NEW METHODOLOGIES FOR PROCESSING MEDICAL IMAGES

By

Ahmed Fathy Mosaad Elnokrashy

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY
in
Systems and Biomedical Engineering

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2014

NEW METHODOLOGIES FOR PROCESSING MEDICAL IMAGES

By
Ahmed Fathy Mosaad Elnokrashy

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY
in
Systems and Biomedical Engineering

Under the Supervision of

Prof. Dr. Yasser M. Kadah

………………………………

Professor of Biomedical Engineering
Systems & Biomedical Engineering
Faculty of Engineering, Cairo University

Prof. Dr. Abo-Bakr M. Youssef

………………………………

Professor of Biomedical Engineering
Systems & Biomedical Engineering
Faculty of Engineering, Cairo University

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2014

NEW METHODOLOGIES FOR PROCESSING MEDICAL IMAGES

By
Ahmed Fathy Mosaad Elnokrashy

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY
in
Systems and Biomedical Engineering

Approved by the
Examining Committee

____

Prof. Dr. Abo-Bakr M. Youssef, Thesis Main Advisor,

____

Prof. Dr. Yasser M. Kadah, Thesis Main Advisor

____

Prof. Dr. Nahed H. Solouma, Internal Examiner

____

Prof. Dr. Mohamed I. El-Adawy, External Examiner

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2014

Engineer's Name: Ahmed Fathy Mosaad Elnokrashy
Date of Birth: 10/02/1975
Nationality: Egyptian
E-mail: a_nokrashy@link.net
Phone: 01111969545
Address: 3Elsadat st. Faisal, Giza

Registration Date: 1/10/2010
Awarding Date: 19/01/2014
Degree: Doctor of Philosophy
Department: Biomedical and Systems Engineering Department

Supervisors: Prof. Dr. Abo-Bakr M. Youssef
Prof. Dr. Yasser M. Kadah

Examiners:

Prof. Dr. Abo-Bakr M. Youssef (Thesis main advisor)
Prof. Dr. Yasser M. Kadah (Thesis main advisor)
Prof. Dr. Nahed Hussein Saloma (Internal examiner)
Prof. Dr. Mohamed Ibrahim Al-Adawi (External examiner)

Title of Thesis:

New Methodologies for processing medical images

Key Words:
Ultrasound;3D;4D; Speckle Noise Reduction; Raycast; Volume Render; Surface Render; Surface Shading

Summary:

Four-dimensional (4D) ultrasound imaging extends the real-time capability of ultrasound to visualize a real-time volume that can be manipulated by the sonographer. Among the different visualization methods, surface rendering is a common mode for displaying volumetric datasets such as in obstetrical applications. A challenge in this mode is that surface shading is required to visualize the surface and enhances the surface contrast and this has very demanding computational requirements for 3D surfaces. Due to the low quality of the 2D ultrasound images as a result of the presence of speckle noise, we develop a new real-time methodology for processing of the 2D ultrasound images before being used in the rendering pipeline. Then, the thesis addresses the development of an optimized high-performance rendering pipeline based on four stages for preprocessing, volume rendering, surface shading, and post-processing. The new approach is implemented to render real ultrasound volumes on a 4D commercial ultrasound imaging system to verify and validate its performance and illustrate its practicality. The results demonstrate diagnostic quality of rendered volumes at a computational time cost that is suitable for 4D real-time processing requirements, given its low cost of required hardware; the new pipeline has potential for making 4D imaging systems more affordable while maintaining diagnostic quality and performance.

Acknowledgment

I am grateful to my thesis main advisor, Prof. Dr. Yasser M. Kadah, for setting the broad lines of this work and for challenging my potential to innovate throughout my career. In addition, I am grateful to IBE Company team (Amr Hendy, Ahmed Ali, Tamer Hosny, Moamen Ahmed, Marwan H. Hussein and Alaa Megawer).

Dedication

To my Parents, my Wife, my sons (Ali and Obai), my daughters (Noureen and Dareen), all my Family and my Friends.

Table of Contents

List of Tables

List of Figures

Nomenclature

Glossary

2D: Two Dimensional iv, xv, 1, 2, 4, 6, 15, 16, 17, 19, 20, 28, 29, 34, 35, 38, 39, 45, 46, 47, 50, 60, 69
3D: Three Dimentional... iv, xv, 1, 2, 3, 15, 16, 17, 19, 20, 21, 27, 28, 29, 30, 32, 33, 34, 35, 38, 39, 40, 42, 45, 46, 47, 50, 54, 69
4D: Four Dimensional .... iv, xv, 1, 2, 3, 15, 16, 17, 29, 34, 35, 37, 39, 50, 60, 69, 75
ADC: Analog to Digital Converter 36
Cg: C For Graphics 31, 32, 33
CPU: Central Processing Unit 29, 42, 45
CT : Computed Tomography 4, 5, 7, 15, 16
DVR: Direct Volume Render 20, 21
FPGA: Feild Programmble Gate Array 36
GFLOPS: Giga Floating Point Operations 36
GPU: Graphics Processing Unit 2, 15, 24, 29, 30, 31, 33, 35, 37, 45, 69
MIP: Maximum (Minimum) Intensity Projection 29
MPR: Multi Planar Reformatting 28, 42, 60
MRI : Magnetic Resonance Imaging 4, 5, 15, 16
OpenGL: Open Graphics Library 32, 33, 34, 37
SR: Surface Render 28, 42
US: Ultrasound 53
z-buffer: A Buffer Holds a plan to surface Distance 44

Abstract

Four-dimensional (4D) ultrasound imaging extends the real-time capability of ultrasound to visualize a real-time volume that can be manipulated by the sonographer. Among the different visualization methods, surface rendering is a common mode for displaying volumetric datasets such as in obstetrical applications. A challenge in this mode is that surface shading is required to visualize the surface and enhances the surface contrast and this has very demanding computational requirements for 3D surfaces. Due to the low quality of the 2D ultrasound images as a result of the presence of speckle noise, we develop a new real-time methodology for processing of the 2D ultrasound images before being used in the rendering pipeline. Then, the thesis addresses the development of an optimized high-performance rendering pipeline based on four stages for preprocessing, volume rendering, surface shading, and post-processing. The new approach is implemented to render real ultrasound volumes on a 4D commercial ultrasound imaging system to verify and validate its performance and illustrate its practicality. The results demonstrate diagnostic quality of rendered volumes at a computational time cost that is suitable for 4D real-time processing requirements, given its low cost of required hardware; the new pipeline has potential for making 4D imaging systems more affordable while maintaining diagnostic quality and performance.

Chapter 1 : Introduction

Medical imaging industry has experienced high growth rates in the past years, especially the ultrasound imaging modality. Compared to other imaging modalities, such as X-ray, CT and MRI, ultrasound is safe, affordable and mobile, making ultrasound imaging one of the most popular clinical imaging modalities. Since there is no clinically documented risk of using ultrasound, it has been used routinely as a screening device in all pregnancies to assess fetal life and function, diagnose physical anomalies, and detect multiple pregnancies. Ultrasound imaging has also been widely used to image the body's soft tissue, organs, and blood flow in real time. It is highly cost effective and can eliminate the need for more invasive and expensive procedures. Because of its mobility, ultrasound can be used in operation rooms or in emergency vehicles.

