Abstract

In breast cancer diagnosis ultrasound examination provides useful additional diagnostic information. However, conventional ultrasound imaging methods lack both high spatial and temporal resolution. Therefore, During the reconstruction, We use Ultrasound Computer Tomography (USCT) to get a high resolution image, the object’s properties (e.g. different local sound speed and absorption) and the sensor’s properties (e.g. the angle dependent efficiency) should be considered. This thesis is concerning ultrasound computer tomography image reconstruction regarding object and sensor properties. After the reconstruction, both spatial and temporal resolution of the object image are improved. The algorithms are implemented in Matlab and C++.

Keyword: ultrasound, image, reconstruction, Matlab
Contents

Abstract 1

1 Introduction 1

2 Reconstruction algorithm background 3
  2.1 USCT Theory Aspects 3
    2.1.1 Ultrasound Computer Tomography 3
    2.1.2 Reconstruction Regarding Object Properties 5
    2.1.3 Reconstruction Regarding Sensor Properties 5
  2.2 Contrast and Sharpness of The Image 10
    2.2.1 contrast 10
    2.2.2 sharpness 10
  2.3 Bresenham Algorithm 11

3 Reconstruction 16
  3.1 Reconstruction Regarding Object Properties 16
    3.1.1 sound speed correction reconstruction 16
    3.1.2 Absorption Correction Reconstruction 21
  3.2 Reconstruction Regarding Sensor’s Properties 22
    3.2.1 Measurement Of Sensor’s Properties 22
    3.2.2 Reconstruction of Distance-Dependent Property 28
    3.2.3 Reconstruction of Angle-Dependent Properties 29

4 reconstructed objects 33
  4.1 Gelatine 33
    4.1.1 Reconstructed Images With Different Kind of Corrections 33
    4.1.2 Sharpness Comparison 34
    4.1.3 Contrast Comparison 39
    4.1.4 The Final Image 43
  4.2 Phantom 44

5 Discussion and Conclusion 47
  5.1 reconstruction regarding object properties 47
Chapter 1

Introduction

In breast cancer diagnosis, ultrasound examination provides useful additional diagnostic information. But the conventional ultrasound imaging methods lack both high spatial and temporal resolution. "The contrast and the resolution depends highly on the used frequency as well as the distance between the transducer array and interesting region within the breast, they are also highly operator-dependent and the image almost impossible to be reproduced" [1]. In Forschungszentrum Karlsruhe, a new method for Ultrasound Computer-Tomography (USCT) was invented, this USCT system was developed for breast imaging. It provides high quality three dimensional images with a high repetition rate.

This master thesis is to reach high quality image (with high spatial and temporal resolution) by reconstruction regarding object and sensor properties. The sharpness of the reconstructed image will be improved by reconstruction regarding the object’s sound speed property and the contrast of the reconstructed image will be improved by reconstruction regarding the object’s absorption property. Further more, the contrast of the reconstructed image will be improved by reconstruction regarding sensor properties includes the distance-dependent property and angle-dependent property reconstruction. Before the reconstruction, two images of object’s absorption and sound speed properties are given, and the sensor property curves (distance dependent property curve and angle dependent property curve) should be measured with a hydrophone, which can measure the ultrasound strength and convert it into electrical signal. The reconstruction is based on these images and curves and the reconstruction algorithms are implemented in Matlab and C++.

The work is described as follows:

Chapter 2 will gives the introduction of USCT, then basic principle of the new reconstruction method and principle of reconstruction regarding object properties are mentioned. Afterward, a useful function—Bresenham Algorithm, which
is used to find the object’s properties at every point in the interesting area is introduced. Also it will states the principle of reconstruction regarding sensor properties and how to calculate the most important two parameters (contrast and sharpness) of an image.

Chapter 3 gives statement of the reconstruction (sound speed correction reconstruction, absorption reconstruction), it will also describe the instrument for measuring sensor’s properties and how to measure the sensor’s properties. Finally, how to implement the sensor’s properties into the reconstruction is stated.

Chapter 4, in this chapter, the result of different reconstruction is shown and also their qualities are calculated. Their qualities are defined by sharpness and contrast of an image.

In the 5th chapter will discuss the problems appear in the reconstruction and some conclusion can be drawn finally.
Chapter 2

Reconstruction algorithm background

2.1 USCT Theory Aspects

2.1.1 Ultrasound Computer Tomography

Computer Tomography is a well-known technology which allows the non-destructive evaluation of the internal structure of objects. This technique has been used successfully in medical testing for many years because of its high resolution. "However, the conventional X-ray Computer Tomography needs many single profiles from different directions, which are required for the acquisition of a single image. Resulting in a much higher level of radiation exposure." [6] Therefore, diagnostic techniques without risks (such as ultrasound examination) are needed.

In breast cancer diagnosis, ultrasound examination provides useful additional diagnostic information. But conventional ultrasound imaging methods lack both high spatial and temporal resolution. "The resolution of conventional imaging systems is dependent on the extent of the aperture over which rays to and from the object are intercepted as well as on the distance of the aperture from the object.” [5] "Usually the scanner is operated manually and the tissue is deformed while getting as close as possible to regions of interest. Therefore image contents and image quality relate strongly to the operator’s experiences. Exact measurement of tissue structures like tumor size is not possible. Unlike the manually controlled linear transducer array, ultrasound computer tomography (USCT) can image a volume directly.” [1]

Ultrasound Computer Tomography systems consist of two main parts: a physical measuring part and a mathematical reconstruction algorithm. In the physical measuring part (Figure 2.1), several thousand ultrasound transducers are arranged in a cylindrical array around a tank containing the object to be examined in a
medium water. Every single transducer is small enough to emit an almost spherical sound-wave. While one transducer is transmitting pulse, all others receive simultaneously. The received signal’s amplitudes were presented in dependence to the propagation time, which is called A-scan signal. Afterwards a different transducer emits the next pulse. For volume reconstruction, every transmitted, scattered and reflected signal is used for the mathematical reconstruction algorithm. The physical measuring part yields integral values of the wanted local parameters (for example: amplitude of the A-scan signal, time of the A-scan signal), all of these parameters are recorded by the computer and the mathematical reconstruction algorithm calculates the local concentrations from the recorded raw data.

Figure 2.1: **USCT system, shown in 2D. A cylinder of ultrasound transducers surround the object. One transducer emits a short ultrasound pulse, all other transducers receive simultaneously. The A-scan at the right side shows the directly transmitted and scattered signals**

Since both sharpness and contrast of the reconstructed images are not so good with other methods. Therefore, a creative method is referred, which can obtain reproducible images with higher resolution and tissue contrast. Reconstruction with this method is based on reconstructed results of some other methods:
- a low resolution image of the local sound velocities (using the transmitted tomography)
- a low resolution image of the local absorption (also using transmission tomography)

This new method combines all two different physical properties (sound velocities and absorption) to gain a high resolution diffraction image of the object.
Meanwhile, some factors (the angle and distance dependent efficiency of the transducers) which will affect the final image’s contrast should be also taken into account. Figure 2.2 shows the basic principle of this new method.

### 2.1.2 Reconstruction Regarding Object Properties

The new method uses diffraction tomography for reconstruction. Firstly, let us see a reconstructed image by this method (figure 2.3).

The reason for the bad sharpness is that the ultrasound passes through the water and the object with different speeds, but the former algorithm only uses a single sound speed. Therefore, the travelling time which the ultrasound transmits from the transducer to each pixel, then reflects back to the receiver is a little shifting (compare to the real travelling time). Which means, the corresponding amplitude for each pixel in the transmission signal is more or less shifting. Then the brightness at each pixel shows more or less shifting. The low contrast is caused by not considering the ultrasound absorption by each pixel at all.

The method to get a higher sharpness image is using a sound speed map, which shows correct sound speed at each corresponding pixel of the reconstructed image. Also, to get a higher contrast image is introducing an absorption map, which has a correct absorption at each corresponding pixel. By calculating average sound speed along the line which the ultrasonic pulse travels through the correct sound speed map, we can get the exact time which the beam travel through each pixel \( \left( \int_{L} \frac{dL}{C(L)} \right) \), and the exact ultrasound absorption rate of each pixel by using the absorption map.

