor of N_c. Typical value of N_c is 10 to 15. In conventional abdominal ultrasound imaging, composite focusing is commonly used.

Splitting of the transmit aperture in a phased array system into N_s subapertures allows one to choose the amount of transmit focusing. A subaperture of only a single element ($N_s = N$) gives full dynamic focusing on transmit, while one subaperture ($N_s = 1$) gives a fixed transmit focus. A subdivision into from two to four subapertures is realistic from the point of view of maintaining coherence subject to tissue motion. Also, it has the advantage that it will give an increase in the transmit depth of field which is equavalent to using composite imaging with $N_c = N_s^2$ transmit focal zones. The acquisition time for this method has decreased by a factor of $N_s = \sqrt{N_c}$ compared to the composite imaging of equivalent quality.

7.1 Conclusion

This thesis gives a review and an evaluation of the synthetic aperture methods for medical ultrasound imaging, and presents a new synthetic aperture method which increases the frame rate in ultrasound imaging. The performance of the synthetic aperture methods were investigated both analytically and experimentally. The synthetic aperture methods include the synthetic aperture focusing method (SAFT), multi-element synthetic aperture focusing method (M-SAF), synthetic focusing method (SF), synthetic receive aperture method (SRA), and the new synthetic transmit aperture method (STA). The most of the synthetic aperture methods are used to reduce the system complexity and cost. The new proposed STA method is to increase the imaging frame rate.

The transmit-receive apodization matrix (T/R matrix) and the concept of effective aperture were used as tools for beam pattern analysis. The thesis shows that the T/R matrix can be modified to analyze the effective aperture when subapertures are overlapping. The performance of the synthetic aperture methods were evaluated experimentally by using RF data of wire phantoms. Reconstructed images of different synthetic aperture methods were compared.

The SAFT method gave the worst image quality due to its small active aperture. In the SAFT, only a single element is used as an active aperture. If the inter-element space is $\lambda/2$, the effective aperture is undersampled. This causes the problem of grating lobes. As discussed earlier, the problem can be solved if a smaller inter-element space is used (equal or less than $\lambda/4$). However, this is not a perfect solution as the number of element is increased. The M-SAF method solves this problem by using multi-element subapertures. Compared to the SAFT, the M-SAF method increases the acoustic power, suppresses the grating lobes, and reduces the sidelobe levels. The image quality of the M-SAF method is better than that of the SAFT method, but is still not satisfactory. The SRA method gives a good lateral resolution by using a large transmit array and a synthetic receive mode. The SF method uses a dynamic transmit and a dynmaic receive focusing, and obviously gives the best image quality. The image quality of the SF method is normally better than that of the composite phased array method. However, the SF method is computationally inefficient and is difficult to implement in real time. The advantage of the new STA method is to increase the imaging frame rate. The image quality of the STA method is compatible to that of the composite phased array imaging.

The sampling requirements for the synthetic aperture ultrasound imaging methods are studied. The T/R matrix and the effective aperture are used to examine if a subaperture overlap is necessary to obtain the Nyquist sampling rate. A general formula is given for the beam pattern of the synthetic aperture methods.

Appendix A

Equipments

This appendix presents the software and the hardware used to implement the synthetic aperture imaging systems and to visualize the results.

A.1 Software

The software packages used includes MATLAB and XITE. MATLAB stands for **MAT**rix **LAB**oratory and is written in C by The Math Works Inc. It is an interactive program for numeric computation and data visualization. XITE stands for **X**-based **I**mage Processing **T**ools and **E**nvironment. It is a software package developed in the Image Processing Laboratory (BLAB), Department of Informatics, University of Oslo. It consists of:

- File format BIFF (BLAB Image File Format)
- X11-based BIFF image display program
- More than 140 C-programs for image processing and analysis
- More than 200 C-subroutines for image processing and analysis
- On-line documentation of all routines and programs
- Some images on BIFF format

The file format BIFF is a raster format for two-dimensional and three-dimensional images. In case of 3-D, the image is viewed as a sequence of frames (bands). There are programs to convert between BIFF and many other image formats. More about the BIFF format and the XITE can be found in BLAB report [26]. The XITE is a free software, can be obtained at `http://www.ifi.uio.no/ blab/Software/Xite/` or through anonymous ftp from `ftp.ifi.uio.no`. A "README" file gives further instructions.

In this work, the programs used for beam pattern analysis and contour plots were written in MATLAB. The implementation of the synthetic aperture imaging systems was done by using C-programming and XITE routines. The programs are listed in Appendix B of this thesis.

A.2 Hardware

The programs used in this research were executed on DEC 500/240 stations, SUN SPARC stations and servers, and Silicon Graphics Power Challenge machines, under an UNIX system at the Department of Informatics, Univeristy of Oslo.

Appendix B

Program Codes for Imaging Systems

This appendix lists program codes for noise reduction, upsampling of signals, and image formation using different imaging methods. A short description of the usage of the programs is given so that the programs can be easily used by other researchers.

B.1 Usage of the Programs

From the raw RF data, the following steps are often necessary to reconstruct an image and analyze the result.

1. Converting the raw data to BIFF format. This can be usually done by using XITE program “raw2biff”.
2. Applying noise reduction by using programs in Sec. B.2.1.
3. Upsampling the data by using programs in Sec. B.2.2.
4. Image reconstruction by using PA, STA, SAF or M-SAF method. The programs are in Sec. B.2.4 to Sec. B.2.7.
5. Adjusting display dynamic range by using programs in Sec. B.2.8.
6. Displaying the image by using XITE program “xshow”.

The following scripts give an example of using the programs. In this example, the input raw data *acuson17* is of signed short format. It contains 128×128 recordings and each recording has 2048 samples. The STA method is used. An image of 512×512 pixels is reconstructed and displayed.

```
raw2biff -pt ss acuson17 acuson17.img 2048 128 128
filt <filter coff> acuson17.img filt_a17.img 1 128
upsamp <filter coff> filt_a17.img up_filt_a17.img 1 128
STA <parameter file> up_filt_a17.img result_a17.img 500
     512 4 <nr subaperture> <nr subaperture> 0.065 0.147 0
chgrange  result_a17.img result_a17_60dB.img 60
xshow  result_a17_60dB.img
```

B.2 Program Codes

B.2.1 Noise Reduction

```
/*
  filt.c: filter the input image for noise reduction
*/

#include <xite/includes.h>
#include <stdio.h>
#include <xite/biff.h>
#include <xite/blab.h>
#include <string.h>

#define size 2048
#define filtersize 129

ISS_IMAGE img_in, img_out;
float bandpass[filtersize+1];
int xsize, ysize;

/* copy name and band in string s   */
void getname(char *s, char *name, int band)
{
  char sd[20];

  strcpy(s, name);
  strcat(s, ":");
  sprintf(sd, "%d", band);
  strcat(s , sd);
  printf(" %d ", band);
}

float read_coff(char *filnavn)
{
  FILE *readf;
  int i = 1;
  float coff;

  readf = fopen(filnavn,"r");
  if (readf == NULL)  return 0;
  else {
    for (i = 1; i <= filtersize; i++) {
      fscanf(readf, "%f", &coff);
      bandpass[i] = coff;
    }

    fclose(readf);
    return 1;
  } /* else  */
}

void bpfilt(int band, int trelem)
{
  int m, k, bfsize=(filtersize-1)/2.0;
  float sum;

  for (m=bfsize+1; m<=filtersize; ++m) {
    sum = 0.0;
    for (k=1; k<=m-1; ++k)
      sum += img_in[1][trelem][m-k]*bandpass[k];
    img_out[band][trelem][m-bfsize] = sum;
  }

  for (m=filtersize+1; m<=xsize; ++m) {
    sum = 0.0;
    for (k=1; k<=filtersize; ++k)
      sum += img_in[1][trelem][m-k]*bandpass[k];
    img_out[band][trelem][m-bfsize] = sum;
  }

  for (m=xsize+1; m<=xsize+bfsize; ++m) {
    sum = 0.0;
     for (k=m-xsize; k<=filtersize; ++k)
       sum += img_in[1][trelem][m-k]*bandpass[k];
     img_out[band][trelem][m-bfsize] = sum;
  }
}


main(int argc, char **argv)
{
  int band, band_start, band_stop, bandno, y, i;
  IPIXTYP pixTyp;
  char *info, imgname[80];

  if (argc < 6) {
     printf("Usage: %s <filter coff> <input image> <output image>
       <band start> <band stop>\n",argv[0]);
     exit(1);
  }

  if (read_coff(argv[1]) != 0)    /* read filters coffsients  */
    printf("\nRead input file with bandpass coffsients: %s\n", argv[1]);
  else {
    printf("OBS: can't open the file %s\n", argv[1]);
    exit(3);
  }
  printf("Bandpass filter input image.\n");
  printf("Please, wait .....\nBand: ");

  band_start = atoi(argv[4]);
  getname(imgname, argv[2], band_start);

  /* Does input image file exist? */
  if ((img_in = (ISS_IMAGE)Iread_image(imgname)) == NULL)  {
    fprintf(stderr, " RFerror: file %s is not a biff-file \n", argv[1]);
    exit(2);
  }

  band_stop = atoi(argv[5]);
  bandno = band_stop - band_start + 1;
  pixTyp = Ipixtyp((IBAND) img_in[1]);
  xsize = Ixsize((IBAND) img_in[1]);
  ysize = Iysize((IBAND) img_in[1]);


  info = (char*)malloc(100);
  sprintf(info, "bandpass");
  img_out = (ISS_IMAGE)Imake_image(bandno, info, pixTyp, xsize, ysize);

  band = 1;
  while (band <= bandno) {
    for (y=1;y<=ysize;y++)
      bpfilt(band,y);

    if (++band <= bandno) {
      getname(imgname, argv[2], band+band_start-1);
      if ((img_in = (ISS_IMAGE)Iread_image(imgname)) == NULL) {
        fprintf(stderr, " RFerror: file %s is not a biff-file \n", imgname);
        exit(2);
      }
    }
  }

  /* write image   */
  if (Iwrite_image((IMAGE)img_out,argv[3])==0)
    printf("\nBandpass filtered image written to BIFF file: %s \n", argv[3]);
  else
    printf("\nWRITE ERROR: when writing file: %s. Disk probably full !!!\n", argv[3]);
}
```