Despite of its many advantages and its popularity, ultrasound images are relatively difficult to interpret. For many years, ultrasound images could only be displayed in a two dimensional (2D) manner. Although internal organs of a patient are three dimensional (3D) in nature, conventional ultrasound machines can only produce real time 2D B-mode images. During an ultrasound examination, a physician must sweep the ultrasound probe against the body to create a series of 2D images of the inner anatomy on the monitor. The physician then constructs these 2D images mentally into 3D to help him/her make a diagnostic decision. This traditional procedure is time consuming, subjective, inconsistent, and difficult to repeat. 3D ultrasound imaging method is an effective way to overcome this limitation. It represents an extremely significant advance in medical imaging and has attracted significant interest from the research and development community to develop.

This thesis is concerned with practical problems in 3D imaging with ultrasound. Common to these problems is the reconstruction and depiction of surfaces from 2D images. In ultrasound imaging problem, the images are basically pulse echo B-mode scans that represent the scattering of ultrasound by tissue in a cross sectional plane through the body. Unlike the CT scan slices, the ultrasound slices collected to form the 3D volume are typically not parallel. The 3D reconstruction problem involves fitting surface models to the collected incomplete surface data. While the parallel nature of CT slices allows conventional raytracing techniques to be applied to render an iso-valued surface corresponding to the skull, tissue surfaces depicted in the ultrasound data are far more challenging since they are incomplete on two accounts. Firstly, the nonparallel nature of slices means that some areas are undersampled relative to others. Secondly, the acoustic interfaces depicted in the B-scan images are contaminated by noise and artifacts. Constructive and destructive interference effects result in noise, known as speckle while shadowing and attenuation artifacts result in signal dropout. Tissue boundaries therefore need to be reconstructed from incomplete acoustic interfaces which have been sampled irregularly.

The primary contributions of this thesis are methods for displaying iso-value surface for nonparallel slices, a new method for reconstructing fetus from ultrasound dataset. Also, a new method for motion compensation of the 4D ultrasound probe is presented. A new proposed rendering pipeline that is suitable for rendering ultrasound dataset and the available GPUs is developed. Finally, the preprocessing is optimized by developing a new speckle filter suitable for the 4D ultrasound and that can be implemented in real time on graphics processing units (GPUs).

1.1. **Organization of Thesis**

The thesis is organized after this Introduction chapter as follows. Chapter 2 describes the system for acquiring the 2D ultrasound data, independently developed in the course of this research. Chapter 3 describes the system for acquiring the 4D ultrasound data, developed in the course of this research. In addition to description of the ultrasound visualization approaches. Chapter 4 describes the system design including the pipeline and all the rendering modes implementation. Chapter 5 describes the 2D image preprocessing in general and the proposed 2D speckle reduction technique used in the pipeline in particular. Finally, Chapter 6 provides the conclusions and suggestions for future work.

1.2. **Motivation**

Ultrasound is a noninvasive, low cost, portable, real time and nonionizing pulse-echo imaging modality, predominantly used as a diagnostic tool in modern medicine. Ultrasound is a valuable way of examining many of the body's internal organs, guiding procedures such as needle biopsies, in which needles are used to extract sample cells from an abnormal area for laboratory testing. It is used for imaging the breasts, guiding biopsy of breast cancer, examining fetus, diagnosing a variety of heart conditions and assessing damage after a heart attack or other illnesses. It not only provides qualitative information but can also provide quantitative information about organ in question. In recent years [1], 3D ultrasound is gaining popularity due to its considerable advantages over 2D ultrasound [2]. It provides a three dimensional view, which allows one to see width, height, and depth of images. In obstetrics, the 3D ultrasound helps visualize the fetal face, including clefts and abnormal facial features. It also helps in diagnosis of congenital abnormalities of a fetus and other birth defects, such as spina bifida or cleft palate. In cardiac examination, it can detect abnormalities not otherwise visible. Moreover, 3D elastography is also in research and development stages and has potential clinical applications, most notably as a means of detecting tumors in soft tissues of prostate and breast. However, fully 3D/4D acquisition systems are not widespread due to economic and technological reasons.

The last decade witnessed the widespread adoption of 3D imaging technologies such as, MRI, CT, and ultrasound imaging. Falling costs have played an instrumental role in making these technologies affordable thus fuelling adoption. Key benefits of 3D imaging technologies driving its adoption among physicians and radiologists include ability to conduct quantitative examinations, make accurate measurements, generate a wider range of viewing planes for gathering in-depth information on images acquired; ability to rotate and manipulate images for better understanding of physiological phenomena; faster diagnosis and patient turnaround times. Economic benefits driving adoption among hospitals include superior utilization of services of radiologists, workflow efficiencies, increased patient throughput, and cost reductions in diagnostic radiology. Progress over the years has expanded the reach of 3D imaging from radiologists to oncologists, referring physicians, cardiologists and surgeons, transforming it into a healthcare enterprise wide tool.

Continued technology developments and the growing need among radiologists and referring physicians to perform volumetric visualization of medical images remains a tidal force driving growth in the 4D medical imaging market. 4D imaging generates dynamic volumetric images that include time and motion with significantly less distortion than 3D. Time resolved 4D Ultrasound are forecast to witness strong growth in the coming years. Cardiology is poised to emerge into a major application area for 4D imaging, given the technology's invaluable benefits in imaging moving structures.

Chapter 2 : 2D Ultrasound imaging

2.1. **Diagnostic Ultrasound**

Ultrasound is the term used to describe an oscillating sound pressure wave with a frequency greater than the upper limit of the human hearing range. Ultrasound is thus not separated from 'normal' (audible) sound based on differences in physical properties, only the fact that humans cannot hear it. For medical applications Ultrasound devices operate with frequencies from 20 kHz up to tens of Mega Hertz.

Although ultrasound has medical applications in surgery and therapy, this thesis is concerned only with diagnostic ultrasound, In particular pulse echo systems. In these systems, images of soft tissues are produced by transmitting ultrasound into the body and detecting echoes produced by re flection at tissue boundaries.
Diagnostic ultrasound has several favorable properties compared to other medical imaging modalities. It is regarded as being relatively safe [3] [4], it involves nonionizing radiation and most examinations are noninvasive and do not distress the patient [4]. It is also safe for the operator; consequently examinations may be realy repeated. Ultrasound does not require the special facilities, which X-ray, CT and MRI require, and portable instruments are available. Ultrasound allows soft tissues, which are difficult to depict by conventional X-ray techniques, to be imaged in detail. Unlike other tomographic techniques, ultrasound offers interactive visualization of underlying anatomy in real time and has the ability to image dynamic structures within the body. Blood flow can be recorded using the Doppler Effect and, in some instances, measured quantitatively. Needles and catheters can also be directed under ultrasonic guidance [5].