These two maps are ready before the "modified" reconstruction.

### 2.1.3 Reconstruction Regarding Sensor Properties

In order to know the ultrasonic sensor’s properties, some basic knowledge of the sensor is given as follows:

**Near Field and Far Field**

Ultrasonic waves are generated by an oscillator, which is already included in the ultrasound sensor, the ratio of oscillator diameter \( D \) to wavelength \( \lambda \) determines the spread of the interference field.
CHAPTER 2. RECONSTRUCTION ALGORITHM BACKGROUND

Figure 2.2: Basic principle of diffraction tomography. The figure(right) is the reconstructed image which is gained from the obtained A-scan signals (for example, the left A-scan signal is a typical A-scan signal sending by the transducer(sending position) and detected by the receiver(receiving position) in the right image). For a more clear explanation, we just consider one sending position, a certain pixel \( (x,y) \) \( (x \) and \( y \) are the coordinates) and one receiving position in the right image(reconstructed image). At the sending position, transducer sends ultrasound pulse, the pulse is reflected by the pixel \( (x,y) \) and then detected by the receiver at the receiving position. The distance between \( (x,y) \) and the sending position is \( a \), the distance between \( (x,y) \) and the receiving position is \( b \), so the whole distance of the pulse travelling from the transducer to \( (x,y) \), and then reflected to the receiver is \( l \) \( (l=a+b) \). If we know the sound speed \( c \) along this travelling line, the exact travelling time \( t_{ab} \) can be obtained by \( l/c \). According to the A-scan signal(Amplitude(P)-Time) and the travelling time \( t_{ab} \), we can get the corresponding reflected signal’s amplitude \( P(t) \) at \( (x,y) \). The sum of all corresponding amplitude values at a certain pixel\( (x,y) \) from each A-scan signal:

\[
f (x,y) = \sum_{\text{sendingpositions}} \sum_{\text{receivingpositions}} (|A(P(t))|)
\]

shows the reflection strength at a certain pixel \( (x,y) \) (in this formula: \( A \) is the analytic continuation signal of each A-scan signal and \( t = \int_{\text{sendingposition}}^{(x,y)} \frac{dl}{c(t)} + \int_{(x,y)}^{\text{receivingposition}} \frac{dl}{c(t)} \), every pixel’s \( f(x,y) \) value is corresponding to each pixel’s brightness which composes the reconstructed image.
CHAPTER 2. RECONSTRUCTION ALGORITHM BACKGROUND

Figure 2.3: the left image is the photo of the Gelatine model (3D) and the right image is the reconstructed image of the model’s top view section. we can see, the sharpness and contrast of the reconstructed image is very low. For example, one thin membrane was reconstructed to two parallel lines and some membranes are almost invisible.

There are extensive fluctuations near the oscillator, known as the near field. These high and low pressure areas are generated because the crystal is not a point source of sound pressure, but rather a series of high and low pressure waves which are joined into a uniform front at the end of the near zone. Because of acoustic variations within a near field, it can be extremely difficult to accurately evaluate the object when they are positioned within this area.

The ultrasonic beam is more uniform in the far field, where the beam spreads out in a pattern originating from the center of the transducer.

The transition between these zones occurs at a distance, \( N \), and is sometimes referred to as the ”natural focus” of a flat ( or unfocused ) transducer. The near/far distance, \( N \), is significant because amplitude variations that characterize the near field change to a smoothly declining amplitude at this point. This area just beyond the near field is where the sound wave is well behaved and at its maximum strength.

The near-field length \( N \) is given by:
where $N$ is the distance to the field point, $D$ is the largest dimension of the rectangle oscillator and $\lambda$ is the wavelength, which equals $c/f$ (sound speed/frequency). In most practical cases, the diameter is much larger than the wavelength and we can simplify Eq.(2.1) as

$$N \approx \frac{D^2}{4 \times \lambda}$$

Distance-dependent Property

Since the near-field has quite a complicated structure, and the far-field’s structure is much simpler, it’s better to put the object to be reconstructed in the far-field, where the more uniform ultrasonic beam is present. However, the ultrasound beam which travels through the far-field does not have the unique strength everywhere. The strength decreases with the increasing of the distance between sender and the measuring point (figure 2.4). Because of
the weak detected signal at a very long distance which may cause a small amplitude (brightness), it is necessary to restore the original signal (at the sending position) everywhere.

**Angle-dependent Property**

During measurement, we found the amplitude of the received signal fluctuated with the angle-change of the receiver at the same distance between sender and receiver, therefore, the amplitude of the detected signal therefore depends not only on the distance, but also on the angle. Actually, the signal’s amplitude is also fluctuating when sending angle changed. (Figure 2.5)

![Figure 2.5: Sender and Receiver’s Angle Property](image)

Figure 2.5 shows the sender’s structure, from which we can further discuss why the amplitude changes with the shifting of receiving and sending angle.
Figure 2.6: Sender's Structure (for example, ultrasonic transceiver developed by Georg Göbel which has 8 elements), the piezo-ceramic material vibrates and emits ultrasound waves, which pass through the adhesive layer and the substrate, then were transmitted into the water.

Figure 2.7 (on page 11) shows the reason why the transmitting signal's amplitude fluctuates with the change of sending(receiving) angle.

2.2 Contrast and Sharpness of The Image

2.2.1 contrast

Contrast and Sharpness are the most important two aspects of images, by which we can evaluate the quality of images. The definition of contrast is: The variation in the intensity of an image formed by an image system as black and white bars. Image contrast is defined as \((\text{Ca}-\text{Cb})/(\text{Ca}+\text{Cb})\), where \(\text{Ca}\) and \(\text{Cb}\) are the illuminance in the images of bright and dark bars respectively. (see Figure 2.8 and Figure 2.9)

2.2.2 sharpness

The sharpness of an image is the point spread function in the image. Figure 2.10 shows an image sharpness calculation method.
The reason why the transmitting signal’s amplitude fluctuates with the change of sending/receiving angle. The reason is: firstly, since there are minimum amplitudes at certain sending/receiving angles, interference exists. Theoretically, when the sender’s sending size is larger than \( \frac{\text{ultrasound wavelength}}{2} \), the precondition of interference fulfilled. Secondly, the sender sends ultrasound with a center frequency and a bandwidth, so there is also a scope of ultrasound wavelengths depends on their frequency. The size of sending area might be larger than \( \frac{\text{ultrasound wavelength}}{2} \) when the sending frequency is very high. Thirdly, when the sending element vibrates and sends ultrasound waves, a certain substrate whose size is much larger than the size of sending element also vibrates (For example, at a certain position A, the substrate sends out waves(CA,BA,DA,EA……) which causes an interference when those waves’ phase difference (\( \phi \)) equal \( \pi \) (or \( k\pi \)). Actually, it is very difficult to get a exact phase difference which equals \( \pi \) (or \( k\pi \)). Normally, when the sum is close to \( \pi \) (or \( k\pi \)), there is a weakest amplitude, that is exactly the valley in our angle dependent signal.)

\[ \Phi = \pi \]

Figure 2.7: The reason why the transmitting signal’s amplitude fluctuates with the change of sending/receiving angle. The reason is: firstly, since there are minimum amplitudes at certain sending/receiving angles, interference exists. Theoretically, when the sender’s sending size is larger than \( \frac{\text{ultrasound wavelength}}{2} \), the precondition of interference fulfilled. Secondly, the sender sends ultrasound with a center frequency and a bandwidth, so there is also a scope of ultrasound wavelengths depends on their frequency. The size of sending area might be larger than \( \frac{\text{ultrasound wavelength}}{2} \) when the sending frequency is very high. Thirdly, when the sending element vibrates and sends ultrasound waves, a certain substrate whose size is much larger than the size of sending element also vibrates (For example, at a certain position A, the substrate sends out waves(CA,BA,DA,EA……) which causes an interference when those waves’ phase difference (\( \phi \)) equal \( \pi \) (or \( k\pi \)). Actually, it is very difficult to get a exact phase difference which equals \( \pi \) (or \( k\pi \)). Normally, when the sum is close to \( \pi \) (or \( k\pi \)), there is a weakest amplitude, that is exactly the valley in our angle dependent signal.)

\[ \Phi = \pi \]

2.3 Bresenham Algorithm

In this image reconstruction, we use Bresenham Algorithm to find the path (pixels) which ultrasound pulse travels from sending position to receiving position in the image. The basic ”line drawing” algorithm used in computer graphics is called Bresenham Algorithm. This algorithm was developed to draw lines on digital plotters, it is also wide-spread used in computer graphics. The algorithm is fast - it can be implemented with integer calculations only - and very simple to describe.