B.2.2 Upsampling of Data

```
/*
   upsamp.c: interpolate signals with interpolation factor 4.
*/

#include <xite/includes.h>
#include <stdio.h>
```

```
#include <xite/biff.h>
#include <xite/blab.h>
#include <string.h>

#define newxsize 8192
#define n 4   /* n: interpolation factor  */
#define filtersize 129

ISS_IMAGE img_in, img_out;
float kaiser[filtersize+1];
float record[newxsize+1];
int xsize, ysize;

/* copy name and band in string s   */
void getname(char *s, char *name, int band)
{
  char sd[20];
  strcpy(s, name);
  strcat(s, ":");
  sprintf(sd, "%d", band);
  strcat(s , sd);
  printf(" %d ", band);
}

float read_filt(char *filnavn)
{
  FILE *readf;
  int i = 1;
  float coff;

  readf = fopen(filnavn,"r");
  if (readf == NULL)  return 0;
  else {
    for (i = 1; i <= filtersize; i++) {
        fscanf(readf, "%f", &coff);
        kaiser[i] = coff;
    }
    fclose(readf);
    return 1;
  } /* else  */
}

void upsamp(int band, int trelem)
{
  int m, k, hfsize=(filtersize-1)/2.0;

  for (m=hfsize+1; m<=filtersize; ++m)
    for (k=1; k<=m-1; ++k)
      img_out[band][trelem][m-hfsize] += n*record[m-k]*kaiser[k];

  for (m=filtersize+1; m<=newxsize; ++m)
    for (k=1; k<=filtersize; ++k)
      img_out[band][trelem][m-hfsize] += n*record[m-k]*kaiser[k];
  for (m=newxsize+1; m<=newxsize+hfsize; ++m)
    for (k=m-newxsize; k<=filtersize; ++k)
      img_out[band][trelem][m-hfsize] += n*record[m-k]*kaiser[k];
}



main(int argc, char **argv)
{
  int band, band_start, band_stop, bandno, y, x;
  IPIXTYP pixTyp;
  char *info, imgname[80];

  if (argc < 6) {
     printf("Usage: %s <filter coff> <input image> <output image>
       <band start> <band stop>\n",argv[0]);
     exit(1);
    };
  /* filter coff: cofficients of an lowpass filter
     input image: which needs to interpolate
     output image: image after interpolation
     band start: image band start number
     band stop: image band stop number
   */

   if (read_filt(argv[1]) != 0)    /* read lowpass filter coffsients  */
    printf("\nRead input file with lowpass filter coffsients: %s\n", argv[1]);
  else {
    printf("OBS: can't open the file %s\n", argv[1]);
    exit(3);
  }

  printf("Interpolate the input image with factor 4.\n");
  printf("Please, wait .....\nBand: ");

  band_start = atoi(argv[4]);
  getname(imgname, argv[2], band_start);

  /* Does input image file exist? */
  if ((img_in = (ISS_IMAGE)Iread_image(imgname)) == NULL)  {
    fprintf(stderr, " RFerror: file %s is not a biff-file \n", argv[1]);
    exit(2);
  }

  band_stop = atoi(argv[5]);
  bandno = band_stop - band_start + 1;
  pixTyp = Ipixtyp((IBAND) img_in[1]);
  xsize = Ixsize((IBAND) img_in[1]);
  ysize = Iysize((IBAND) img_in[1]);


  info = (char*)malloc(100);
  sprintf(info, "highpass");
  img_out = (ISS_IMAGE)Imake_image(bandno, info, pixTyp, newxsize, ysize);
  band = 1;
  while (band <= bandno) {
    for (y=1;y<=ysize;y++) {
      for (x=1; x<= xsize; ++x)
        record[n*(x-1)+1] = img_in[1][y][x];
        upsamp(band,y);
    }

    if (++band <= bandno) {
      getname(imgname, argv[2], band+band_start-1);
      if ((img_in = (ISS_IMAGE)Iread_image(imgname)) == NULL) {
        fprintf(stderr, " RFerror: file %s is not a biff-file \n", imgname);
        exit(2);
      }
    }
  }


  /* write image   */
  if (Iwrite_image((IMAGE)img_out,argv[3])==0)
    printf("\nInterpolated image written to BIFF file: %s \n", argv[3]);
  else
    printf("\nWRITE ERROR: when writing file: %s. Disk probably full !!!\n", argv[3]);
}
```