2.1.1. Ultrasound Compared to other Medical Imaging Techniques

Compared to other medical imaging techniques (like MRI, CT and X-ray) the advantages of ultrasound are [6]:

- Ultrasound examinations are noninvasive.
- It images muscle and soft tissue very well and is particularly useful for delineating the interfaces between solid and fluid filled spaces.
- It renders "live"' images, where the operator can dynamically select the most useful section for diagnosing and documenting changes, often enabling rapid diagnoses.
- No harmful effects have been reported. It has no known long term side effects and rarely causes any discomfort to the patient.
- Equipment is widely available and comparatively flexible. Small, easily carried scanners are available; examinations can be performed at the bedside.
- Relatively inexpensive, quick and convenient, compared to techniques such as X-rays or MRI scans.

  On the downside:
- The resolution of images is often limited. This is being overcome as time passes, but there are still many situations where X-rays produce a much higher resolution.
- Limited penetration depth (3-25cm depending on frequency).

- Ultrasound also does not pass well through bone, so that the method is of limited use in diagnosing fractures. It is possible to obtain quite good ultrasound scans of the brain, but much greater detail is obtained by an MRI scan.
- Ultrasound is reflected very strongly on passing from tissue to gas, or vice versa. This means that ultrasound cannot be used for examinations of areas of the body containing gas, such as the lung and the digestive system.
- The method is operator dependent. A high level of skill and experience is needed to acquire good quality images and make accurate diagnoses.

Finally Table 1 shows a comparison between the deferent known imaging modalities [7].

Table 1. Comparison of imaging modality

Modality	**Ultrasound**	**X-Ray**	**CT**	**MRI**
What is Imaged	Mechanical Properties	Mean tissue absorption	Tissue absorption	Biochemistry (T1 and T2)
Access	Small windows adequate	2 sides needed	Circumferential Around body	Circumferential Around body
Spatial resolution	Frequency and axially dependent 0.3-3mm	1mm	1mm	1mm
Penetration	Frequency dependent 3-25cm	Excellent	Excellent	Excellent
Safety	Very good	Ionizing radiation	Ionizing radiation	Very good
Speed	100 Frame/Sec	Minutes	$\frac{1}{2}$Minute to Minutes	10 Frame/Sec
Cost	$	$	$$$$	$$$$$$$$
Portability	Excellent	Good	Poor	Poor

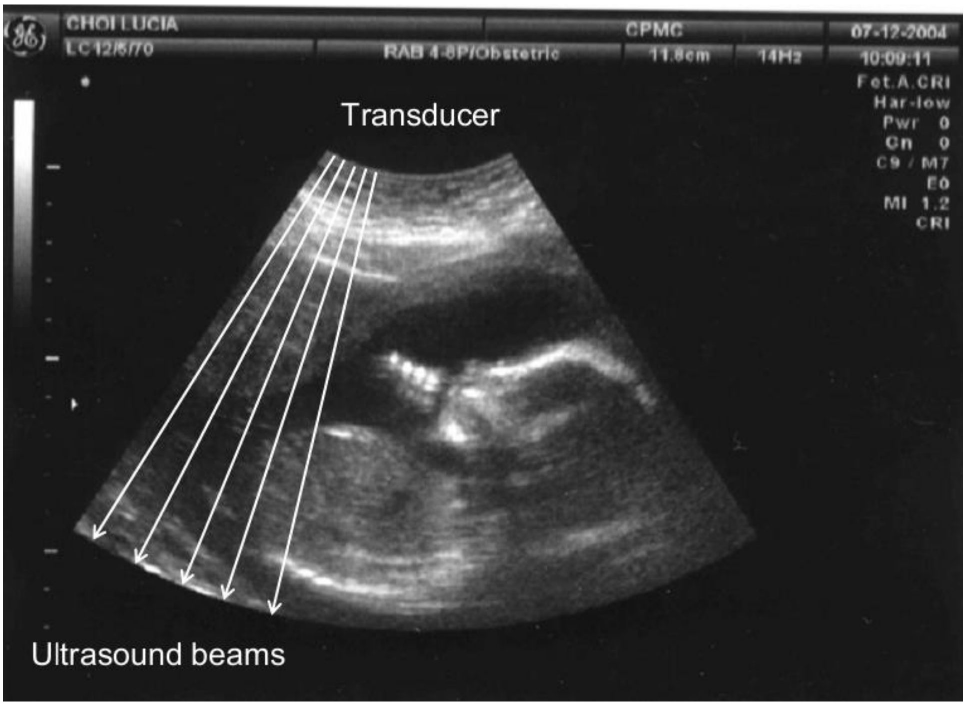


Figure 1. Example of a B-Mode image of a fetus. With arrows indicating the direction of scanlines/beams.

2.2. **Ultrasound Image Formation**

The 2D B-mode image is formed from a large number of B-mode lines, where each line in the image is produced by a pulse–echo sequence. In early B-mode systems, the brightness display of these echoes was generated as follows. As the transducer transmits the pulse, a display spot begins to travel down the screen from a point corresponding to the position of the transducer, in a direction corresponding to the path of the pulse (the ultrasound beam). Echoes from targets near the transducer return first and increase the brightness of the spot. Further echoes, from increasing depths, return at increasing times after transmission as the spot travels down the screen. Hence, the distance down the display at which each echo is displayed is related to its depth below the transducer. The rate at which the display spot travels down the screen determines the scale of the image. A rapidly moving spot produces a magnified image. The pulse–echo sequence, described above, resulted in the display of one line of information on the B-mode image [8]. A complete B-mode image such as that in Figure 1 is made up typically of 128 or more B-mode lines. Let us consider a linear array probe, where the image is formed as illustrated in Figure 2. During the first pulse–echo sequence, an image line is formed, say on the left of the display. The active area of the transducer, and hence the beam, is then moved long the array to the adjacent beam position. Here a new pulse–echo sequence produces a new image line of echoes, with a position on the display corresponding to that of the new beam. The beam is progressively stepped along the array with a new pulse–echo sequence generating a new image line at each position. One complete sweep may take perhaps 1/30th of a second. This would mean that 30 complete images could be formed in 1s, allowing real time display of the B-mode image. That is, the image is displayed with negligible delay as the information is acquired, rather than recorded and then viewed, as with a radiograph or CT scan.

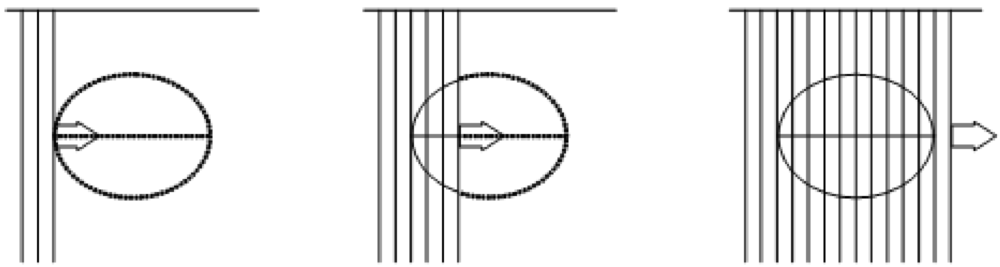


Figure 2. Formation of a 2D B-mode image. The image is built up line by line as the beam is stepped along the transducer array.