Assume a line with the start point \((x_1,y_1)\) and the ending point \((x_2,y_2)\) in device space. If \(dx = x_2 - x_1\) and \(dy = y_2 - y_1\), we define the driving axis (CX) to be the x-axis if \(|dx| \geq |dy|\), and the y-axis if \(|dy| > |dx|\). The CX
Contrast Calculation. There is an image, draw a line cross the object, the information about the brightness and darkness along the line can be obtained in figure 2.9

is used as the “controlling axis” for the algorithm and is the axis of maximum movement. Within the main loop of the algorithm, the coordinate corresponding to the CX is incremented by one unit. The coordinate corresponding to the other axis (usually denoted the passive axis or PX) is only incremented as needed.

The best way to describe Bresenham algorithm is to work through an example. Consider the following example, in which we wish to draw a line from (0, 0) to (5, 3) in device space (figure 2.11 on page 15)

Bresenham’s algorithm begins with the point (0, 0) and “illuminates” that pixel. Since $x$ is the DA in this example, it then increments the $x$ coordinate by one. Rather than keeping track of the $y$ coordinate (which increases by $m = dy/dx$, each time the $x$ increases by one), the algorithm keeps an error bound $\epsilon$ at each stage, which represents the negative of the distance from the point where the line exits the pixel to the top edge of the pixel (see the figure 2.11). This value is first set to $m - 1$, and is incremented by $m$ each time the $x$ coordinate is incremented by one. If $\epsilon$ becomes greater than zero, we know that the line has moved upwards one pixel, and that we must increment our $y$ coordinate and readjust the error to represent the distance from the top of the new pixel - which is done by subtracting one from $\epsilon$

The reader can examine the above illustration and the following table (figure
Figure 2.9: Contrast Calculation. This figure shows pixels’ brightness value changing along the bright line in figure 2.8. By sampling several peaks, we can calculate their contrast separately, then average them, a final image contrast can be obtained. Of course, for a more accurate calculation, we can draw several parallel lines instead of one line cross the object in figure 2.8 and analyze them.

Assuming that the DA is the $x$-axis, an algorithmic description of Bresenham’s algorithm is as follows:

The start point $(x_1,y_1)$ and the ending point $(x_2,y_2)$ are assumed not equal and integer valued. $\varepsilon$ (in the following lines $\varepsilon$ will be written as epsilon) is assumed to be real.

\[
\begin{align*}
\text{dx} &= x_2-x_1; \\
\text{dy} &= y_2-y_1; \\
\text{m} &= \text{dy}/\text{dx}; \\
\text{j} &= y_1; \\
\varepsilon \text{psilon} &= \text{m}-1; \\
\text{for i = x1:1:(x2-1)} & \quad \text{disp (i,j);} \\
& \quad \text{if (epsilon >=0)} \\
& \quad \quad \text{j} = \text{j} + 1;
\end{align*}
\]
Figure 2.10: Sharpness Calculation Method. This curve shows also the pixels’s brightness value changing along a certain bright line in an image (like figure 2.8). For example, there are two peaks: peak1 and peak2. At peak1, the width at its "half-peak" is \( P_b \) and its height is \( P_a \), the sharpness of peak1 is: \( \frac{P_b}{P_a} \). The same as peak2, sharpness of peak2 is: \( \frac{P_d}{P_c} \). By comparing same peaks of different images, we can compare their sharpness.

```
epsilon = epsilon - 1;
end;
epsilon = epsilon + m;
end;
```

All algorithms presented in these notes assume that \( dx \) and \( dy \) are positive. If this is not the case, the algorithm is virtually the same except for the following:

- \( \epsilon \) is calculated using \( \frac{dy}{dx} \)
- \( \epsilon \) is calculated using \( \frac{dy}{dx} \)
- \( x \) and \( y \) are decremented (instead of incremented) by one if the sign of \( dx \) or \( dy \) is less than zero, respectively [8].
Figure 2.11: Bresenham Example

<table>
<thead>
<tr>
<th>$(x, y)$</th>
<th>$\epsilon$</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(0, 0)$</td>
<td>-0.4</td>
<td>illuminate pixel $(0, 0)$</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>increment $\epsilon$ by 0.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>increment $x$ by 1</td>
</tr>
<tr>
<td>$(1, 0)$</td>
<td>0.2</td>
<td>illuminate pixel $(1, 0)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>since $\epsilon &gt; 0$</td>
</tr>
<tr>
<td></td>
<td>-0.8</td>
<td>increment $y$ by 1</td>
</tr>
<tr>
<td></td>
<td>-0.2</td>
<td>decrement $\epsilon$ by 1</td>
</tr>
<tr>
<td>$(2, 1)$</td>
<td>-0.2</td>
<td>illuminate pixel $(2, 1)$</td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>increment $\epsilon$ by 0.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>increment $x$ by 1</td>
</tr>
<tr>
<td>$(2, 1)$</td>
<td>-0.2</td>
<td>illuminate pixel $(2, 1)$</td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>increment $\epsilon$ by 0.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>increment $x$ by 1</td>
</tr>
<tr>
<td>$(3, 1)$</td>
<td>0.4</td>
<td>illuminate pixel $(3, 1)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>since $\epsilon &gt; 0$</td>
</tr>
<tr>
<td></td>
<td>-0.6</td>
<td>increment $y$ by 1</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>decrement $\epsilon$ by 1</td>
</tr>
<tr>
<td>$(4, 2)$</td>
<td>0.0</td>
<td>illuminate pixel $(4, 2)$</td>
</tr>
</tbody>
</table>

Figure 2.12: This table shows the complete operation of Bresenham Algorithm on the above example
Chapter 3

Reconstruction

This chapter will give statement of the reconstruction process regarding object (for example: gelatine)’s properties (sound speed correction and absorption reconstructions), how to measure sensor’s properties (distance and angle dependent properties) and implement them into the reconstruction algorithm.

3.1 Reconstruction Regarding Object Properties

3.1.1 sound speed correction reconstruction

artificial sound speed model

For reconstructing gelatine, since there is some error between the real sound speed map and the real sound speed, an simplified artificial sound speed map is set up for a more precise reconstruction. (figure 3.1).

This model is assumable because:

Firstly, the membranes in the object so thin that their influences on the sound speed of gelatine can be ignored.

Secondly, different amount density of contrast agency in each department does not influence the sound speed.

Therefore, the whole object can be looked as a homogeneous gelatine object.

The model only divides the whole systems into two parts: one is water (black part) and the other is gelatine (white part). The problem is: how to calculate
Figure 3.1: Artificial Model for Sound Speed Reconstruction Map. In this model, the white part shows sound speed in the gelatine (each department in the object are ignored) and the black part shows sound speed in the water.

their sound speed separately?

The method for calculating the sound speed in gelatine (we already know the sound speed in the water) at this case is:

Firstly, load the reconstructed image (high resolution image using reflected signal but only with a unique sound speed), use a software CorelDRAW to draw another image (newpicture) with the same size and structure, find the approximate border between gelatine and water, then set value in the gelatine area to 1 and in the water area to 0.