B.2.3 Libraries and Procedures

```
#include <xite/includes.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <xite/biff.h>
#include <xite/blab.h>

#define MAX(a,b)  (a>b) ? (a):(b)
#define PI2  6.28318530717959   /* 2*PI  */
#define sinPI4  0.7071          /* sin(PI/4)  */
#define MaxNoelem  128          /* max. number of elements in an aperture  */
#define NBeam 201               /* number of beams */
#define RD 512                  /* number of focusing points along range */
#define FNTr  2.0               /* F-number of transmitting  */
#define FNRe  1.5               /* F-number of receing  */

/* ---------------
   Golbal variable
   ---------------
 */
int NR,     /* number of channels  */
    NS,     /* number of samples per record  */
    BS;     /* number of bytes per sample  */
```

```
float SF,   /* A/D sampling rate in Hz  */
      StF,  /* transducer center frequency in Hz */
      ID,   /* element space in m   */
      V,    /* sound velocity in m/sec   */
      TOff; /* time offset (in sec)  */

float Fixdelta[NBeam+1][MaxNoelem+1],    /* delay for fixed focus  */
      ***Trdelta,    /* Composite transmit delay matrix  */
      ***Ddelta,     /* Dynamic delay matrix[range][angle][element] */
      ***Dsubdelta;  /* Dynamic delay matrix on synthetic aperture  */

float polar[RD+1][NBeam+1];  /*  beam:  polar[range][theta] */
float hamming[MaxNoelem];    /*  Hamming window  */

ISS_IMAGE img_in;  /* input image with raw data in signed short  */
IR_IMAGE img_out;  /* output image after beamforming  */

int size,          /* size of output image in pixel */
    interpft,      /* interpolation factor, must be integer  */
    no_samp,       /* total number of samples per record  */
    subelem,       /* number of elements in a subaperture  */
    tgc;           /* variable for time gain compensation  */

float img_lengd,   /* image size in sec */
      IDsec,       /* inter-element distance in sec  */
      pix_size,    /* pixel size in sec  */
      fcsize,      /* range segment in sec */
      fdepth,      /* fixed focus point in sec */
      sbeam;       /* one segment of 90 degrees sector  */

/* ------------------
      Prosedures
   ------------------  */
/* time gain compensation  */
float TGC(float dp)  /* dp: range in sec  */
{
  float a = 0.5; /* (dB/cm/MHz) */

  return pow(10, a*StF*0.000001*dp*V*100/10);
}

/* read parameter from the file named "filnavn"  */
int read_prm(char *filnavn)
{
  FILE *readf;
  char line[80];

  readf = fopen(filnavn,"r");
  if (readf == NULL)  return 0;
  else {
    fgets(line, sizeof(line), readf);  /* skip first two lines  */
    fgets(line, sizeof(line), readf);

    fgets(line, sizeof(line), readf);
    sscanf(line, "%d", &NR);
    fgets(line, sizeof(line), readf);
    sscanf(line, "%f", &SF);
    SF *= 1000000;   /* in Hz  */
    fgets(line, sizeof(line), readf);
    sscanf(line, "%f", &StF);
    StF *= 1000000;  /*  Hz  */
    fgets(line, sizeof(line), readf);  /* skip this line  */
    fgets(line, sizeof(line), readf);
    sscanf(line, "%f", &TOff);
    TOff *= 0.000001;   /* in sec  */
    fgets(line, sizeof(line), readf);
    sscanf(line, "%d", &NS);
    fgets(line, sizeof(line), readf);
    sscanf(line, "%d", &BS);
    fgets(line, sizeof(line), readf);  /* skip this line  */
    fgets(line, sizeof(line), readf);
    sscanf(line, "%f", &ID);
    ID *= 0.001;  /* in meter  */
    fgets(line, sizeof(line), readf);
    sscanf(line, "%f", &V);
    V *= 1000;   /* in meter/sec   */

    fclose(readf);
    return 1;
  } /* else  */
} /* read_prm   */

/* compute fixed focus delay  */
void compute_fixdel(void)
{
  int elem, beam;
  float stheta, b1, tmp1, tmp2, tmp3, tmp4;

  tmp1 = fdepth*fdepth;
  tmp2 = 2*fdepth;
  for (beam=1; beam<=NBeam; ++beam) {
    stheta = (beam-1.0)*sbeam-sinPI4;
       /* focus on a half circle, -sin(pi/4)<=stheta<=sin(pi/4) */
    tmp3 = stheta*tmp2;

    for (elem=1; elem<=NR; ++elem) {
      tmp4 = (-NR/2.0+elem-0.5)*IDsec;
      b1 = sqrt(tmp1+tmp4*(tmp4-tmp3));
      /* use formula: b^2 = a^2 + c^2 - 2*a*c*cos(B)
         a,b,c: 3 lines of a triangle
         B: angle between a and c  */
      Fixdelta[beam][elem] = fdepth-b1;  /* sec */
    } /* for elem.. */
  } /* for beam..  */
} /* compute_fix... */

/* compute composite transmit focus delay  */
void compute_fzone(int zone)
```

```
{
  int lpzone, z, r_start, r_stop = 0, beam, elem, rd, rd1, rd2,
      rdstart, rdstop;
  float r, b1, tmp, tmp1, tmp2, tmp3, tmp4, stheta, tmp5;

  tmp5 = 0.5*NS/SF+0.5*TOff;
  rd1 = (int)rint(0.5* TOff/fcsize);
  if (img_lengd<tmp5) rd2 = (int)RD;
  else rd2 = (int)rint(tmp5/fcsize);
  rdstop = rd1;
  lpzone = (int)rint((rd2-rd1)/(float)zone);
  for(z=1; z<=zone; ++z) {
    r_start = r_stop+1;
    rdstart = rdstop+1;
    rdstop = rdstart+lpzone;
    r_stop = rdstop;
    if (rdstop > rd2) { rdstop = rd2; r_stop = RD; }
    r = (rdstart+rdstop)*fcsize/2.0;

    tmp1 = r*r;
    tmp2 = 2*r;

    for (beam=1; beam<=NBeam; ++beam) {
      stheta = (beam-1.0)*sbeam-sinPI4;
      tmp3 = stheta*tmp2;

      for (elem=1; elem<=NR; ++elem) {
        tmp4 = (-NR/2.0+elem-0.5)*IDsec;
        b1 = sqrt(tmp1+tmp4*(tmp4-tmp3));
        tmp = r-b1;

        for (rd=r_start; rd <=r_stop; ++rd)
          Trdelta[rd][beam][elem] = tmp;
      }
    }
  }
} /* compute_fzone */

/* Compute dynamic focus delay on synthetic aperture */
void compute_subdel(int nopulse)
{
  int elem, selem, beam, rd, elem_start, elem_stop;
  float stheta, b1, tmp1, tmp2, tmp3, tmp4, b2, r, tmp5, tmp6;

  tmp5 = subelem*IDsec;
  for (rd=1; rd<=RD; ++rd) {
    r = fcsize * rd;
    tmp1 = r * r;
    tmp2 = 2 * r;

    for (beam = 1; beam <= NBeam; ++beam) {
      stheta = (beam - 1.0) * sbeam - sinPI4;
      tmp3 = stheta * tmp2;
      for (selem = 1; selem <= nopulse; ++selem) {
        tmp4 = (-nopulse / 2.0 + selem - 0.5) * tmp5;
        b1 = sqrt(tmp1 + tmp4 * (tmp4 - tmp3));
        Dsubdelta[rd][beam][selem] = r-b1;  /* sec */
      }
    } /* for beam  */
  } /* for rd  */
} /* compute_del */

/* Compute dynamic focus delay  */
void compute_del(void)
{
  int elem, beam, rd;
  float stheta, b1, tmp1, tmp2, tmp3, tmp4, r;

  for (rd=1; rd<=RD; ++rd) {
    r = fcsize*rd;
    tmp1 = r*r;
    tmp2 = 2*r;

    for (beam=1; beam<=NBeam; ++beam) {
      stheta = (beam-1.0)*sbeam-sinPI4;
      tmp3 = stheta*tmp2;

      for (elem=1; elem<=NR; ++elem) {
        tmp4 = (-NR/2.0+elem-0.5)*IDsec;
        b1 = sqrt(tmp1+tmp4*(tmp4-tmp3));
        Ddelta[rd][beam][elem] = r-b1;  /* sec */
      } /* for elem */
    } /* for beam  */
  } /* for rd  */
} /* compute_del */

/* scan conversion  */
void scanconv(void)
{
  int x, y;
  float stheta, avstx, avsty, avstr, r_low, r_high, b_low, b_high,
        r_weight, b_weight, r, b, q, p_low, p_high, p1, p2, p3, p4,
        tmp1, tmp2, tmp3;

  q = pix_size / fcsize;
  tmp1 = 0.5*TOff;
  tmp3 = 0.5*NS/SF+tmp1;
  if (tmp3>img_lengd) tmp3 = img_lengd;
  for (y=1; y<=size; ++y)
    for (x=1; x<=size; ++x) {
      avstx = -size/2.0-0.5+x;
      avsty = y-1;
      avstr = sqrt(avstx*avstx+avsty*avsty);
      tmp2 = avstr*pix_size;
      if (tmp2>=tmp1 && tmp2<=tmp3) {
        stheta = avstx/avstr;
        r = avstr*q;
        b = (stheta+sinPI4)/sbeam+1;

        if ((r>1.0) && (r<(float)(RD-1)) &&
           (b >1.0) && (b <(float)(NBeam-1))) {
          r_low = floor(r);
          r_high = r_low+1.0;
          r_weight = r-r_low;
                  b_low = floor(b);
          b_high = b_low+1.0;
          b_weight = b-b_low;

          /* bilinear interpolation  */
           p1 = abs(polar[(int)r_low][(int)b_high]);
          p2 = abs(polar[(int)r_low][(int)b_low]);
          p3 = abs(polar[(int)r_high][(int)b_high]);
          p4 = abs(polar[(int)r_high][(int)b_low]);

          p_low = (p1-p2)*b_weight+p2;
          p_high = (p3-p4)*b_weight+p4;
          if (tgc == 1)   /* Time gain compensation is needed  */
            img_out[1][y][x] = ((p_high-p_low)*r_weight+p_low)*TGC(tmp2);
          else  img_out[1][y][x] = (p_high-p_low)*r_weight+p_low;
        }
      }
    }
}  /* scanconv  */


/* display a dynamic range with given range  */
void chgrange(IR_IMAGE img, float range, int band)
{
  int x, y, xsize, ysize;
  float tmp, maxval = 0.0;

  xsize = Ixsize((IBAND) img[1]);
  ysize = Iysize((IBAND) img[1]);
  for (y=1; y<=ysize; ++y)
    for (x=1;x<=xsize; ++x)
      maxval = MAX(abs(img[band][y][x]), maxval);
      /* find the largest value in the whole image  */

  for (y=1; y<=ysize; ++y)
    for (x=1; x<=xsize; ++x) {
      if (img[band][y][x]==0) tmp = 20*log10(1/maxval);
      else  tmp = 20*log10(abs(img[band][y][x])/maxval);

      img[band][y][x] = MAX(tmp,-range);
      /* change value to a given range if tmp<-range  */
    } /* for */
} /* chgrange  */
```