2.3. B-mode formats

The B-mode image, just described, was produced by a linear transducer array, i.e. a large number of small transducer elements arranged in a straight line. The ultrasound beams, and hence the B-mode lines, were all perpendicular to the line of transducer elements, and hence parallel to each other Figure 3-a. The resulting rectangular field of view is useful in applications, where there is a need to image superficial areas of the body at the same time as organs at a deeper level.

Other scan formats are often used for other applications. For instance, a curvilinear transducer Figure 3-b gives a wide field of view near the transducer and an even wider field at deeper levels. Curvilinear field of view are widely used in obstetric scanning to allow imaging of more superficial targets, such as the placenta, while giving the greatest coverage at the depth of the baby. The sector field of view Figure 3-d is preferred for imaging of the heart, where access is normally through a narrow acoustic window between the ribs. In the sector format, all the B-mode lines are close together near the transducer and pass through the narrow gap, but diverge after that to give a wide field of view at the depth of the heart. Transducers designed to be used internally, such as intravascular or rectal probes, may use the radial format Figure 3-e as well as sector and linear fields of view. The radial beam distribution is similar to that of beams of light from a lighthouse. This format may be obtained by rotating a single element transducer on the end of a catheter or rigid tube, which can be inserted into the body. Hence, the B-mode lines all radiate out from the center of the field of view.

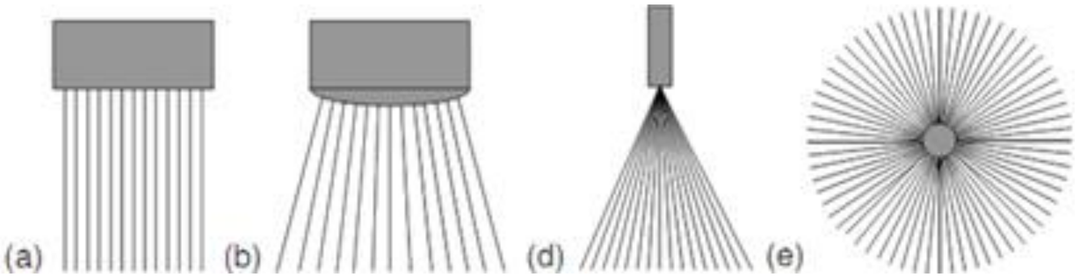


Figure 3. Scan Lines arrangements for the most common B-mode formats. These are (a)linear, (b)curvilinear, (d)sector , (e)radial

2.4. Scan Line Formation

In order to interrogate a particular scan line, an ‘active group’ of adjacent transducer elements, centered on the required scan line, is used. While that scan line is being interrogated, all the other elements in the probe are disconnected and idle. First, a pulse is transmitted; say using the central 32 elements of the group. This pulse travels along the transmit beam, centered on the scan line. As soon as the pulse has been transmitted, a different combination of elements, still centered on the scan line, act together as a receiving transducer, defining the receive beam. The number of elements used for reception is initially less than that used for transmission, but this number is progressively increased as echoes return from deeper and deeper targets until it eventually exceeds that used for transmission. Both the transmit and receive beams can be focused, or otherwise altered, by controlling the signals to or from each of the elements in the active group, as described below. Once all echoes have been received from one scan line, a new active group of elements, centered on the next scan line, is activated. This is achieved by dropping an element from one end of the old group and adding a new one at the other end Figure 4, This advances the center of the active group, and hence the scan line, by the width of one element. The new scan line is then interrogated by a new transmit and receive beam, centered on that line. The process is repeated until all the scan lines across the field of view have been interrogated, when a new sweep across the whole array is commenced.

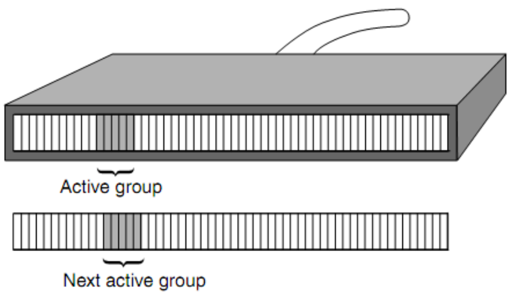


Figure 4 The active group is stepped along the array by dropping an element from one end and adding a new on to the other. In reality, the active group would contain at least 16 elements rather than the five shown here.

2.5. **Scan line Beam Shape**

Scan plane focusing in transmission since the cylindrical lens does nothing to reduce the beam width in the scan plane, an electronic method of focusing must be provided, if good lateral resolution is to be achieved. the transmission focus at the depth for which optimum lateral resolution is desired. This ensures that the transmission beam is as narrow as possible there (the receive beam must also be narrow there, but this is considered next) Usually an arrowhead or other indicator alongside the image indicates the depth at which the transmission focus has been set. Pulses from all the elements in the active group must arrive at the transmission focus simultaneously in order to concentrate the power into a narrow 'focal zone'. However, the distance between an element and the focus, which lies on the beam axis passing through the center of the group, is slightly, but crucially, greater for the outer elements of the group than for more central elements. Pulses from elements further from the center of the active group must, therefore, be transmitted slightly earlier than those nearer the center Figure 5. The ultrasound software builds a look up table into the machine for each possible choice of transmission focus depth available to the operator. These tell the controlling computer the appropriate early start for each element. At points outside the required focal zone, the individual pulses from different elements arrive at different times, producing no more than weak acoustic noise.

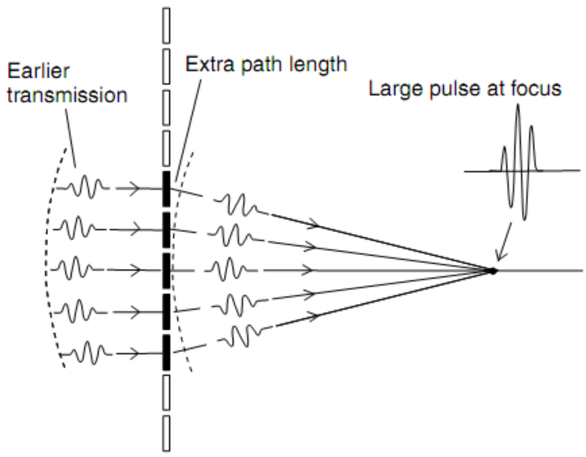


Figure 5. Creating a transmission focus for a line array transducer. In order to form a large-amplitude pulse at the focus, pulses from all elements must arrive there at the same time. This is achieved by transmitting slightly earlier from