Secondly, select a sending position and a receiving position which are already known exactly (their coordinate and the distance (Dis) between them) in the reconstruction system. Get the recorded A-scan signal (from this sending position to this receiving position) and find corresponding time of the maximum value. This time is the ultrasound pulse's travelling time from the sending position to the receiving position (Ttotal).

Then use a Matlab function called improfile which can get the pixels information (their coordinates and their values) through the straight line between the
corresponding sending position pixel and receiving position pixel in the newpicture, which was set up at the first step.

According to these pixel’s values, it’s easy to find the pixels number in the gelatine (just count the number of 1) and water (count the number of 0) along this straight line, this means, we know the line’s length in the water (LenW) and the gelatine (LenG).

Since we know the sound speed in water (Cwater) and the total travelling time (Ttotal), the time spends in the water is \( T_{\text{water}} = \frac{\text{LenW}}{\text{Cwater}} \) the time spend in the gelatine is: \( T_{\text{gelatine}} = T_{\text{total}} - T_{\text{water}} \). Then the sound speed in the Gelatine is:

\[
C_{\text{gelatine}} = \frac{\text{LenG}}{T_{\text{gelatine}}}
\]

Finally, a simple sound speed model was built up, then we can use it for sound speed correction reconstruction.

**implementation of Bresenham Algorithm**

The basic idea to get the exact ultrasound travelling time from a certain sending position to a certain receiving position is summing up the sound travelling time through each pixels along the corresponding ultrasound pulse’s travelling path. Therefore, Bresenham Algorithm is implemented to gain those pixels and the sound speed values at those pixels in the artificial model of sound speed map.

(Figure 3.2) shows an ultrasound pulse passes through pixels with different speeds. The total travelling time \( t \) is calculated by:

\[
t = \int_A^B \frac{dL}{C(L)}
\]  

(3.1)

Here we use a pixel’s length as the unit for this calculation. However, the lengths which the beam passes through each pixel are different and it’s very difficult to calculate lengths for each pixel. Therefore, an average pixel length is accepted. It is given by:

\[
\text{average pixel length} = \frac{\text{total length which the beam passes through}}{\text{pixels number}}
\]

(3.2)

As we assuming, an ultrasonic pulse passes through every pixel with different speed but with the same length, then we can obtain an average sound speed for
Figure 3.2: Implement Bresenham Algorithm in Reconstruction

the whole length. This sound speed is given by:

\[
\text{average sound speed (Cave)} = \frac{\sum_{i=0}^{N} \text{soundspeed of each pixel along the line}(C_i)}{\text{pixels number}}
\]

(3.3)

By using this average sound speed (Cave), the corresponding point (the exact brightness of the pixel) in the transmission signal is found.

During the reconstruction, the program which part takes most of the time is the Bresenham algorithm. In Matlab, there is a function called improle which can find all pixels along the straight line from one pixel to another pixel. But this function costs a lot of time in Matlab because it does not calculate as a matrix and Matlab is only famous and quick for the matrix calculation.

For saving calculation time, the Bresenham algorithm in C++ language is developed, which takes much less time in loop-calculation than Matlab.

In order to saving more time, we analyze the reconstruction image (Figure 3.3)

Consider one pixel, the path travels from sending position to receiving position can be divided into two paths: path1 is from sending position to the pixel and path2 is from the pixel to the receiving position. When the receiver moves
to next receiving position, we can find only path2 changes, path1 is the same as former’s. The trick is: During the calculation of one sending position, we can just calculate path1 once at the first receiving position and save it. For the other receiving positions, what we need to do is calculating path2 and load path1. Then we can get the whole path which ultrasonic signal travels through.

Above all, the correct amplitude is obtained in the A-scan signal due to the correct transmission time.

**implementation of the real sound speed map**

In order to get a more “real” sound speed reconstruction image, the real sound speed map is used after reconstruction with the artificial sound speed map.

As it mentioned, there exist some errors between the real sound speed and sound speed in the referred real sound speed map. We can find the reconstructed image using referred real sound speed map is not so good (Figure 4.2 on page 35). Therefore, a sound speed modification was made on the referred real sound speed map before this reconstruction. The reconstruction with this more correct sound speed map (shown in Figure 4.3 on page 35) is obviously better than the former one.
3.1.2 Absorption Correction Reconstruction

Absorption correction reconstruction is for a better contrast image. The procedure of absorption correction reconstruction is almost the same as it of sound speed correction reconstruction.

Firstly, find the corresponding sending position(Sp), a certain interesting pixel(Pixel) and the corresponding receiving position(Rp) in the referred low resolution absorption map. Then using Bresenham Algorithm to obtain the absorption value at each pixel along the straight lines which are from Sp to Pixel, and from Pixel to Rp.

Then, calculate the average absorption by:

\[
\text{average absorption (Aa)} = \frac{\sum_{i=0}^{N} \text{absorption of each pixel (Ai)}}{\text{pixels number}}
\]  

(3.4)

Since the average absorption is defined as:

\[
\text{average absorption} = \frac{\text{the attenuation of the transmission signal}}{\text{distance}}
\]  

(3.5)

\[
= \ln \left( \frac{\text{signal amplitude without object (voltage)}}{\text{signal amplitude with object (voltage)}} \right)
\]  

(3.6)

and the factor be compensated for the transmission signal is defined as:

\[
\text{compensation factor} = \frac{\text{signal amplitude without object (voltage)}}{\text{signal amplitude with object (voltage)}}
\]  

(3.7)

With (3.6) and (3.5)

\[
\text{compensation factor} = e^{\text{the attenuation of the signal}} = e^{\text{average absorption (Aa) \times distance}}
\]  

(3.8)

After the absorption reconstruction, the value of each pixel is:

\[
f(x, y) = \sum_{\text{sending positions}} \sum_{\text{receiving positions}} (|A(P(t))| \times \text{compensation factor})
\]  

(3.9)
3.2 Reconstruction Regarding Sensor’s Properties

3.2.1 Measurement Of Sensor’s Properties

Since we do not know the exact properties of the ultrasonic sensor before reconstruction. A special experimental system is set up measurement of the sensor’s properties.

The experimental system is set up as follows:

![Measurement System Diagram](image)

Figure 3.4: Measurement System

The rectangular wave is generated from the signal generator and converted into pulse by the high voltage supply as well as amplified by the power amplifier,
then the pulse is transmitted to the piezo and stimulates it. The ultrasonic pulse sent from the piezo is received by the hydrophone which can detect the ultrasonic pulse’s pressure and convert it into electrical signal, this electrical signal is also amplified by an amplifier and then recorded by the oscilloscope and computer. The oscilloscope and computer are also monitoring the signal generated by the signal generator.

Figure 3.5: Measuring Instrument

Instead of the cylinder, a rectangular plastic tub is used for the measuring vessel during this measurement because of its more useable size. And the angle and distance positions which are to be measured are printed on a plastic folio, this folio is pasted on the bottom of the plastic tub as a reference scale table. (Figure 3.6)

Then the piezo sender is fixed on one side of the tub as the hydrophone is also fixed at the same height on a mobile object, which is too heavy to sway in the water easily.
measurement

During the measurement, the hydrophone is shifted one angle after another at the same radius till a half circle measured, then it is shifted to another distance. The point is: the hydrophone is always opposite to the center of the piezo sender to make sure that it can receive the transmission signal as maximum as possible at every position and no received signal’s frequency shifting, therefore, all of the measured signals are comparable.

We obtained an A-scan signal which includes time, amplitude, temperature, angle etc. (Figure 3.7)

All of those amplitude values (at the same angle (90 degree)) compose a curve changed with the correspond distance. (Figure 3.11)

Since the instrument is controlled manually, an measuring angle gap as precise as one degree or the distance gap to 1 mm are almost impossible. So the angle gap is taken by 5 degree and the distance gap is taken by 10 mm. We can see the error between the theoretical distance-property curve and the measured distance-property curve.(see figure 3.12)

The angle and distance-dependent curves show the angle and distance properties of the ultrasonic sender in voltage. Actually, the hydrophone receives
CHAPTER 3. RECONSTRUCTION

Figure 3.7: Ultrasound Sensor’s Angle-Property Curve at 122 mm radium. This Sensor was developed by Forschungszentrum Karlsruhe. During the measurement, A-scan signals were measured at every 5 degree on a certain radium which is from the sender(center) to the hydrophone. This sensor’s angle dependent property curve shows the maximum amplitude value of every measured A-scan signal which fluctuates with the change of measuring angle.