B.2.4 PA Imaging Method

```
/***************

   PA.c - Convert full set raw data to B-scan image, using
                delay and sum beamformer
   Various focus can be applied:
     FTR: fixed focus on transmit and receive
     FTDR: fixed focus on transmit and dynamic focus on receive
     CTDR: coposite transmit focus and dynamic receive focus
     DTR: dynamic focus on transmit and receive
   Window apodization (hamming window) is used on receive.
   Time gain compensation is possible.
*/

#include "acuson.h"

/* ------------------
      Prosedures
   ------------------  */
void compute_polarFTR(int elem_start, int elem_stop)
{
  int beam, rd, elem, elemTr, elemRe, N, i,
      Trnr_start, Trnr_stop, Renr_start, Renr_stop;
  float r, sum, s, s_low, s_high, s_weight, a0 = 0.53836, a1 = 0.46164;

  for (rd = 1; rd <= RD; ++rd) {
    r = rd * fcsize;    /* range  */

    Trnr_start = Renr_start = (elem_start+elem_stop-1)/ 2;
    Trnr_stop = Renr_stop = Trnr_start + 1;

    elem = (int)rint(r / (2 * FNTr * IDsec));
    Trnr_start -= elem;
    Trnr_stop += elem;
    if (Trnr_start < elem_start)  Trnr_start = elem_start ;
    if (Trnr_stop > NR)  Trnr_stop = elem_stop;

    elem = (int)rint(r / (2 * FNRe * IDsec));
    Renr_start -= elem;
    Renr_stop += elem;
    if (Renr_start < elem_start) Renr_start = elem_start;
    if (Renr_stop > elem_stop)  Renr_stop = elem_stop;

    N= Renr_stop-Renr_start;
    for (i=0; i<=N; ++i)
      hamming[i] = a0-a1*cos(PI2*i/N);

    for (beam = 1; beam <= NBeam; ++beam) {
      sum = 0.0;
      for (elemTr = Trnr_start; elemTr <= Trnr_stop; ++elemTr)
        for (elemRe = Renr_start; elemRe <= Renr_stop; ++elemRe)
          if (elemTr != elemRe && elemRe != 1 && elemRe != 2) {
            s = (2*r-TOff-Fixdelta[beam][elemTr]-Fixdelta[beam][elemRe])*SF*interpft;
          s_low = floor(s);
          s_high = s_low + 1.0;
          s_weight = s - s_low;

          if (s > 1.0 && s < (float)(no_samp-1))
            /* linear interpolation  */
            sum = sum+((img_in[elemTr][elemRe][(int)s_high]-
                      img_in[elemTr][elemRe][(int)s_low])
                      *s_weight+img_in[elemTr][elemRe][(int)s_low])
                      *hamming[elemRe-Renr_start];
        } /* if (elemTr .. */
      polar[rd][beam] = abs(sum);
    }  /* for (beam..  */
  }  /* for (rd ...   */
} /* compute_polarFTR   */


void compute_polarFTDR(int elem_start, int elem_stop)
{
  int beam, rd, elem, elemTr, elemRe, N, i,
      Trnr_start, Trnr_stop, Renr_start, Renr_stop;
  float r, sum, s, s_low, s_high, s_weight, a0 = 0.53836, a1 = 0.46164;

  for (rd = 1; rd <= RD; ++rd) {
    r = rd * fcsize;

    Trnr_start = Renr_start = (elem_start + elem_stop - 1) / 2;
    Trnr_stop = Renr_stop = Trnr_start + 1;

    elem = (int)rint(r / (2 * FNTr * IDsec));
    Trnr_start -= elem;
    Trnr_stop += elem;
    if (Trnr_start < elem_start)  Trnr_start = elem_start ;
    if (Trnr_stop > elem_stop)  Trnr_stop = elem_stop;

    elem = (int)rint(r / (2 * FNRe * IDsec));
    Renr_start -= elem;
    Renr_stop += elem;
    if (Renr_start < elem_start) Renr_start = elem_start;
    if (Renr_stop > elem_stop)  Renr_stop = elem_stop;

    N= Renr_stop-Renr_start;
    for (i=0; i<=N; ++i)
      hamming[i] = a0-a1*cos(PI2*i/N);

    for (beam = 1; beam <= NBeam; ++beam) {
      sum = 0.0;
      for (elemTr = Trnr_start; elemTr <= Trnr_stop; ++elemTr)
        for (elemRe = Renr_start; elemRe <= Renr_stop; ++elemRe)
          if (elemTr != elemRe && elemRe != elemRe != 2) {
            s = (2*r-TOff-Fixdelta[beam][elemTr]-Ddelta[rd][beam][elemRe])*SF*interpft;
            s_low = floor(s);
            s_high = s_low + 1.0;
            s_weight = s - s_low;
            if (s > 1.0 && s < (float)(no_samp - 1))
                    sum = sum+((img_in[elemTr][elemRe][(int)s_high]-
                        img_in[elemTr][elemRe][(int)s_low])
        *s_weight+img_in[elemTr][elemRe][(int)s_low])
                        *hamming[elemRe-Renr_start];
          }  /* if (elemTr ... */
      polar[rd][beam] = abs(sum);
    }  /* for (beam..  */
  }  /* for (rd ...   */
} /* compute_polarFTDR   */

void compute_polarCTDR(int elem_start, int elem_stop)
{
  int beam, rd, elem, elemTr, elemRe, N, i,
      Trnr_start, Trnr_stop, Renr_start, Renr_stop;
  float r, tgc, sum, s_low, s_high, s_weight, s, a0 = 0.53836, a1 = 0.46164;

  for (rd = 1; rd <= RD; ++rd) {
    r = rd * fcsize;

    Trnr_start = Renr_start = (int)rint((elem_start + elem_stop - 1) / 2.0);
    Trnr_stop = Renr_stop = Trnr_start + 1;

    elem = (int)rint(r / (2 * FNTr * IDsec));
    Trnr_start -= elem;
    Trnr_stop += elem;
    if (Trnr_start < elem_start)  Trnr_start = elem_start ;
    if (Trnr_stop > elem_stop)  Trnr_stop = elem_stop;

    elem = (int)rint(r / (2 * FNRe * IDsec));
    Renr_start -= elem;
    Renr_stop += elem;
    if (Renr_start < elem_start) Renr_start = elem_start;
    if (Renr_stop > elem_stop)  Renr_stop = elem_stop;

    N= Renr_stop-Renr_start;
    for (i=0; i<=N; ++i)
      hamming[i] = a0-a1*cos(PI2*i/N);

    for (beam = 1; beam <= NBeam; ++beam) {
      sum = 0.0;

      for (elemTr = Trnr_start; elemTr <= Trnr_stop; ++elemTr)
       for (elemRe = Renr_start; elemRe <= Renr_stop; elemRe++)
         if (elemTr != elemRe && elemRe != 2 && elemRe != 1) {
           s = (2*r-TOff-Trdelta[rd][beam][elemTr]
                     -Ddelta[rd][beam][elemRe])*SF*interpft;
           s_low = floor(s);
           s_high = s_low + 1.0;
           s_weight = s - s_low;

           if (s > 1.0 && s < (float)(no_samp - 1))
             sum = sum+((img_in[elemTr][elemRe][(int)s_high]-
                       img_in[elemTr][elemRe][(int)s_low])
                       *s_weight+img_in[elemTr][elemRe][(int)s_low])
                       *hamming[elemRe-Renr_start];
         }  /* if (elemTr !=   */
      polar[rd][beam] = abs(sum);
    }  /* for (beam..  */
  }  /* for (rd ...   */
} /* compute_polarCTDR   */


void compute_polarDTR(int elem_start, int elem_stop)
{
  int beam, rd, elem, elemTr, elemRe, N, i,
      Trnr_start, Trnr_stop, Renr_start, Renr_stop;
  float r, sum, s_low, s_high, s_weight, s, a0 = 0.53836, a1 = 0.46164;

  for (rd = 1; rd <= RD; ++rd) {
    r = rd * fcsize;

    Trnr_start = (int)rint((elem_start + elem_stop - 1) / 2.0);
    Trnr_stop = Trnr_start+1;
    Renr_start = (int)rint(NR/2.0); Renr_stop = Renr_start+1;

    elem = (int)rint(r / (2 * FNTr * IDsec));
    Trnr_start -= elem;
    Trnr_stop += elem;
    if (Trnr_start < elem_start)  Trnr_start = elem_start ;
    if (Trnr_stop > elem_stop)  Trnr_stop = elem_stop;

    elem = (int)rint(r / (2 * FNRe * IDsec));
    Renr_start -= elem;
    Renr_stop += elem;
    if (Renr_start < 1) Renr_start = elem_start;
    if (Renr_stop > NR)  Renr_stop = elem_stop;

    N= Renr_stop-Renr_start;
    for (i=0; i<=N; ++i)
      hamming[i] = a0-a1*cos(PI2*i/N);

    for (beam = 1; beam <= NBeam; ++beam) {
      sum = 0.0;
      for (elemTr = Trnr_start; elemTr <= Trnr_stop; ++elemTr)
       for (elemRe = Renr_start; elemRe <= Renr_stop; elemRe++)
         if (elemTr != elemRe && elemRe != 1 &&elemRe != 2) {
           s = (2*r-TOff-Ddelta[rd][beam][elemTr]-Ddelta[rd][beam][elemRe])*SF*interpft;
           s_low = floor(s);
           s_high = s_low + 1.0;
           s_weight = s - s_low;
           if (s > 1.0 && s < (float)(no_samp - 1))
             sum = sum+((img_in[elemTr][elemRe][(int)s_high]
                       -img_in[elemTr][elemRe][(int)s_low])
       *s_weight+img_in[elemTr][elemRe][(int)s_low])
                       *hamming[elemRe-Renr_start];
           }  /* if (elemTr !=   */
      polar[rd][beam] = abs(sum);
    }  /* for (beam..  */
  }  /* for (rd ...   */
} /* compute_polarDTR   */


/* Fixed focus on both transmit and receive  */
void FixTR(float range)
{
  printf("Compute delay\n");
  compute_fixdel();
  printf("  Compute sum\n");
  compute_polarFTR(1,NR);
  printf("\nScan converting\n");
  scanconv();
  chgrange(img_out, range, 1);
} /* FixTR  */

/* Fixed focus on transmit and dynamic focus on receive  */
void FTDR(float range)
{
  printf("Compute delay\n");
  compute_fixdel();
  compute_del();
  printf("  Compute sum\n");
  compute_polarFTDR(1,NR);
  printf("\nScan converting\n");
  scanconv();
  chgrange(img_out, range, 1);
} /* FTDR  */

void CTDR(float range, int nrzones)
{
  printf("Compute delay\n");
  compute_fzone(nrzones);
  compute_del();
  printf("  Compute sum\n");
  compute_polarCTDR(1, NR);
  printf("  Scan converting\n");
  scanconv();
  chgrange(img_out, range, 1);
}

/* Dynamic focus on both transmit and receive  */
void DTR(float range)
{
  printf("Compute delay\n");
  compute_del();
  printf("  Compute sum\n");
  compute_polarDTR(1,NR);
  printf("  Scan converting\n");
  scanconv();
  chgrange(img_out, range, 1);
} /* DTR  */
```

```
main(int argc, char **argv)
{
  int bandno, ysize, i, j, nrzones;
  float range;
  char *info;

  /* Allocate memory for delay-arrays */
  if ((Ddelta = (float ***) malloc((RD+1)*sizeof(float **))) == NULL) {
    fprintf(stdout,"Feil i malloc (2)\n\n Exit ....\n\n");
    exit(0);
  }

 if ((Trdelta = (float ***) malloc((RD+1)*sizeof(float **))) == NULL) {
    fprintf(stdout,"Feil i malloc (3)\n\n Exit ....\n\n");
    exit(0);
  }

  for (i=0; i<RD+1; i++) {
    if ((Ddelta[i] = (float **) malloc((NBeam+1)*sizeof(float *))) == NULL) {
      fprintf(stdout,"Feil i malloc (4), i=%d\n\n Exit ....\n\n",i);
      exit(0);
    }
    if ((Trdelta[i] = (float **) malloc((NBeam+1)*sizeof(float *))) == NULL) {
      fprintf(stdout,"Feil i malloc (5), i=%d\n\n Exit ....\n\n",i);
      exit(0);
    }

    for (j=0; j<NBeam+1; j++) {
      if ((Ddelta[i][j] = (float *) malloc((MaxNoelem+1)*sizeof(float))) == NULL) {
        fprintf(stdout,"Feil i malloc (4), i=%d, j=%j\n\n Exit ....\n\n",i,j);
        exit(0);
      }
      if ((Trdelta[i][j] = (float *) malloc((MaxNoelem+1)*sizeof(float))) == NULL) {
        fprintf(stdout,"Feil i malloc (5), i=%d, j=%j\n\n Exit ....\n\n",i,j);
        exit(0);
      }
    }
  }

  if (argc != 10 && argc != 11) {
    printf("\nUsage: %s <parameter file> <rawdata image> <output image>\n
      <FTR/FTDR/CTDR/DTR> <range (dB)> <output image size (pixel)>\n
      <interpolation factor> <image size in meter>\n
      <Time Gain Compensation (1/0)> /[<focus depth (in meter)/#focal zones>]\n\n",
       argv[0]);
    exit(1);
  };
  /*
     parameter file: file includes parameters conserning rawdata image
     raw data image: rf-data
     output imgae: new image after delay and sum beamforming
     FTR: fixed focus both on transmit and receive
     FTDR: fixed focus on transmit and dynamic focus on receive
     CTDR: composite transmit focus and dynamic receive focus
     DTR: dynamic focus both on transmit and receive
     range (dB): dynamic display range
     output image size: size of output image in pixel
     interpolation factor: must be integer
     tgc: need time gain compensation? 1/0(y/n)
     focus depth: given in meterif FTR or FTDR is wanted
   */

  if (strcmp(argv[4], "FTR") != 0 && strcmp(argv[4], "FTDR") != 0 &&
      strcmp(argv[4], "DTR") != 0 && strcmp(argv[4], "CTDR") != 0) {
    printf("\nOBS: Unknown focus mode, valid one: FTR, FTDR, CTDR, DTR\n\n");
    exit(2);
  }

  if ((strcmp(argv[4], "FTR") == 0 || strcmp(argv[4], "FTDR") == 0
       || strcmp(argv[4], "CTDR") == 0)  && argc != 11) {
    /*  focus is not given  */
    printf("\nOBS: Focus depth or number of focal zones is not given\n");
    printf("\nUsage: %s <parameter file> <rawdata image> <output image>\n
      <FTR/FTDR/CTDR/DTR> <range (dB)> <output image size (pixel)>\n
      <interpolation factor> <image size in meter>\n
      <Time Gain Compensation (1/0)> /[<focus depth (in meter)/#focal zones>]\n\n",
      argv[0]);
    exit(3);
  };

  /* Does input parameter file exist? */
  if (read_prm(argv[1]) != 0)     /* read necessary parameters  */
    printf("\nRead parameter file: %s\n", argv[1]);
  else {
    printf("OBS: can't open the file %s\n", argv[1]);
    exit(4);
  }

  printf("Parameter: NR%d NS%d BS%d SF%0.fHz StF%0.fHz TOff%fsec\nID%fm V%0.fm/s\n",
    NR, NS, BS, SF, StF, TOff, ID, V);

  /* Does input image file exist? */
  if ((img_in = (ISS_IMAGE)Iread_image(argv[2])) != NULL)
    printf("\nRead image: %s\n", argv[2]);
  else {
    fprintf(stderr, " RFerror: file %s is not a biff-file \n", argv[2]);
    exit(6);
  }

  range = atof(argv[5]);
  size = atoi(argv[6]);
  interpft = atoi(argv[7]);
  img_lengd = atof(argv[8])/V;   /* in sec  */
  tgc = atoi(argv[9]);

  bandno = Inbands((IMAGE) img_in);
```

```
  ysize = Iysize((IBAND) img_in[1]);
  no_samp = Ixsize((IBAND) img_in[1]);

  /*  check if input image matches parameter file and interpolation factor */
  if (no_samp != NS * interpft) {
    printf("Input image doesn't match either input parameter file
      or interpolation factor.\n\n");
    exit(7);
  }

  IDsec = ID / V;
  pix_size = img_lengd / (float)size;
  fcsize = img_lengd / (float)RD;
  sbeam = 2 * sinPI4 / (float)(NBeam - 1);

  info = (char*)malloc(100);
  if (strcmp(argv[4], "DTR") == 0) {
      printf("%s interpolation factor %d image length %0.fmm\n", argv[4],interpft,
        atof(argv[8])*1000);
      sprintf(info, "%s %s i%d",argv[0],argv[4], interpft);
  }
  else if (strcmp(argv[4], "CTDR") == 0) {
    nrzones = atoi(argv[10]);
    sprintf(info, "%s %s focal zones%d i%d",argv[0],argv[4],
      nrzones, interpft);
    printf("%s img_lengd %fmm focal zones%d interpolation factor %d\n",
            argv[4], atof(argv[8])*1000, nrzones, interpft);
  }
  else {
    sprintf(info, "%s %s focus%fmm i%d",argv[0],argv[4], atof(argv[10])*1000, interpft);
    fdepth = atof(argv[10]) / V; /* in sec */
    printf("%s img_lengd %fmm fdepth %fmm interpolation factor %d\n",
            argv[4], atof(argv[8])*1000, atof(argv[10])*1000, interpft);
  }

  /* define output image  */
  img_out = (IR_IMAGE)Imake_image(1, info, Ireal_typ, size, size);
  Iappend_line((IMAGE) img_out, info);

  if (strcmp(argv[4], "FTR") == 0)  FixTR(range);
  else  if (strcmp(argv[4], "FTDR") == 0)  FTDR(range);
  else if (strcmp(argv[4], "CTDR") == 0)  CTDR(range,nrzones);
  else DTR(range);

  if (Iwrite_image((IMAGE) img_out, argv[3])==0)
    printf("\nBeamformed cartsian image written to BIFF file: %s \n", argv[3]);
  else
    printf("WRITE ERROR: when writing file: %s. Disk probably full !!!\n", argv[3]);
}
```