4.1. Reception Dynamic Focusing

Focusing in reception means that, for each scan line, the scanner is made particularly sensitive to echoes originating at a specified depth (the receive focus) on the scan line. This also results in the receive beam being narrowed near this focus, further improving lateral resolution. In order for the sensitivity to be high for an echo coming from, or near, the receive focus, the echo signals produced by all transducer elements in the active group must contribute simultaneously to the resultant electronic echo signal. As in the case of transmission focusing, allowance must be made for the fact that the distance between the required focus and a receiving element is greater for elements situated towards the outside of the group than for those near the center. This is done by electronically delaying the electrical echo signals produced by all transducer elements except the outermost, before summing them together Figure 6 The delays are chosen such that the sum of the travel time as a sound wave (from the focus to a particular element) plus the delay imposed on the electrical echo signal is the same for all elements. This means that the imposed delays are greater for elements closer to the center of the active group, for which the sound wave travel times are least. In this way, the echo signals are all aligned in phase at the summing point and a large summed signal is obtained for echoes from the desired receive focal zone, but only a weak summed signal (acoustic noise) results from echoes from elsewhere. In practice, focusing in reception is controlled automatically by the machine; with no receive focus control available for the operator. This is because the ideal depth for the reception focus at any time is the depth of origin of the echoes arriving at the transducer at that time. This is zero immediately after transmission, becoming progressively greater as echoes return from deeper and deeper targets. Since the time needed for a two way trip increases by 13 µs for every additional 1 cm of target depth, the machine automatically advances the receive focus at the rate of 1 cm every 13 µs. The continual advancement of the receive focus to greater and greater depths gives rise to the name 'dynamic focusing in reception'. In fact, high performance machines advance the reception focus in several hundred tiny steps (as many as one for each image pixel down a scan line).

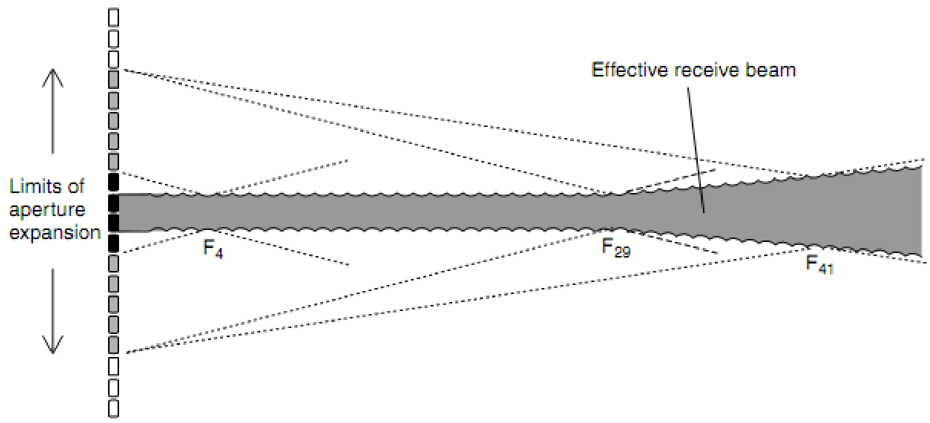


Figure 6. Dynamic focusing and aperture in reception. The machine automatically changes the delays so that the receive focus advances at the rate of 1 cm every 13 µs. At the same time the aperture is expanded, so that the width of the beam at all the foci remain constant (up to the 29th focus in this example). If the aperture stops expanding, the beam widths at deeper foci become progressively greater. The scalloped lines enclosing all the focal zones indicate "the effective receive beam".

2.6. Multiple Zone Focusing

Further improvement in lateral resolution, albeit at the expense of frame rate, is possible by subdividing each scan line into two or more depth zones and interrogating each zone with a separate transmission pulse, focused at its center Figure 7 For example, the operator might select transmission foci at two different depths F1 and F2. These would be indicated by two arrowheads or other focus indicators down the side of the image. One pulse would be transmitted with a focus at F1 and echoes from depths up to about halfway between F1 and F2 would be captured. Then a second pulse would be transmitted with a focus at F2 and echoes from all greater depths would be captured. The greater the number of transmission focal zones, the greater the depth range over which the 'effective transmission beam' is narrow. Unfortunately, the greater the number of focal zones, the longer is spent on each scan line, and so the lower the frame rate. When using multiple transmission focal zones, other transmission parameters such as center frequency, pulse length and shape, aperture, and apodization may all be optimized independently for each of the focal zones. These changes can take account of the fact that pulses sent out to interrogate deeper regions will experience greater attenuation of the high frequencies in their spectra.

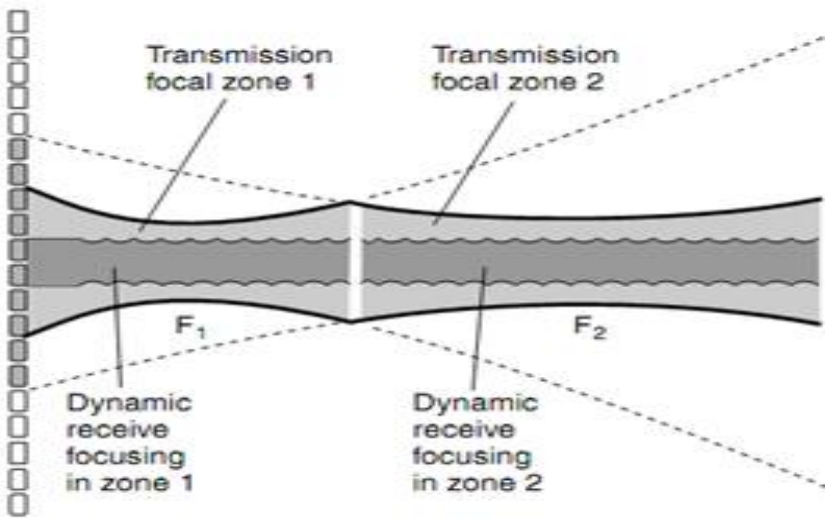


Figure 7. Multiple-zone focusing. The operator has selected two focal zones (F1 and F2). Targets lying between the transducer and a point about half-way between the two foci are interrogated with a transmission pulse focused at F1 . Targets beyond the halfway p point are interrogated by transmitting another pulse along the same scan line, but focused at F2 . The heavy and light scalloped lines indicate the 'effective transmission beam', and the 'effective receive beam',

2.7. **Axial and Lateral Resolution**

The axial and lateral resolutions of a transducer are determined by the emitted pulse duration and the beam width of the transducer (–3 or –6 dB beam width), respectively because whether the echoes from two targets in the axial or in the lateral direction can be separated or resolved is directly related to these parameters. This is graphically illustrated in Figure 8-a, where it can be seen that the echoes from two targets can be clearly resolved if they are far apart. As the targets are moved increasingly closer (as shown in Figure 8-a–b), they become increasingly difficult to resolve. Figure 8-b shows what happens when the two targets coincide. The distance in this figure represents the axial or lateral distance. The beam width at the focal point of a transducer, W_b is linearly proportional to the wavelength.