Figure 3.8: Ultrasound Sensor’s Angle-Property Curve at 40 mm radium.

the ultrasonic pressure firstly, afterwards this pressure is converted into voltage through a circuit as follows: (Figure 3.13)
According to the calibration curve of the hydrophone, we can get a relationship between the pressure and the voltage signal which is recorded.
CHAPTER 3. RECONSTRUCTION

Figure 3.11: The measured sensor’s distance-property curve at 90 degree.

Figure 3.12: The theoretical Sensor’s distance-Property curve. As we talked in literature part, the curve should fulfil: \[ \text{amplitude (volt)} - n(\text{volt}) = \frac{m(\text{mm}^2 \text{volt})}{\text{distance (mm)}} \], (m and n are constants). By sampling several points in the measured curve, we can get the average m value. Then this average m is used for theoretical sensor distance-dependent curve drawing.

\[
\text{Recorded Signal (volt)} = 1580 \times \text{Original Signal (volt)}
\]

\[
\text{Original Signal (volt)} = 2.73 \times 10^{-7} \times \text{Pressure (Pa)}
\]
CHAPTER 3. RECONSTRUCTION

Figure 3.13: Convert from pressure to voltage. The function of hydrophone is changing the ultrasonic pressure into voltage. Because this voltage is very small, an amplifier is used to amplify the original signal, the gain of this amplifier is 70dB (3160 times). Then this amplified signal is transferred to the oscilloscope by a cable. Since the cable resistance and the terminal resistance are both 50 ohm, therefore, amplitude of the signal detected by the oscilloscope is the half of the amplified signal’s amplitude because of the voltage division. Above all, the original signal (voltage) has been amplified 1580 times when it reaches the oscilloscope.

Then
\[
\text{Pressure (Pa)} = \frac{\text{Recorded Signal (Volt)}}{1580 \times 2.73 \times 10^{-7}}
\]  
(3.11)

3.2.2 Reconstruction of Distance-Dependent Property

In order to improve the image’s contrast further, the reconstruction algorithm concerning distance-dependent property is implemented. This algorithm is based on the measured distance curve (Figure 3.11).

Before using of this distance-dependent curve, some pre-procedures have been done:

1. Normalization, shown in Figure 3.14

2. Interpolation. Since during the sensor distance property measurement, we sampled the measuring points at every 10mm, an interpolation algorithm is implemented to calculate those normalized amplitude at the distances not sampled.

Above all, we can get the normalized amplitude for the reconstruction regarding sensor’s distance property. The compensation factor at a certain distance for the sensor’s distance dependent property is: \(\frac{1}{\text{normalized amplitude}}\).

After the absorption reconstruction and the reconstruction regarding sensor’s distance-dependent property with the sensor distance dependent curve (normalized curve), the value of at a certain pixel \((f(x,y))\) can be restored as follows:
Figure 3.14: Normalized Amplitude-Distance curve. In this figure, the real amplitude (Volt) has been already normalized, which means, the maximum value (max) of the real amplitude curve is taken as 1, the ratios of any amplitude (Volt) at a certain distance and the max is taken as the normalized amplitudes at this distance.

\[
f(x, y) = \sum_{\text{sending positions}} \sum_{\text{receiving positions}} (|A(P(t))|) \times (\text{absorption compensation factor}) \times (\text{distance compensation factor})
\]

3.2.3 Reconstruction of Angle-Dependent Properties

The reconstruction of angle-dependent properties can also improve the image’s contrast. Before this reconstruction, the measured angle-dependent curve is also normalized and then interpolated, like distance-dependent curve. (Figure 3.15)

Till now, the normalized amplitude corresponding to each angle (0° to 180°) is obtained, but how can we get the correct angle? (Figure 3.16) shows the problem:

The situation above is also fit for the sending position, the strength of sent ultrasound pulse at different positions varied with the change of the sending angle (between the direction vector and the vector of each position which is related to the sending position).

The solution is (Figure 3.17): find a reference vector \((x', y')\) which is always perpendicular to the direction vector but on its left side (look the following short program). Since the angle between the reference angle and the vector1 \((\phi)\) fulfills: \((0^\circ \leq \phi < 180^\circ)\). Then, \(\phi\) is unique:
Figure 3.15: *Normalized Angle Curve*

Figure 3.16: *Problem of Angle Calculation.* In this figure, the receiving position, the direction vector and the "pixel’s vector" (vector1) are known. How can we get the exact receiving angle $\phi_2$? Since there exists another vector (vector2) which is symmetric about the direction vector to vector1, if we simply calculate the angle $\beta$ which is equal to the angle $\alpha$, how can we choose the exact $\phi_2 = 90^\circ + \beta$ or $\phi_2 = 90^\circ - \alpha$?

$$\cos \phi = \frac{\text{reference vector} \cdot \text{vector1}}{|\text{reference vector}||\text{vector1}|}$$
$$= \frac{x' \times x_1 + y' \times y_1}{\sqrt{(x')^2 + (y')^2} \sqrt{x_1^2 + y_1^2}}$$

(3.12)
And the angle $\phi$ is exact the angle we wanted.

Figure 3.17: Solution of Angle Calculation

The method to get the perpendicular vector on the left side of a original vector is:

Figure 3.18: Perpendicular Vector Calculation

Assume there is a known vector $\vec{a} (Xa,Ya)$ and its perpendicular vector on the left side is $\vec{b} (Xb,Yb)$, the program to find $\vec{b}(Xb,Yb)$ is as follows:
if (Xa > 0) then {
    Yb = +1;
    Xb = - Ya/Xa;
}
else {
    if(Xa < 0) then {
        Yb = -1;
        Xb = Ya/Xa;
    }
    else {
        if ((Xa == 0) and (Ya < 0)) then {
            Yb = 0;
            Xb = 1;
        }
        else {
            if ((Xa == 0) and (Ya > 0)) then {
                Yb = 0;
                Xb = -1;
            }
        }
    }
}

Above all, we can get the compensation factor for the reconstruction regarding sensor’s angle dependent property angle compensation factor. The compensation factor for a certain pixel at an exact sending position and receiving position is: normalized amplitude (the normalized amplitude is the amplitude corresponds to the pixel’s sending or receiving angle at the normalized sensor’s angle-dependent curve.

After the absorption reconstruction, the reconstruction regarding sensor’s distance-dependent property and sensor’s angle-dependent property, the value of at a certain pixel \( f(x,y) \) can be restored as follows:

\[
 f(x, y) = \sum_{\text{sending positions}} \sum_{\text{receiving positions}} |A(P(t))| \times (\text{absorption compensation factor}) \times (\text{distance compensation factor}) \times (\text{angle compensation factor})
\]

In this chapter, in order to get a higher sharpness and contrast image than other old methods, different algorithms are implemented. Result of the reconstruction will be shown in next chapter.
Chapter 4

reconstructed objects

This chapter will show reconstructed images of two kinds of object. One is gelatine, the other is phantom.

4.1 Gelatine

4.1.1 Reconstructed Images With Different Kind of Corrections

In this part, I only used the image reconstructed from the signal of one receiving element (element 5) for a simple explanation.

As I mentioned, firstly a sound speed model was used for my first image reconstruction (see Figure 4.1 on page 34).

After a better image is obtained with the artificial sound speed model, the real sound speed map which was created with Radon-Transformation [4] was used for a "true" reconstruction. The reconstructed image is as follows: (Figure 4.2 on page 35)

Since the quality of this image is not so good, a corrected sound speed map is used for reconstruction. With this map, the quality is improved rapidly. (Figure 4.3 on page 35)

The contrast is improved further in the following image since absorption is taken into account: (Figure 4.4 on page 36)

By introducing the sensor’s distance-dependent property into reconstruction: (Figure 4.5 on page 36), we can see there is almost no difference between the
CHAPTER 4. RECONSTRUCTED OBJECTS

4.1.2 Sharpness Comparison

In order to analyze these two images (before and after sound speed correction) more quantitative, their sharpness and contrast are calculated.