B.2.5 STA Imaging Method

```
/****
```

```
   STA.c - use synthetic transmit aperture method to convert sample data
   to B-scan image.
   Time gain compensation is possible.
*/

/* ---------------------------------------
   NAME
     STA (synthetic transmit aperture)
   PURPOSE

   INPUT

   OUTPUT

   HISTORY
     Hongxia Yao - Juni, 1997: Created.

   LAST MODIFIED
     Sep. 10. 1997 - 15:25.
* ----------------------------------------- */
#include "acuson.h"

/* ------------------
      Prosedures
   ------------------   */

void compute_polar(int band, int elem_start, int elem_stop, int nopulse)
{
  int beam, rd, samp, elem, elemTr, elemRe, N, i,
      Trnr_start, Trnr_stop, Renr_start, Renr_stop, trbeam;
  float r, sum, s_low, s_high, s_weight, s, xsub, d, m, l,
        sphi, stheta, a0 = 0.53836, a1 = 0.46164;

  xsub = (-nopulse/2.0+band-0.5)*subelem*IDsec;
  for (rd=1; rd<=RD; ++rd)
    {
      r = rd*fcsize;

      Trnr_start = (int)rint((elem_start+elem_stop-1)/2.0);
      Renr_start = (int)rint(NR/2.0);
      Trnr_stop = Trnr_start+1;
      Renr_stop = Renr_start+1;

      elem = (int)rint(r/(2*FNTr*IDsec));
      Trnr_start -= elem;
      Trnr_stop += elem;
      if (Trnr_start < elem_start)  Trnr_start = elem_start ;
      if (Trnr_stop > elem_stop)  Trnr_stop = elem_stop;

      elem = (int)rint(r/(2*FNRe*IDsec));
      Renr_start -= elem;
      Renr_stop += elem;
      if (Renr_start < 1) Renr_start = 1;
      if (Renr_stop > NR)  Renr_stop = NR;

      N= Renr_stop-Renr_start;
      for (i=0; i<=N; ++i)
hamming[i] = a0-a1*cos(PI2*i/N);

      for (beam=1; beam<=NBeam; ++beam)
{
   sum = 0.0;

   sphi = (beam-1.0)*sbeam-sinPI4;
   d = r*cos(asin(sphi));
   m = r*sphi;
   l = m-xsub;
   stheta = l/sqrt(l*l+d*d);
   trbeam = (int)rint((stheta+sinPI4)/sbeam+1.0);

   if (trbeam>=1 && trbeam<=NBeam) {
   for (elemTr=Trnr_start; elemTr<=Trnr_stop; ++elemTr)
    for (elemRe=Renr_start; elemRe<=Renr_stop; ++elemRe)
      {
if (elemTr!= elemRe && elemRe!=2 && elemRe!=1) {
  s = (2*r-TOff-Fixdelta[trbeam][elemTr-elem_start+1]
       -Dsubdelta[rd][beam][band]
                       -Ddelta[rd][beam][elemRe])*SF*interpft;

s_low = floor(s);
s_high = s_low+1.0;
s_weight = s-s_low;

if (s>1.0 && s<(float)(no_samp-1))
  sum = sum+((img_in[elemTr][elemRe][(int)s_high]
        -img_in[elemTr][elemRe][(int)s_low])*s_weight
        +img_in[elemTr][elemRe][(int)s_low])
        * hamming[elemRe-Renr_start];
}  /* if (elemTr !=   */
      }  /* for (elemRe..  */ }

  polar[rd][beam] += sum;
}  /* for (angle..  */
    }  /* for (rd ...   */
} /* compute_polarDTR   */



/* Fixed focus on transmit and dynamic focus on receive  */
void FTDR(float range, int nopulse)
{
  int elem_start = 1, elem_stop, band;

  printf("Compute delay\n");
  compute_fixdel();
  compute_subdel(nopulse);
  compute_del();
  for (band=1; band<=nopulse; ++band)
    {
      elem_stop = elem_start+subelem-1;
      printf("Band %d, channels: %d to %d.\n", band, elem_start, elem_stop);

      printf("  Compute sum\n");
      compute_polar(band, elem_start, elem_stop, nopulse);
      elem_start = elem_stop+1;
    }

  printf("  Scan converting\n");
  scanconv();
  chgrange(img_out,range, 1);
} /* DTR  */


main(int argc, char **argv)
{
  int  no_subapt, bandno, ysize, i, j;
  float range;
  char *info;


  /* Allokere plass til delay-arrayer */
  if ((Ddelta = (float ***) malloc((RD+1)*sizeof(float **))) == NULL) {
    fprintf(stdout,"Feil i malloc (2)\n\n Exit ....\n\n");
    exit(0);
  }

 if ((Dsubdelta = (float ***) malloc((RD+1)*sizeof(float **))) == NULL) {
    fprintf(stdout,"Feil i malloc (3)\n\n Exit ....\n\n");
    exit(0);
  }

  for (i=0; i<RD+1; i++) {

    if ((Ddelta[i] = (float **) malloc((NBeam+1)*sizeof(float *))) == NULL) {
      fprintf(stdout,"Feil i malloc (4), i=%d\n\n Exit ....\n\n",i);
      exit(0);
    }
    if ((Dsubdelta[i] = (float **) malloc((NBeam+1)*sizeof(float *))) == NULL) {
      fprintf(stdout,"Feil i malloc (5), i=%d\n\n Exit ....\n\n",i);
      exit(0);
    }

    for (j=0; j<NBeam+1; j++) {
      if ((Ddelta[i][j] = (float *) malloc((MaxNoelem+1)*sizeof(float)))
          == NULL) {
        fprintf(stdout,"Feil i malloc (4), i=%d, j=%j\n\n Exit ....\n\n",i,j);
        exit(0);
      }

      if ((Dsubdelta[i][j] = (float *) malloc((MaxNoelem+1)*sizeof(float)))
          == NULL) {
        fprintf(stdout,"Feil i malloc (5), i=%d, j=%j\n\n Exit ....\n\n",i,j);
        exit(0);
      }
    }
  }

  if (argc < 11)
    {
      printf("Usage: %s <parameter file> <rawdata image> <output image> <range (dB)>\n
        <output image size (pixel)> <interpolation factor> <nr subaperture>\n
        <focus depth (in meter)> <image size in meter> <TGC (1/0)> \n\n",
        argv[0]);
      exit(1);
    };
  /*
     parameter file: file includes parameters conserning rawdata image
     raw data image: rf-data
     output imgae: new image after delay and sum beamforming
     FTR: fixed focus both on transmit and receive
     FTDR: fixed focus on transmit and dynamic focus on receive
     DTR: dynamic focus both on transmit and receive
     range (dB): dynamic display range
     output image size: size of output image in pixel
     interpolation factor: must be integer
     no subaperture: number of subaperture
     focus depth: given in meter
     TGC: time gain compensation (1(yes) or 0(no))
   */

  /* Does input parameter file exist? */
  if (read_prm(argv[1]) != 0)     /* read necessary parameters  */
    printf("\nRead parameter file: %s\n", argv[1]);
  else {
    printf("OBS: can't open the file %s\n", argv[1]);
    exit(2);
  }

  printf("Parameter: NR%d NS%d BS%d SF%0.fHz StF%0.fHz TOff%fsec \nID%fm V%0.fm/s\n",
    NR, NS, BS, SF, StF, TOff, ID, V);


  /* Does input image file exist? */
  if ((img_in = (ISS_IMAGE)Iread_image(argv[2])) != NULL)
    printf("\nRead image: %s\n", argv[2]);
  else {
    fprintf(stderr, " RFerror: file %s is not a biff-file \n", argv[2]);
    exit(4);
  }

  range = atof(argv[4]);
  size = atoi(argv[5]);
  interpft = atoi(argv[6]);
  no_subapt = atoi(argv[7]);
  fdepth = atof(argv[8])/V;
  img_lengd = atof(argv[9])/V;
  tgc = atoi(argv[10]);

  subelem = (int)rint(NR/no_subapt);

  /*
     check if input image matches parameter file and interpolation factor
   */
  bandno = Inbands((IMAGE) img_in);
  ysize = Iysize((IBAND) img_in[1]);
  no_samp = Ixsize((IBAND) img_in[1]);
  if (no_samp != NS * interpft) {
    printf("Input image doesn't match either input parameter file or
      interpolation factor.\n\n");
    exit(7);
      }

  info = (char*)malloc(100);

  sprintf(info, "%s fd%0.fmm s%d i%d RD%d",argv[0],atof(argv[8])*1000,
  no_subapt, interpft, RD);

  /* define output image  */
  img_out = (IR_IMAGE)Imake_image(1, info, Ireal_typ, size, size);
  Iappend_line((IMAGE) img_out, info);

  IDsec = ID / V;
  pix_size = img_lengd / (float)size;
  fcsize = img_lengd / (float)RD;
  sbeam = 2 * sinPI4 / (float)(NBeam - 1);
  printf("Image size %0.fmm focus depth %0.fmm\n",
 atof(argv[9])*1000,atof(argv[8])*1000);

  FTDR(range, no_subapt);

  if (Iwrite_image((IMAGE) img_out, argv[3])==0)
    printf("\nBeamformed cartsian image written to BIFF file: %s \n", argv[3]);
    else
    printf("WRITE ERROR: when writing file: %s. Disk probably full !!!\n",
   argv[3]);
}
```