$$W_b \approx f_{\#}\lambda. \tag{2-1}$$

Here $f_{\#}$ is the f number defined as the ratio of focal distance to aperture dimension. For a circular transducer of diameter 2a and a focal distance of 4a, the transducer has $f_{\#}$ of 2. The depth of focus, D_f i.e., within this region, the intensity of the beam within –1.5dB or –3 dB of the maximal intensity at the focus is also linearly related to the wavelength.

$$D_f \approx f_{\#}^2 \lambda. \tag{2-2}$$

From these relationships, it is clear that an increase in frequency that decreases wavelength improves lateral and axial resolutions by reducing the beam width and the pulse duration if the number of cycles in a pulse is fixed. Unfortunately, these improvements are achieved at a cost of a shorter depth of focus. The axial and lateral

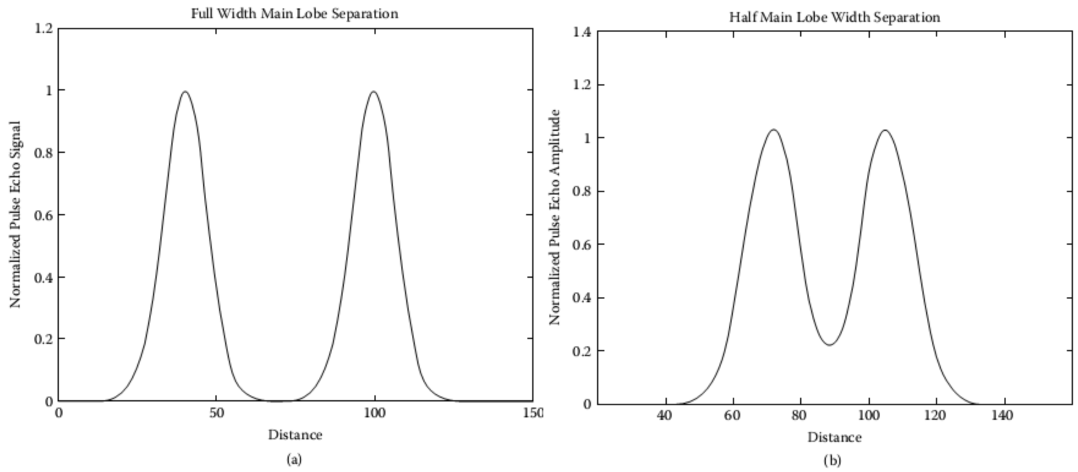


Figure 8. The spatial resolutions of an ultrasonic transducer in the axial and lateral directions are determined by the pulse duration and beam width.

resolution of a transducer can be improved from an increase in the bandwidth by using backing and/or matching and focusing. The spectrum of an ultrasonic pulse varies as it penetrates into tissue because the attenuation of the tissues is frequency dependent. It is known that the center frequency and bandwidth of an ultrasonic pulse decrease as the ultrasound pulse penetrates deeper. In other words, the axial resolution of the beam worsens as the beam penetrates more deeply into the tissue. In commercial scanners, pulse shape and duration are maintained by time–gain–compensation and some form of signal processing [9].

2.8. Attenuation and Penetration Depth

Attenuation of ultrasound results from several phenomena including absorption, scattering, reflection and beam divergence. Despite the complexity of the phenomena, the attenuation property of soft tissue increases approximately linearly with frequency over the frequency range used by diagnostic ultrasound. Thus, a given power and frequency establishes a maximum depth of penetration beyond which echoes are too weak to be detected. External scanning frequencies of 3-7 MHz are common. Higher frequencies, up to 30MHz, are usually used in catheter or trans-esophageal imaging systems where the region of interest is closer to the transducer. For a given transmitted power a tradeoff arises between axial resolution and penetration depth. The transmitted power associated with a pulse is limited by safety considerations. The biological effects of ultrasound are also a function of the carrier frequency, the length of pulses, the pulse repetition frequency, and the maximum pulse amplitude [10].

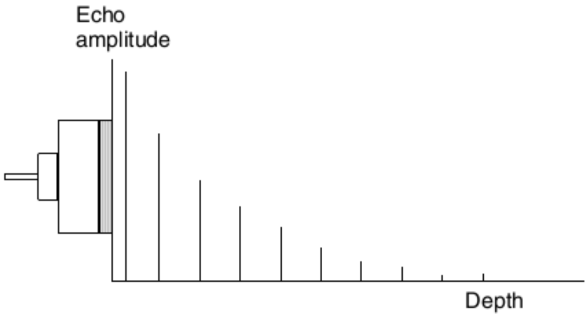


Figure 9. Attenuation results in echoes from interfaces at large depths being smaller than those from similar interfaces near the transducer.

As described, when a transmitted/received ultrasound pulse propagates through tissue, it is attenuated (made smaller). Hence, an echo from an interface at a large depth in tissue is much smaller than that from a similar interface close to the transducer Figure 9. The attenuation coefficient of tissues is measured in dB/cm/MHz. For example, if a particular tissue attenuates a 1 MHz ultrasound pulse by 1.5 dB/cm/MHz, the amplitude of the pulse will be reduced by 15 dB when it reaches an interface 10 cm from the transducer. The echo from this interface will be attenuated by 15 dB also on its journey back to the transducer, so that compared to an echo from a similar interface close to the transducer; the echo will be smaller by 30dB. In this tissue, echoes received from similar interfaces will be smaller by 3 dB for each centimeter of depth [8].

2.9. **Time Gain Compensation**

In a B-mode image, the aim is to relate the display brightness to the strength of the reflection at each interface regardless of its depth. However, as we have just noted, echoes from more distant targets are much weaker than those from closer ones. Hence, it is necessary to compensate for this attenuation by amplifying echoes from deep tissues more than those from superficial tissues. As echoes from deep interfaces take longer to arrive after pulse transmission than those from superficial interfaces, this effect can be achieved by increasing the amplification of echo signals with time Figure 10. The technique is most commonly called time–gain compensation (TGC).

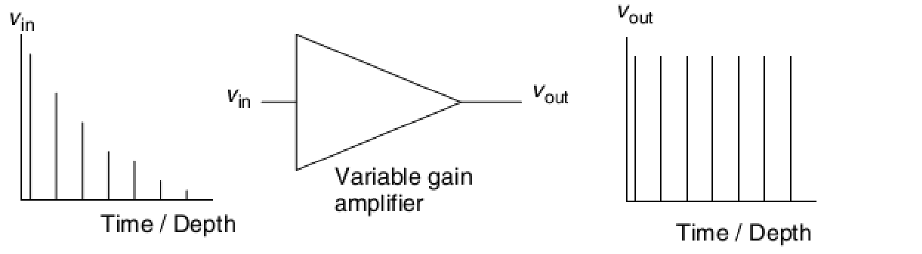


Figure 10. Time gain compensation (TGC)

2.10. **Speckle Noise**

Speckle appearing in any coherent images is due to the coherent interference of waves reflected from many elementary scatterers [11]. This effect causes a pixel-to-pixel variation in intensities, and the variation manifests itself as a granular noise pattern in the images. Speckle noise complicates the image interpretation and image analyses, and reduces the effectiveness of image segmentation and feature classification. Understanding speckle statistics is essential for better information extraction by designing intelligent algorithms for speckle filtering [12].