In each image, six bright lines were sampled for sharpness compare. (see Figure 4.7) They are at the same positions in these two images.

For a comparison example, pixels along each of the fourth line in figure 4.7
Figure 4.2: Reconstructed Image with Real Sound Speed Map. This image is even if worse than the former one which reconstructed with the artificial sound speed model. The reason is that the sound speed in this given sound speed map is not correct. (See chapter Discussion and Conclusion)

Figure 4.3: Reconstructed Image with Corrected Sound Speed Map Correction

compose the following curves separately (Figure 4.8). In this figure, we can eas-
Figure 4.4: *Reconstructed Image with Sound Speed and Absorption Correction*

Figure 4.5: *Reconstructed Image with Sound Speed, Absorption and Sensor’s Distance Property Correction*

ily choose three peaks for comparison: peak1, peak2 and peak3 which shows the inner membranes in the gelatine object (in figure 4.7).

According to the sharpness definition, we can calculate the sharpness of each
CHAPTER 4. RECONSTRUCTED OBJECTS

Figure 4.6: Reconstructed Image with Sound Speed, Absorption, Sensor’s Distance Property and Sensor’s Angle Property Correction

Figure 4.7: The left image is the reconstructed image before sound speed correction and the right one is the reconstructed image after sound speed correction.
Figure 4.8: Sharpness Calculation. This figure shows pixels’ brightness value changing along each of the fourth bright line in figure 4.7. The left one is sampled from the reconstructed image before sound speed correction and the right one is after sound speed correction. In this figure, y coordinate shows the pixel’s brightness and x coordinate shows the pixel’s number along the bright line.

peak along every sampled bright line in figure 4.7.

For peak1:

<table>
<thead>
<tr>
<th>bright line</th>
<th>before sound speed correction</th>
<th>after sound speed correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>bright line1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>bright line2</td>
<td>4.29</td>
<td>3</td>
</tr>
<tr>
<td>bright line3</td>
<td>15</td>
<td>5</td>
</tr>
<tr>
<td>bright line4</td>
<td>16</td>
<td>6</td>
</tr>
<tr>
<td>bright line5</td>
<td>13.75</td>
<td>6</td>
</tr>
<tr>
<td>bright line6</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>average</td>
<td>9.79</td>
<td>4.83</td>
</tr>
</tbody>
</table>

For peak2:
CHAPTER 4. RECONSTRUCTED OBJECTS

<table>
<thead>
<tr>
<th>bright line1</th>
<th>before sound speed correction</th>
<th>after sound speed correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>2.76</td>
<td></td>
</tr>
<tr>
<td>bright line2</td>
<td>22.7</td>
<td></td>
</tr>
<tr>
<td>1.18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bright line3</td>
<td>15.38</td>
<td></td>
</tr>
<tr>
<td>1.36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bright line4</td>
<td>13.53</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bright line5</td>
<td>14.4</td>
<td></td>
</tr>
<tr>
<td>0.95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bright line6</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>1.45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>average</td>
<td>16.17</td>
<td></td>
</tr>
<tr>
<td>1.62</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For peak3:

<table>
<thead>
<tr>
<th>bright line1</th>
<th>before sound speed correction</th>
<th>after sound speed correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1.5</td>
<td></td>
</tr>
<tr>
<td>bright line2</td>
<td>5.65</td>
<td></td>
</tr>
<tr>
<td>0.48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bright line3</td>
<td>3.66</td>
<td></td>
</tr>
<tr>
<td>2.15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bright line4</td>
<td>3.78</td>
<td></td>
</tr>
<tr>
<td>0.46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bright line5</td>
<td>5.52</td>
<td></td>
</tr>
<tr>
<td>0.45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bright line6</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>average</td>
<td>6.44</td>
<td></td>
</tr>
<tr>
<td>0.94</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Obviously, the smaller the value, the sharper the peak. Then we can draw a conclusion: image2 (with sound speed correction) is much sharper than image1 (without sound speed correction).

4.1.3 Contrast Comparison

comparison between images with unique sound speed and with both sound speed and absorption correction

Using the same six straight lines in these two images, just as discussion in sharpness comparison. We can also get three peaks (they are corresponding to the membranes in the gelatine objects) For peak1:

<table>
<thead>
<tr>
<th>line</th>
<th>without sound speed correction</th>
<th>with sound speed and absorption correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>line1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>line2</td>
<td>0.0369</td>
<td>0.0547</td>
</tr>
<tr>
<td>line3</td>
<td>0.0203</td>
<td>0.0331</td>
</tr>
<tr>
<td>line4</td>
<td>0.0290</td>
<td>0.0163</td>
</tr>
<tr>
<td>line5</td>
<td>0.0270</td>
<td>0.0223</td>
</tr>
<tr>
<td>line6</td>
<td>0.0174</td>
<td>0.0162</td>
</tr>
<tr>
<td>average</td>
<td>0.02612</td>
<td>0.02852</td>
</tr>
</tbody>
</table>

For peak2:
Figure 4.9: The left image is the reconstructed image before sound speed correction and the right one is the reconstructed image after sound speed and absorption correction.

<table>
<thead>
<tr>
<th>line</th>
<th>without sound speed correction</th>
<th>with sound speed and absorption correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>line1</td>
<td>0.0456</td>
<td>0.0887</td>
</tr>
<tr>
<td>line2</td>
<td>0.0410</td>
<td>0.1113</td>
</tr>
<tr>
<td>line3</td>
<td>0.0541</td>
<td>0.1170</td>
</tr>
<tr>
<td>line4</td>
<td>0.0647</td>
<td>0.1262</td>
</tr>
<tr>
<td>line5</td>
<td>0.0693</td>
<td>0.1334</td>
</tr>
<tr>
<td>line6</td>
<td>0.0401</td>
<td>0.0951</td>
</tr>
<tr>
<td>average</td>
<td>0.0525</td>
<td>0.11195</td>
</tr>
</tbody>
</table>

For peak3:

<table>
<thead>
<tr>
<th>line</th>
<th>without sound speed correction</th>
<th>with sound speed and absorption correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>line1</td>
<td>0.0594</td>
<td>0.1230</td>
</tr>
<tr>
<td>line2</td>
<td>0.0830</td>
<td>0.1883</td>
</tr>
<tr>
<td>line3</td>
<td>0.1267</td>
<td>0.2151</td>
</tr>
<tr>
<td>line4</td>
<td>0.1402</td>
<td>0.2103</td>
</tr>
<tr>
<td>line5</td>
<td>0.0924</td>
<td>0.1608</td>
</tr>
<tr>
<td>line6</td>
<td>0.0668</td>
<td>0.1350</td>
</tr>
<tr>
<td>average</td>
<td>0.09475</td>
<td>0.1721</td>
</tr>
</tbody>
</table>

Therefore, the contrast for reconstructed image without sound speed correction is:
the contrast for reconstructed image with sound speed and absorption correction is:

$$\frac{0.02612 + 0.0525 + 0.09475}{3} = 0.05779$$

Obviously, the latter one’s contrast is higher than the former one.

comparison between images with sound speed and absorption correction and with sound speed, absorption and sensor distance-property correction

We also use the same six bright lines in these two images for contrast comparison:

For peak1:
CHAPTER 4. RECONSTRUCTED OBJECTS

<table>
<thead>
<tr>
<th></th>
<th>image with sound speed and absorption correction</th>
<th>image with sound speed , absorption sensor distance-dependent property correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>line1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>line2</td>
<td>0.0547</td>
<td>0.0546</td>
</tr>
<tr>
<td>line3</td>
<td>0.0331</td>
<td>0.0331</td>
</tr>
<tr>
<td>line4</td>
<td>0.0163</td>
<td>0.0180</td>
</tr>
<tr>
<td>line5</td>
<td>0.0223</td>
<td>0.0244</td>
</tr>
<tr>
<td>line6</td>
<td>0.0162</td>
<td>0.0162</td>
</tr>
<tr>
<td>average</td>
<td>0.02852</td>
<td>0.02926</td>
</tr>
</tbody>
</table>