B.2.6 SAFT Imaging Method

```
/***************

   SAF.c - convert sample data to B-scan image by using synthetic aperture
           focusing technique
*/

#include "acuson.h"
```

```
/* ------------------
      Prosedures
   ------------------  */
void compute_polarDTR(int elem_start, int elem_stop)
{
  int beam, rd, samp, elem, elemTr, elemRe, i;
  float r, sum, s_low, s_high, s_weight, s;

  for (rd = 1; rd <= RD; ++rd) {
    r = rd * fcsize;
    for (beam = 1; beam <= NBeam; ++beam) {
      sum = 0.0;
      for (i=1; i<=NR; ++i) {
        elemRe = i+1;
        if (elemRe == 2) elemRe = 3;
        if (elemRe > NR) elemRe = NR-1;
        s = (2*r-TOff-Ddelta[rd][beam][i]-Ddelta[rd][beam][elemRe])*SF*interpft;
        s_low = floor(s);
        s_high = s_low+1.0;
        s_weight = s-s_low;

        if (s > 1.0 && s < (float)(no_samp - 1))
          sum = sum +((img_in[i][elemRe][(int)s_high]-img_in[i][elemRe][(int)s_low])
                *s_weight+img_in[i][elemRe][(int)s_low]);
        }
      polar[rd][beam] = abs(sum);
    }  /* for (beam..  */
  }  /* for (rd ...   */
} /* compute_polarDTR   */


/* Dynamic focus on both transmit and receive  */
void DTR(float range)
{
  printf("Compute delay\n");
  compute_del();
  printf("  Compute sum\n");
  compute_polarDTR(1,NR);
  printf("  Scan converting\n");
  scanconv();
  chgrange(img_out, range, 1);
} /* DTR  */

main(int argc, char **argv)
{
  int bandno, ysize, i, j;
  float range;
  char *info;

  /* Allokere plass til delay-arrayer */
  if ((Ddelta = (float ***) malloc((RD+1)*sizeof(float **))) == NULL) {
    fprintf(stdout,"Feil i malloc (1)\n\n Exit ....\n\n");
    exit(0);
  }
```

```
for (i=0; i<RD+1; i++) {
  if ((Ddelta[i] = (float **) malloc((NBeam+1)*sizeof(float *))) == NULL) {
    fprintf(stdout,"Feil i malloc (2), i=%d\n\n Exit ....\n\n",i);
    exit(0);
  }
  for (j=0; j<NBeam+1; j++) {
    if ((Ddelta[i][j] = (float *) malloc((MaxNoelem+1)*sizeof(float))) == NULL) {
      fprintf(stdout,"Feil i malloc (2), i=%d, j=%j\n\n Exit ....\n\n",i,j);
      exit(0);
    }
  }
}

if (argc != 9) {
    printf("\nUsage: %s <parameter file> <rawdata image> <output image>\n
      <range (dB)> <output image size (pixel)>\n
      <interpolation factor> <image size in meter> <Time Gain Compensation (1/0)>\n\n",
       argv[0]);
    exit(1);
};

/* Does input parameter file exist? */
if (read_prm(argv[1]) != 0)     /* read necessary parameters  */
  printf("\nRead parameter file: %s\n", argv[1]);
else {
  printf("OBS: can't open the file %s\n", argv[1]);
  exit(4);
}

printf("Parameter: NR%d NS%d BS%d SF%0.fHz StF%0.fHz TOff%fsec \nID%fm V%0.fm/s\n",
        NR, NS, BS, SF, StF, TOff, ID, V);

/* Does input image file exist? */
if ((img_in = (ISS_IMAGE)Iread_image(argv[2])) != NULL)
  printf("\nRead image: %s\n", argv[2]);
else {
  fprintf(stderr, " RFerror: file %s is not a biff-file \n", argv[2]);
  exit(6);
}

range = atof(argv[4]);
size = atoi(argv[5]);
interpft = atoi(argv[6]);
img_lengd = atof(argv[7])/V;
tgc = atoi(argv[8]);

/*  check if input image matches parameter file and interpolation factor */
bandno = Inbands((IMAGE) img_in);
ysize = Iysize((IBAND) img_in[1]);
no_samp = Ixsize((IBAND) img_in[1]);

if (no_samp != NS*interpft) {
  printf("Input image doesn't match either input parameter file or
      interpolation factor.\n\n");
    exit(7);
  }

  IDsec = ID/V;
  pix_size = img_lengd/(float)size;
  fcsize = img_lengd/(float)RD;
  sbeam = 2*sinPI4/(float)(NBeam-1);

  info = (char*)malloc(100);
  printf("%s interpolation factor %d image length %0.fmm\n", argv[0],interpft,
    atof(argv[7])*1000);
  sprintf(info, "%s i%d",argv[0],interpft);

  /* define output image  */
  img_out = (IR_IMAGE)Imake_image(1, info, Ireal_typ, size, size);
  Iappend_line((IMAGE) img_out, info);

  DTR(range);

  if (Iwrite_image((IMAGE) img_out, argv[3])==0)
    printf("\nBeamformed cartsian image written to BIFF file: %s \n", argv[3]);
    else
    printf("WRITE ERROR: when writing file: %s. Disk probably full !!!\n", argv[3]);
}
```