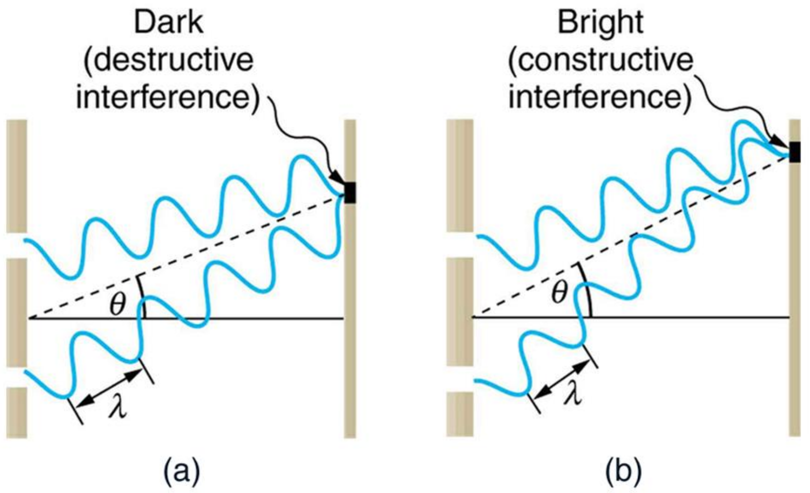


Figure 11. Constructive and Destructive interference

Medical ultrasound is coherent imaging system, the scattering or reflection of acoustic waves is due to the inhomogeneities in the medium's density and/or compressibility, when these backscattered acoustic pulses arrive back to the transducer, they may be in phase or out of phase. The combined received acoustic energy exhibits both constructive and destructive interference Figure 11, giving rise to a granular pattern called speckle.

Chapter 3 : 3D/4D Ultrasound Imaging

3D reconstruction of ultrasound refers to placing the acquired 2D US images into 3D image in their correct relative positions. The majority of ultrasound imaging is 2D, however all anatomy is three-dimensional. 3D ultrasound is the least prevalent amongst other medical imaging modalities, such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET) or conventional B-mode and Doppler 2D ultrasound. Each has its own distinct advantages and is appropriate for particular situations. In comparison to 2D ultrasound, although 3D offers tremendous benefits, there are sufficient limitations which 3D ultrasound can potentially overcome [13].

3.1. 3D/4D ultrasound Challenges

4D ultrasound brings with it the challenge of working with data acquired in Polar coordinate. This problem, related to interpolation and coordinate transformation, adds to that of segmentation posed by the many artifacts present in ultrasound data. In addition, this modality shares the common problems of surface visualization and working under ultrasound physics constraints as mentioned in Chapter 2, which are associated with all 3D/4D medical imaging modalities.

3.1.1. Data Registration

The first challenge in 4D ultrasound is to calculate exactly where the data is in space. 4D scanning is based on mechanical moving probe, thought the scanning of the 2D from is done while the probe is moving, that leads to misaligned frames in the scanning of the probe in the two wobbling direction.

3.1.2. Data Interpolation

It is not possible to display irregularly spaced data on a regular grid, such as a computer screen, without first interpolating this data. In fact the majority of 4D ultrasound systems interpolate the data using a custom made hardware or the power of the new GPUs. The great advance in the GPU 3D texture interpolation simplifies this problem, while stile the Cartesian to polar grid is the major challenge.

3.1.3. Data Segmentation

Segmentation is the process of identifying, within the data, a particular region of interest. This is a prerequisite for displaying surfaces or for accurately measuring organ volume. In this context, segmentation is generally performed by constructing cross sections of an organ, either in a B-scan, or in slices through a voxel array. However, it can also be performed directly in 4D. The design of automatic segmentation algorithms is probably the most challenging problem in 4D ultrasound, and indeed in medical imaging and computer vision in general. This is equally the case for voxel based and scan plane based data. The problem is accentuated with ultrasound because of the many artifacts present in the images, compared with those of other imaging modalities. Automatic approaches are only feasible if strong assumptions are made about the data; in practice this means such algorithms are restricted to the data.

3.1.4. **Surface Visualization**

The visualization of surfaces is particularly challenging in 4D ultrasound. Gradient operators can be used in raycasting for 3D or offline rendering, but it is not suitable for real time volume rendering such as the 4D. We used the image space shading for the surface visualization of the ultrasound dataset. The main challenges here are the distance map image generation; distance map is used in the image space shading. Another challenge is the filtration of the distance map image, due to the speckle nature of the ultrasound the distance map is so noise.

3.1.5. **Clinical Constraints**

The importance of maintaining the integrity of the original data in subsequent processing has already been established as a prime concern in any clinical context. This context also places other constraints on algorithms designed for freehand 3D ultrasound data; namely speed and simplicity. Speed is important for any clinical tool since the clinician has limited time to spend with the patient, and this time should not be wasted waiting for algorithms to complete. In practice, where processing of medical data takes more than a few minutes, it becomes an offline task to be performed after the patient has left. This is the case for much processing of CT and MRI data. Ultrasound, however, is a real time modality where diagnosis is generally made during the examination of the patient. The most significant advantage of this approach is the ability to re-examine the patient appropriately, to verify conclusions reached from the initial data. It would be a step backwards to design an algorithm that forced ultrasound analysis to become an offline process. This is particularly important in freehand 3D ultrasound, where acquiring a good data set is dependent on both scanning technique and a still patient; the latter is not an easy requirement to satisfy for fetal scans. The clinician must be able to review the data immediately after acquisition, in order to check that these requirements were met, and acquire further data if not. The need for simplicity applies both to the underlying algorithms and the tools into which they are built. Clearly, it is good for the clinician to be able to use the tools with relative ease. More importantly, though, it must be clear what the data represents. It must also be possible to compare and contrast this data with similar data from other patients. This task becomes more complicated with increasing dependency of the display algorithms on user parameters.

3.2. **Collecting 3D Ultrasound Data**

The beam formation and 2D scanning techniques been designed to acquire echo information from a single cross section through the target tissues. The echo information would then be processed and displayed as a real time 2D, B-mode. The same beam formation and scanning techniques can be extended to acquire echo information from a 3D volume of tissue. The resulting 3D volume data set can then be processed to create a number of alternative modes of display, as will be described in the following Chapters. Repetition of the 3D acquisition and display at a few hertz (refresh rates up to 20 Hz are possible) results in a moving 3D display, referred to as 4D [14].