For peak2:

<table>
<thead>
<tr>
<th></th>
<th>image with sound speed and absorption correction</th>
<th>image with sound speed , absorption sensor distance-dependent property correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>line1</td>
<td>0.0887</td>
<td>0.0887</td>
</tr>
<tr>
<td>line2</td>
<td>0.1113</td>
<td>0.1204</td>
</tr>
<tr>
<td>line3</td>
<td>0.1170</td>
<td>0.1170</td>
</tr>
<tr>
<td>line4</td>
<td>0.1262</td>
<td>0.1262</td>
</tr>
<tr>
<td>line5</td>
<td>0.1334</td>
<td>0.1334</td>
</tr>
<tr>
<td>line6</td>
<td>0.0951</td>
<td>0.0951</td>
</tr>
<tr>
<td>average</td>
<td>0.11195</td>
<td>0.11346</td>
</tr>
</tbody>
</table>

For peak3:

<table>
<thead>
<tr>
<th></th>
<th>image with sound speed and absorption correction</th>
<th>image with sound speed , absorption sensor distance-dependent property correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>line1</td>
<td>0.1230</td>
<td>0.1231</td>
</tr>
<tr>
<td>line2</td>
<td>0.1883</td>
<td>0.1632</td>
</tr>
<tr>
<td>line3</td>
<td>0.2151</td>
<td>0.2151</td>
</tr>
<tr>
<td>line4</td>
<td>0.2103</td>
<td>0.2103</td>
</tr>
<tr>
<td>line5</td>
<td>0.1608</td>
<td>0.1691</td>
</tr>
<tr>
<td>line6</td>
<td>0.1350</td>
<td>0.1331</td>
</tr>
<tr>
<td>average</td>
<td>0.1721</td>
<td>0.16898</td>
</tr>
</tbody>
</table>

Therefore, the contrast for reconstructed image with sound speed and absorption correction is:

$$\frac{0.02926 + 0.11346 + 0.16898}{3} = 0.10419$$

the contrast for reconstructed image with sound speed absorption and sensor distance-dependent property correction is:

$$\frac{0.02852 + 0.11195 + 0.1721}{3} = 0.1039$$
We can see the contrast of the former one is slightly better than the latter one. The reason will be discussed in the discussion chapter.

4.1.4 The Final Image

The final image is the sum of the reconstructed image for each receiving element. (Figure 4.11), the first image is the reconstructed image with only a unique sound speed and without any other correction. The second one is reconstructed with sound speed, absorption and sensor’s distance property correction.

![Contrast Compare of the final Image](image)

Figure 4.11: Contrast Compare of the final Image

We can also easily get the contrast of these two images:

Contrast for the first image is 0.10298 and contrast for the second image is 0.05182.

Above all, the sharpness and contrast of the image are both improved by using all different reconstruction methods.
4.2 Phantom

Figure 4.12 shows two sound speed reconstruction images of phantom2, the first one is reconstructed with the unique sound speed and the second one is reconstructed with the sound speed map.

For sharpness comparison:

![Reconstructions with Unique Sound Speed and with Original Sound Speed Map](image)

Figure 4.12: Reconstructions with Unique Sound Speed and with Original Sound Speed Map. These two images are reconstructed by the signal received by the same receiving element. The left image is reconstructed image without sound speed correction and the right one is reconstructed image only with original sound speed map (without absorption map). We also sample three parallel lines in each image. According to these lines, we can also calculate both image’s sharpness and contrast like before.

<table>
<thead>
<tr>
<th>Peak1:</th>
<th>sharpness</th>
<th>reconstructed image without sound speed correction</th>
<th>reconstructed image with original sound speed map property correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>line1</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>line2</td>
<td>0.4</td>
<td></td>
<td>0.61</td>
</tr>
<tr>
<td>line3</td>
<td>0.25</td>
<td></td>
<td>0.62</td>
</tr>
<tr>
<td>average</td>
<td>0.383</td>
<td></td>
<td>0.577</td>
</tr>
</tbody>
</table>

Peak2:
Figure 4.13: Curves composed by pixel’s value along the second line in each image. The x-coordinate shows the pixel’s sequence (from top to bottom) and the y-coordinate show the pixel’s value.

<table>
<thead>
<tr>
<th>sharpness</th>
<th>reconstructed image without sound speed correction</th>
<th>reconstructed image with original sound speed map property correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>line1</td>
<td>0.2</td>
<td>0.37</td>
</tr>
<tr>
<td>line2</td>
<td>0.364</td>
<td>0.23</td>
</tr>
<tr>
<td>line3</td>
<td>0.37</td>
<td>0.43</td>
</tr>
<tr>
<td>average</td>
<td>0.311</td>
<td>0.343</td>
</tr>
</tbody>
</table>

Check these two tables, we can find the second image’s sharpness even worse than it of the first one. (the reason see Chapter Discussion and Conclusion).

For contrast comparison:

<table>
<thead>
<tr>
<th>reconstructed image without sound speed correction</th>
<th>reconstructed image with original sound speed map property correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.444</td>
<td>1.392</td>
</tr>
</tbody>
</table>

The second one’s contrast is also worse than that of the first image.

Then, we tried to correct the sound speed map, the following image is the reconstructed image with the corrected sound speed map.

<table>
<thead>
<tr>
<th>peak1</th>
<th>peak2</th>
</tr>
</thead>
<tbody>
<tr>
<td>line1</td>
<td>0.32</td>
</tr>
<tr>
<td>line1</td>
<td>0.28</td>
</tr>
<tr>
<td>line1</td>
<td>0.25</td>
</tr>
<tr>
<td>average</td>
<td>0.283</td>
</tr>
</tbody>
</table>
Figure 4.14: Reconstruction with Unique Sound Speed and Corrected Sound Speed Map

It’s contrast is: 1.416.

After the sound speed map correction, the sharpness and contrast of the reconstructed image (Figure 4.14) are both improved, but the quality is still not so good enough. Therefore, the accuracy of the sound speed map decides largely the sharpness and contrast of the result image.
Chapter 5

Discussion and Conclusion

In this chapter, I will discuss the reconstruction algorithm, the measurement, reconstructed results and reasons which cause the error.

5.1 reconstruction regarding object properties

5.1.1 Bresenham Algorithm

Actually, the Bresenham Algorithm takes most of the time during reconstruction. If we set the algorithm into Matlab which calculates all pixels along the path from all sending positions to all receiving positions (in the program, this part is mainly a improfile function included into two for-loops ), It takes long time to run the program because Matlab is not good at for-loop calculation. Therefore, we introduce a C++ procedure which can be called by Matlab to take over the heaviest task.