B.2.7 M-SAF Imaging Method

```
/***************

   MSAF.c - using multi-element synthetic aperture focusing method, convert
   sample data to B-scan image.
*/

#include "acuson.h"

float tau[MaxNoelem+1];  /* defocused delay  */

/* ------------------
      Prosedures
   ------------------  */

/* Compute delay for dynamic focus for nopulse */
void compute_syndel(int nopulse)
{
  int elem, selem, beam, rd, elem_start, elem_stop;
  float stheta, b1, tmp1, tmp2, tmp3, tmp4, b2, r, tmp6;

  for (rd=1; rd<=RD; ++rd) {
    r = fcsize * rd;
    tmp1 = r * r;
    tmp2 = 2 * r;
    for (beam = 1; beam <= NBeam; ++beam) {
      stheta = (beam - 1.0) * sbeam - sinPI4;
      tmp3 = stheta * tmp2;

      for (selem = 1; selem <= nopulse; ++selem) {
        tmp4 = (-nopulse / 2.0 + selem - 0.5) * IDsec;
        b1 = sqrt(tmp1 + tmp4 * (tmp4 - tmp3));
        Dsubdelta[rd][beam][selem] = r-b1;  /* sec */
      }
    } /* for beam  */
  } /* for rd  */
} /* compute_del */



void compute_polar(int band, int elem_start, int elem_stop)
{
  int beam, rd, samp, elem, elemTr, elemRe,
      Trnr_start, Trnr_stop, Renr_start, Renr_stop, trbeam;
  float r, sum, s_low, s_high, s_weight, s, xsub, d, m, l, sphi, stheta;


  for (rd = 1; rd <= RD; ++rd) {
    r = rd * fcsize;

    for (beam = 1; beam <= NBeam; ++beam) {
      sum = 0.0;

      for (elemTr = elem_start; elemTr <= elem_stop; ++elemTr)
        for (elemRe = elem_start; elemRe <= elem_stop; elemRe++)
          if (elemTr != elemRe && elemRe != 2 && elemRe != 1) {
            s = (2*r-TOff-tau[elemTr-elem_start+1]-Dsubdelta[rd][beam][band]
                 -Ddelta[rd][beam][elemRe])*SF*interpft;

            s_low = floor(s);
            s_high = s_low+1.0;
            s_weight = s-s_low;

            if (s>1.0 && s<(float)(no_samp-1))
              sum = sum+((img_in[elemTr][elemRe][(int)s_high]
                        -img_in[elemTr][elemRe][(int)s_low])
        *s_weight+img_in[elemTr][elemRe][(int)s_low]);
          }  /* if (elemTr !=   */
      polar[rd][beam] += sum;
    }  /* for (beam..  */
  }  /* for (rd ...   */
} /* compute_polar  */

void msaf(float range, int nopulse)
{
  int elem_start = 1, elem_stop, band;

  printf("Compute delay\n");
  compute_syndel(nopulse);
  compute_del();

  for (band = 1; band <= nopulse; ++band) {
    elem_stop = elem_start+subelem-1;
    printf("Band %d, channels: %d to %d.\n", band, elem_start, elem_stop);
    printf("  Compute sum\n");
    compute_polar(band, elem_start, elem_stop);
    elem_start = elem_stop -subelem+2;
  }

  printf("  Scan converting\n");
  scanconv();
  chgrange(img_out, range, 1);
}


main(int argc, char **argv)
{
  int  no_subapt, bandno, ysize, elem, i, j;
  float range, xn;
  char *info;

   /* Allokere plass til delay-arrayer */
  if ((Dsubdelta = (float ***) malloc((RD+1)*sizeof(float **))) == NULL) {
    fprintf(stdout,"Feil i malloc (1)\n\n Exit ....\n\n");
    exit(0);
  }

  if ((Ddelta = (float ***) malloc((RD+1)*sizeof(float **))) == NULL) {
    fprintf(stdout,"Feil i malloc (2)\n\n Exit ....\n\n");
    exit(0);
  }

  for (i=0; i<RD+1; i++) {
    if ((Ddelta[i] = (float **) malloc((NBeam+1)*sizeof(float *))) == NULL) {
      fprintf(stdout,"Feil i malloc (3), i=%d\n\n Exit ....\n\n",i);
      exit(0);
    }

      if ((Dsubdelta[i] = (float **) malloc((NBeam+1)*sizeof(float *)))==NULL) {
        fprintf(stdout,"Feil i malloc (4), i=%d\n\n Exit ....\n\n",i);
        exit(0);
      }

      for (j=0; j<NBeam+1; j++) {
        if ((Ddelta[i][j] = (float *) malloc((MaxNoelem+1)*sizeof(float))) == NULL) {
          printf("Feil i malloc (3), i=%d, j=%j\n\n Exit ....\n\n",i,j);
          exit(0);
        }

        if ((Dsubdelta[i] = (float **) malloc((NBeam+1)*sizeof(float *))) == NULL) {
          fprintf(stdout,"Feil i malloc (4), i=%d\n\n Exit ....\n\n",i);
          exit(0);
        }
      }
    }

  if (argc != 10) {
      printf("\nUsage: %s <parameter file> <rawdata image> <output image>\n
        <range (dB)> <output image size (pixel)> <interpolation factor>\n
        <#elem in subaperture> <image size in meter> <Time Gain Compensation (1/0)>\n\n",
        argv[0]);
      exit(1);
  };

  /* Does input parameter file exist? */
  if (read_prm(argv[1]) != 0)     /* read necessary parameters  */
    printf("\nRead parameter file: %s\n", argv[1]);
  else {
    printf("OBS: can't open the file %s\n", argv[1]);
    exit(2);
  }

  printf("Parameter: NR%d NS%d BS%d SF%0.fHz StF%0.fHz TOff%fsec \nID%fm V%0.fm/s\n",
    NR, NS, BS, SF, StF, TOff, ID, V);

  /* Does input image file exist? */
  if ((img_in = (ISS_IMAGE)Iread_image(argv[2])) != NULL)
    printf("\nRead image: %s\n", argv[2]);
  else {
    fprintf(stderr, " RFerror: file %s is not a biff-file \n", argv[2]);
    exit(4);
  }

  range = atof(argv[4]);
  size = atoi(argv[5]);
  interpft = atoi(argv[6]);
  subelem = atoi(argv[7]);
  img_lengd = atof(argv[8])/V;
  tgc = atoi(argv[9]);

  /*   check if input image matches parameter file and interpolation factor */
  bandno = Inbands((IMAGE) img_in);
  ysize = Iysize((IBAND) img_in[1]);
  no_samp = Ixsize((IBAND) img_in[1]);
  if (no_samp != NS * interpft) {
    printf("Input image doesn't match either input parameter file or
      interpolation factor.\n\n");
    exit(7);
  }

  info = (char*)malloc(100);
  sprintf(info, "%s #elem%d i%d RD%d",argv[0], subelem, interpft, RD);

  /* define output image  */
  img_out = (IR_IMAGE)Imake_image(1, info, Ireal_typ, size, size);
  Iappend_line((IMAGE) img_out, info);
  IDsec = ID/V;
  pix_size = img_lengd/(float)size;
  fcsize = img_lengd/(float)RD;
  sbeam = 2 * sinPI4/(float)(NBeam - 1);

  /* compute defocusing delay tau */
  for (elem=1;elem<=subelem; ++elem) {
    xn=(-subelem / 2.0 + elem - 0.5) * IDsec;
    tau[elem]=xn*xn/V/IDsec/subelem;
  }

  msaf(range, NR-subelem+1);

  if (Iwrite_image((IMAGE) img_out, argv[3])==0)
    printf("\nBeamformed cartsian image written to BIFF file: %s \n", argv[3]);
  else
    printf("WRITE ERROR: when writing file: %s. Disk probably full !!!\n", argv[3]);
}
```

B.2.8 Dynamic Display Range

```
/*
   chgrange.c:  change display dynamic range of a image to a given range
 */

#include <xite/includes.h>
#include <stdlib.h>
#include <xite/biff.h>
#include <xite/blab.h>

#define MAX(a,b)  (a>b) ? (a) :(b)

main(argc, argv)
int argc;
char **argv;
{
  int x, y, xsize, ysize, band, bandno;
  float range, tmp;
  IR_IMAGE img_in, img_out;
  char *info, txt[80];

  if (argc != 4) {
     printf("Usage: %s <input image> <output image> <range>\n",argv[0]);
     exit(1);
  };
  /* input image: which needs change range
     output imgae: new image after range shifting
     range: dispaly dynamic range, 0~range dB
   */

  range = atof(argv[3]);

  printf("\nRead image: %s\n",argv[1]);
  img_in = (IR_IMAGE)Iread_image(argv[1]);
  printf("Finish reading: %s\n",argv[1]);

  xsize = Ixsize((IBAND) img_in[1]);
  ysize = Iysize((IBAND) img_in[1]);
  bandno = Inbands((IMAGE) img_in);

  info = (char*)malloc(100);
  Iget_line((IMAGE) img_in, txt);
  sprintf(info, "range %d dB %s", (int)range, txt);
  img_out = (IR_IMAGE)Imake_image(bandno, info, Ireal_typ, ysize, xsize);
  Iappend_line((IMAGE) img_out, info);

  /* change range  */
  for (band = 1; band <= bandno; ++band)
    for (y = 1;y <= ysize; ++y)
      for (x = 1; x <= xsize; ++x)
img_out[band][y][x] = MAX(img_in[band][y][x], -range);


  /* write image   */
  if (Iwrite_image((IMAGE)img_out,argv[2])==0)
    printf("\nBeamformed image written to BIFF file: %s \n", argv[2]);
    else
    printf("WRITE ERROR: when writing file: %s. Disk probably full !!!\n",
   argv[2]);

}
```

Bibliography

[1] Kirk W. Beach. 1975-2000: A quarter century of ultrasound technology. *Ultrasound in Med. and Biol.*, **18**(4):377–388, 1992.