It takes 1382.9 seconds ( around 23minutes ) to finish the for-loop algorithm for one sending and receiving positions in Matlab (on the computer with pentium 4 2.2G Hz processor and 1G byte RAM ). therefore totally it takes:

\[
23 \frac{\text{minutes}}{\text{position}} \times 9100 \text{positions} = 209300 \text{minutes} \approx 145 \text{days} \quad (5.1)
\]

But in C++, only 0.96 second was taken for calculating one sending position and one receiving position, so for all measured positions:
CHAPTER 5. DISCUSSION AND CONCLUSION

Figure 5.1: Error of Bresenham Algorithm

\[
0.96 \text{ seconds/position} \times 9100 \text{ positions} = 8736 \text{ seconds} = 2.42 \text{ hours}
\]

(5.2)

Obviously, C++ takes much shorter time for this calculation. However, error still exists in the calculation. (Figure 5.1)

For example, assume \(C_1=1480 \text{ m/s}, C_2=1490 \text{ m/s}, C_3=1500 \text{ m/s}, C_4=1510 \text{ m/s}, C_5=1520 \text{ m/s}\), and the size of each pixel is 1mm × 1mm. So we can get \(L_1=1.3 \text{ mm}, L_2=0.26 \text{ mm}, L_3=1.04 \text{ mm}, L_4=0.52 \text{ mm}, L_5=0.78 \text{ mm}\). Therefore, the exact travelling time from A to B is:

\[
\text{Total Travelling Time} = \text{time}_1 + \text{time}_2 + \text{time}_3 + \text{time}_4 + \text{time}_5
\]

\[
= \frac{L_1}{C_1} + \frac{L_2}{C_2} + \frac{L_3}{C_3} + \frac{L_4}{C_4} + \frac{L_5}{C_5} \left( \frac{\text{meter}}{\text{meter/Second}} \right)
\]

\[
= \frac{0.0013}{1480} + \frac{0.00026}{1490} + \frac{0.00104}{1500} + \frac{0.00052}{1510} + \frac{0.00078}{1520}
\]

\[
= 2.603737 \mu S
\]

(5.3)

Using the approximate algorithm, firstly assume the sound pulse travels through each pixel with the same length (according Bresenham Algorithm, the pulse travels only three pixels from A to B, they are \(C_1, C_2, C_4\)), so the average length which ultrasound pulse travels through each pixel \(\Delta L = \frac{AB}{3} = 1.3 \text{ mm}\), this travelling time is:
\[ \text{travelling time} = \text{time}_a + \text{time}_b + \text{time}_c \]
\[ = \frac{\Delta L}{C_1} + \frac{\Delta L}{C_2} + \frac{\Delta L}{C_4} \left( \frac{\text{meter}}{\text{Second}} \right) \]
\[ = \frac{0.0013}{1480} + \frac{0.0013}{1490} + \frac{0.0013}{1510} \]
\[ = 2.611789 \mu S \] (5.4)

The difference between these two time is:
\[ 2.611789 \mu S - 2.603737 \mu S = 0.008052 \mu S \] (5.5)

This example shows the reason why errors exists in the real Bresenham Algorithm applications. Since the reconstructed image is not only 3 × 3 pixels, the error should be larger than this value.

5.1.2 Sound Speed Reconstruction

Sound Speed Reconstruction of Gelatine

During the sound speed reconstruction of gelatine, the result images are very different by using the artificial sound speed map (which is already discussed in chapter 3) and the given real sound speed map. Compare their average sound speed in the ring region in both map, we found the average sound speed in the artificial sound speed map is 1507.5 m/s whereas it in the real sound speed map is 1484.8 m/s.

Consider the sound speed in these two maps, the sound speed at each pixel in the artificial map is around 20 m/s higher than that of the given real sound speed map. We prefer the former one because the artificial sound speed map is calculated from real signals of the measurement. In order to obtain a more ”real” sound speed map, 20 m/s is added to the sound speed at each pixel of the given real sound speed map, then it is used for a new reconstruction. Afterwards, we can find the reconstructed image is quite similar to that by using the artificial sound speed map.

Sound Speed Reconstruction of Phantom

In the sound speed reconstruction of phantom, it has a more complicated structure and the sound speed in the phantom is not so homogeneous as it of gelatine. Therefore, it’s very difficult to set up a sound speed model like that of gelatine reconstruction.
But we can still test if the real sound speed map is correct or not by calculating of several sending and receiving positions using the measure signal. Figure 5.2

![Image: Test of Phantom’s Sound Speed](image)

**Figure 5.2: Test of Phantom’s Sound Speed**

The procedures are:

Firstly, select a sending and receiving position (for example, sending position 0 and receiving position 50) and find their corresponding pixels in the real sound speed map.

Secondly, load the transmission signal at this sending and receiving position, find out the start point of the signal and calculate the corresponding time of this point. This time is the travelling time from the sending position to the receiving position, the real average sound speed from the sender to the receiver is:

\[
\text{real average sound speed} = \frac{\text{distance between sender and receiver}}{\text{travelling time}} \quad (5.6)
\]

Afterward, use improfile function in the real sound speed map to get all of the pixels and their sound speed on the straight line from sending position 0 to receiving position 50, sum all the sound speed up and divide this sum by the pixel number, here we get the average sound speed along this straight line in the real sound speed map.

By calculating these two values, we got the real sound speed is: 1539.6 $\text{m/s}$ and the sound speed in the map is: 1511.3 $\text{m/s}$, therefore the difference between them is 1539.6 $\text{m/s}$ - 1511.3 $\text{m/s}$ = 28.3 $\text{m/s}$.

In order to get a more accurate value, several more sending and receiving positions were sampled. After the same calculation procedure, we found the
compensation values fluctuating in a large area. So, for a more complicated structure object, it is very difficult to find a value to compensate for the error between the real sound speed and it in the sound speed map and how good the image depends on the accuracy of the sound speed map very much. That is the reason why the reconstructed image’s quality (using phantom as a object ) is worse than that by using phantom as a object.

5.1.3 Absorption Reconstruction

Though we can somehow test the sound speed, the absorption of the object is very difficult to test. All of the absorption reconstruction algorithms origin from theoretic formula, We can only define if the absorption reconstruction works or not only by comparing the images’ quality which are before and after reconstruction.

5.2 Reconstruction Regarding Sensor’s Properties

5.2.1 Sensor’s Properties Measurement

In the sensor’s properties measurement, we used a set of instrument designed by ourselves. Though careful measurements were taken and one experiment was repeated several times for getting a more optimum result, the measured data curve still does not fit the theoretical curve very well.

The reason is that the hydrophone was shifted manually and the measurement points were aimed by eyesight which caused big error. as we know,when human eyes look down into the water from the air, there is a refraction, which causes a error, how big the error is depends the depth of the water because human eyes can not set their eyes 100 percent perpendicular to the bottom.(Figure 5.3)

And how big is the string is also a factor which causes error, the larger the diameter of the string, the bigger the error is.

5.2.2 Sensor’s Distance-dependent Property

During the contrast comparison last chapter, we can see the reconstructed image regarding sensor’s distance dependent property is slightly worse than the image
Figure 5.3: Error Caused by Refraction. Since human can not set their eyesight exactly perpendicular to the bottom, error causes by refraction appears. B is the point which should be aimed and A is the point which human eyes saw, AB is the error.

The reason for that could be:

Firstly, the measured distance curve which is used for reconstruction is not accurate.

Secondly, because of the reconstruction, some noise might be amplified.

5.2.3 Sensor’s angle-dependent Property

We can see, there is a bright ring surround the object in the reconstructed image concerning sensor’s angle property. This phenomenon is caused by a "over compensation". (Figure 5.4)

Each sending position, according to the sensor’s angle property curve, there are two angle areas which need higher amplitude compensation than other angles. For example, sending position A has such two angle areas ($\beta_1$ and $\beta_2$), sending position B also has two angle areas ($\alpha_1$ and $\alpha_2$). These areas compose a public area which has a higher compensation amplitude than other part, therefore a bright ring appears.
Actually, there is no object in the bright ring, so theoretically it should not be some reflect signal in those area. Therefore, there must not be such a higher amplitude than other areas, even if their amplitudes already be amplified.

The most probability is that the A-scan transmission signal has some noise. Of course, the noise exists in any part of the transmission signal, but how large the noise appears in the image depends on the amplification factor. The higher the amplification factor, the higher the noise appearance.

Figure 5.4: Bright Ring Caused By Over Compensation
Bibliography


Chapter 6

Acknowledge

I want to give my appreciation to my advisor, Dr. Rainer Stotzka, for his invaluable guidance, help and encouragement during my master thesis.

I thank my supervisor Dieter Hoepfel for his guidance, encouragement and support during my thesis.

I would like to acknowledge Thomas Deck, Susan Vaziri Elahi and Georg Goebel for their great help during my programming and measurement.

I appreciate Tim Mueller and Klaus Schlote-Holubek for their encouragement and experience.

My appreciation also goes to Simon Kelly for his carefully correction of my thesis.