[2] S. Bennett, D.K. Peterson, D. Corl, and G.S. Kino. A real-time synthetic aperture digital acoustic imaging system. *Acoust. Imaging*, **10**:669–692, 1980.

[3] A.P. Berkhoff, H.J. Huisman, J.M. Thijssen, E.M.G.P. Jacobs, and R.J.F. Homan. Fast scan conversion algorithms for displaying ultrasound sector images. *Ultrasonic Imaging*, **16**:87–108, 1994.

[4] Richard Y. Chiao and Lewis J. Thomas. Aperture formation on reduced-channel arrays using the transmit-receive apodization matrix. 1996.

[5] Richard E. Davidsen, Jørgen A. Jensen, and Stephen W. Smith. Two-dimentional random arrays for real time volumetric imaging. *Ultrasonic Imaging*, **16**:143–163, 1994.

[6] Brett L. Douglas and Hua Lee. Synthetic aperture sonar imaging with a multiple-element receive array. In *IEEE ICASSP*, volume **5**, pages 445–449, 1993.

[7] Al Elgarem. Multidepth synthetic aperture processing of ultrasonic data. *IEEE Trans. Ultrason. Ferroelec. Freq. Contr.*, **36**(3):384–385, 1989.

[8] K.R. Erikson, F.J. Fry, and L.P. Jones. Ultrasound in medicine - a review. *IEEE Trans. Sonics Ultrason.*, **SU-21**(3):144–170, 1974.

[9] S. Freeman, Pai-Chi Li, and M. O'Donnell. Retrospective dynamic transmit focusing. *Ultrasonic Imaging*, **17**:173–196, 1995.

[10] Richard L. Goldberg, Stephen W. Smith, and Lewis F. Brown. In vivo imaging using a copolymer phased array. *Ultrasonic Imaging*, **14**:234–248, 1992.

[11] Aage Gronningsaeter, Bjørn A.J. Angelsen, Audun Gresli, Hans Torp, and David T. Linker. Blood noise reduction in intravascular ultrasound imaging. *IEEE Trans. Ultrason., Ferroelec. and Freq. Cont.*, **42**(2):200–209, 1995.

[12] Fredric J. Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *IEEE*, **66**:51–83, 1978.

[13] Andreas Heimdal and Hans Torp. Blood signal detection in intravascular ultrasound imaging. In *Proc. NOBIM*, pages 88–96, 1995.

[14] S. Holm and K. Kristoffersen. Analysis of worst-case phase quantization sidelobes in focused beamforming. In *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, pages 593–599, 1992.

[15] Sverre Holm. Digital beamforming in ultrasound imaging. In *Nordic Signal Processing Symposium*, 1994.

[16] Sverre Holm. *Focused multi-element synthetic aperture imaging*. Department of Informatics, University of Oslo, 1995.

[17] Sverre Holm. Medical ultrasound transducers and beamforming. In *Proc. 15th Int. Congress on Acoustics, Trondheim, Norway*, pages 26–30, June 1995.

[18] Sverre Holm, Bjørnar Elgetun, and Geir Dahl. Properties of the beampattern of weight- and layout-optimized sparse arrays. *IEEE Trans. Ultrason., Ferroelec. and Freq. Cont.*, **44**(5):983–991, 1997.

[19] Sverre Holm and Hongxia Yao. *Improved framerate with synthetic transmit aperture imaging using prefocused subapertures,*. IEEE Ultrasonics Symposium, Oct. 1997.

[20] Syed O. Ishrak, K.R. Erikson, W.A. Reckwerdt, and R.S. Spratt. *Method and apparatus for focusing transmission and reception of ultrasonic beams.* US Patent, 5301674, Apr. 1994.

[21] Don H. Johnson and Dan E. Dudgeon. *Array Signal Processing: Concepts and Techniques.* Prentice-Hall, 1993.

[22] Mustafa Karaman, Pai-Chi Li, and Matthew O'Donnell. Synthetic aperture imaging for small scale systems. *IEEE Trans. Ultrason. Ferroelec. Freq. Contr.*, **42**:429–442, 1995.

[23] Gordon S. Kino. *Acoustic Waves: Devices, Imaging, and Analog Signal Processing.* Prentice-Hall, 1987.

[24] H.G. Larsen and S.C. Leavitt. An image display algorithm for use in real-time sector scanners with digital scan converters. *IEEE Ultrasonic Symposium Proceedings*, pages 763–765, 1980.

[25] Nadav Levanon. *Radar Principles.* Wiley-Interscience, 1988.

[26] T. Lønnestad. The biff image concept, file format, and routine library. BLAB Report No. 29, Dept. of Informatics, Univ. of Oslo, 1990.

[27] Geoffrey R. Lockwood, Pai-Chi Li, Matthew O'Donnell, and F. Stuart Foster. Optimizing the radiation pattern of sparse periodic linear arrays. *IEEE Trans. Ultrason., Ferroelec. and Freq. Cont.*, **43**(1):4–12, 1996.

[28] Jian-Yu Lu, Hehong Zou, and James F. Greenleaf. Biomedical ultrasound beam forming. *Ultrasound in Med. and Biol.*, **20**:403–428, 1994.

[29] Alberto Moreira. Real-time synthetic aperture radar (sar) processing with a new subaperture approach. *IEEE Trans. on Geoscience and Remote Sensing*, **30**(4):714–722, 1992.

[30] Wayne Niblack. *An Introduction to Digital Image Processing.* Prentice-Hall, 1986.

[31] M. O'Donnell, W.E. Engeler, J.T. Pedicone, A.M. Itani, S.E. Noujaim, R.J. Dunki-Jacobs, W.M. Leue, C.L. Chalek, L.S. Smith, J.E. Piel, R.L. Harns, K.B. Welles, and W.L. Hinrichs. Real-time phased array imaging using digital beam forming and autonomous channel control. In *Ultrasonics Symposium*, pages 1499–1502, 1990.

[32] M. O'Donnell and L. J. Thomas. Efficient synthetic aperture imaging from a circular apertur with possible application to catheter-based imaging. *IEEE Trans. Ultrason. Ferroelec. Freq. Contr.*, **39**(3):366–380, 1992.

[33] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing.* Prentice-Hall, 1989.

[34] Allan D. Pierce. *Acoustics : An Introduction To its Physical Principles and Application.* Woodbury, N.Y. : Acoustical Society of America, 1994.

[35] W.A. Reckwerdt, S.O. Ishrak, and W. Bao. *Method and apparatus for performing imaging.* US Patent, 5379642, Jan. 1995.

[36] William D. Richard and R. Martin Arthur. Real-time ultrasonic scan conversion via linear interpolation on oversampled vectors. *Ultrasonic Imaging*, **16**:109–123, 1994.

[37] Bjørn A.J. Angelsen. *Waves, Signals and Signal Processing in Medical Ultrasonics.* Trondheim : Department of Physiology and Biomedical Engineering, Norwegian University of Science and Technology, 1996.

[38] D.E. Robinson and P.C. Knight. Interpolation scan conversion in pulse-echo ultrasound. *Ultrasonic Imaging*, **4**:297–310, 1982.

[39] J. Shen, H. Wang, C. Cain, and E. S. Ebbini. Post-beamforming processing technique for enhancing conventional pulse-echo ultrasound imaging contrast resolution. In *IEEE Ultrasonics Symposium Proceedings*, pages 1319–1322, 1995.

[40] K. Kirk Shung, Michael B. Smith, and Benjamin Tsui. *Principles of Medical Imaging.* Academic Press, 1992.

[41] Stephen W. Smith. A contrast-detail analysis of diagnostic ultrasound imaging. *Am. Assoc. Phys. Med*, **9**(1):4–12, 1982.

[42] S.W. Smith, G.E. Trahey, and O.T. von Ramm. Two-dimensional arrays for medical ultrasound. *Ultrasonic Imaging*, **14**:213–233, 1992.

[43] R.N. Thomson. Transverse and longitudinal resolution of the synthetic aperture focusing technique. *Ultrasonics*, pages 9–15, 1984.

[44] Gregg E. Trahey and Levin F. Nock. Synthetic receive aperture imaging with phase correction for motion and for tissue inhomogeneities — part I: Basic principles. *IEEE Trans. Ultrason. Ferroelec. Freq. Contr.*, **39**(4):489–495, 1992.

[45] Gregg E. Trahey and Levin F. Nock. Synthetic receive aperture imaging with phase correction for motion and for tissue inhomogeneities — part II: Effects of and correction for motion. *IEEE Trans. Ultrason. Ferroelec. Freq. Contr.*, **39**(4):496–501, 1992.

[46] Daniel H. Turnbull, Paul K. Lum, Andrew T. Kerr, and F. Stuart Foster. Simulation of B-scan images from 2-D transducer arrays: Part I – methods and quantitative contrast measurements. *Ultrasonic Imaging*, **14**:323–343, 1992.

[47] M.J. Vonesh, C. Kequing, and M. Radvany. Digital subtraction for noise reduction in intravascular data. In *Proc. IEEE Conf. Computers in Cardiology*, pages 329–332, 1991.

[48] William F. Walker and Gregg E. Trahey. *Real-Time Synthetic Receive Aperture Imaging: Experimental Results.* Department of Biomedical Engineering, Duke University, Durham, NC.

[49] Lawrence J. Ziomek. *Fundamentals of Acoustic Field Theory and Space-Time Signal Processing.* CRC Press, 1995.