There are two commonly used approaches to the design of 3D/4D transducers. The design illustrated in Figure 12 incorporates an array transducer, in this case a curvilinear array, mounted on a swivel inside an enclosed bath of acoustic coupling fluid (linear array transducers are also used). The transmit pulse and returning echoes pass through a thin acoustic window to acquire a 2D section from the adjacent tissues. A stepper motor, coupled to the swivel by pulleys and a drive belt, rotates the transducer array so that the 2D scan plane is swept at about 90° to the imaged plane, to interrogate and acquire echoes from a conical volume of tissue. The orientation of each scan plane is known by the amount of stepper motor number of steps movement. A home sensor is used for self-calibration of the motor movement. A sequence of closely spaced, adjacent 2D sections is acquired, which constitutes a 3D volume data set. By repeating the sweeping motion of the array from side to side, a 4D image can be created. Due to the mechanical nature of the transducer movement, 4D acquisition rates are limited to a few volumes per second.

The spatial resolution of the acquired data is subject to the same limitations as described earlier for linear array transducers. That is, the lateral resolution in the original scan plane is much better than that in the elevation plane, due to the lack of electronic focusing in the elevation direction. Hence, the units of volume, the voxels, are not square and B-scan images, created in the elevation plane from the 3D data, will have reduced lateral resolution.

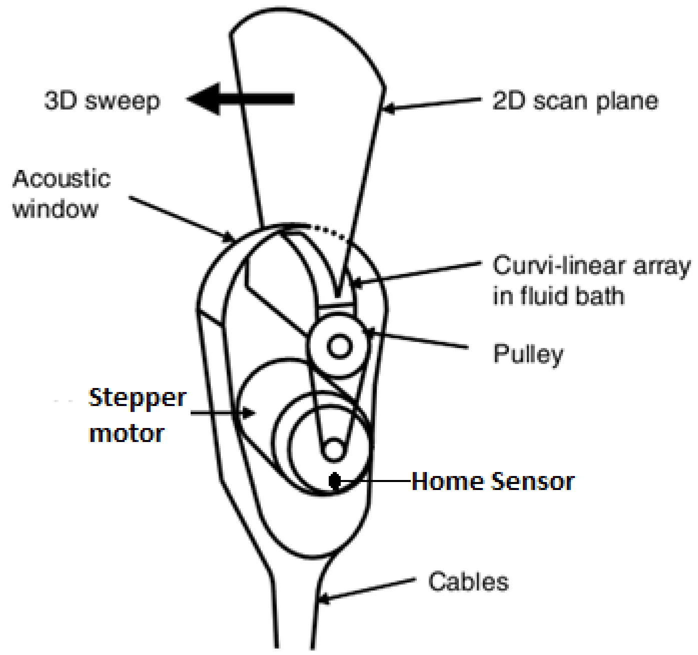


Figure 12. The mechanical 3D transducer contains a stepped array mounted on a swivel in an enclosed fluid bath beneath a thin acoustic window. The image plane is swept from side to side by a motor coupled via pulleys and drive belt. 3D volumes can be acquired at up to 10 Hz.

The alternative, commonly used 3D transducer is the 2D matrix array Figure 13. This transducer contains several thousand square elements arranged in a two dimensional matrix. Using beam-steering techniques as described earlier, the matrix array can be used to create a 2D sector scan in a single plane. By applying beam-steering techniques also in the orthogonal direction (elevation), the 2D sector can be swept sideways to describe a pyramidal volume. As the matrix of elements is square, it is possible to achieve equivalent resolution in the lateral (scan plane) and elevation directions by applying the same degree of electronic focusing and aperture control to both planes. However, the spatial resolution is subject to the limitations of phased array transducers described above and the relatively small number of elements (approximately 50) along each side of the array. As the matrix array has no moving parts, the 3D acquisition rate is limited only by speed-of-sound. If the imaged depth and angle of sweep are restricted, volume rates of up to about 20 Hz are possible. The stored volume data can be interrogated and displayed as a sequence of adjacent 2D sector scans in any B-mode plane or as a C-scan, where the imaged section is parallel to the transducer face. Alternative modes of 3D display, including surface rendering, will be described in detail in the following sections.

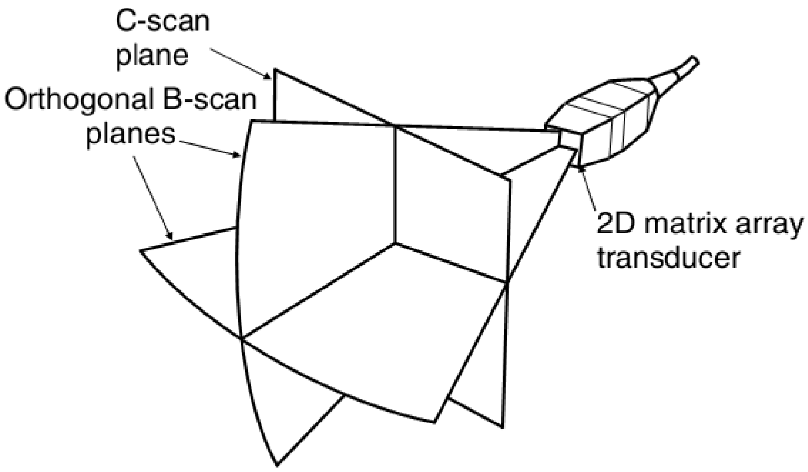


Figure 13. The 2D matrix-array transducer has a square array of elements, e.g. 50 × 50. It uses beam-steering techniques to sweep a 2D sector scan through a 3D volume or interrogate orthogonal B-scan planes. Volume data can be displayed as a series of B-scan planes or as a C-scan. 3D display modes are also widely used.

3.3. **3D Ultrasound Visualization Approaches**

3D ultrasound visualization generally uses three basic approaches; slice projection, surface fitting and volume rendering that are discussed in the following sections [1].

3.3.1. **Slice Projection**

Slicing methods are one of several interactive techniques available for physician review of patient data. Extraction of a planar image of arbitrary orientation at a particular location in a 3DUS data set utilizes standard coordinate transformations and rotations. This is the most computationally straightforward approach to review data throughout the volume, requiring minimal processing with isotropic data. Interactive display of planar slices offers the physician retrospective evaluation of anatomy, particularly viewing of arbitrary planes perpendicular to the primary exam axis and other orientations not possible during data acquisition. Multiple slices displayed simultaneously can be particularly valuable to assist in understanding patient anatomy. Typically, slicing methods are fully interactive, replicating the scanner operational feel. Multiplan displays are often combined with rendered images to assist in localization of slice plane position (Figure 16 and Figure 38). Currently, these review approaches are the most common method used to review 3DUS data.

3.3.2. **Surface Rendering**

Surface fitting provides a good means for visualizing spatial relationships for the entire volume in a readily comprehended manner. Lighting or surface shading enhances the visualization of the volume. Surface extraction algorithms typically the most challenge, and allot of work done in this area. For the offline application surface fitting is the major algorithm, that done by fitting the surface to a planner surface primitive e.g. polygons. While for real time applications, the image space and P-splatting are the major algorithms. Image space shading showed a good performance, but surface detection and z-buffer generation affect greatly the surface quality.

3.3.3. **Volume Rendering**

Volume rendering methods map individual voxels (V) directly onto the screen without using geometric primitives, but require that the entire data set be sampled each time an image is rendered or rerendered. In some situations, a low resolution pass is sometimes used to create low quality images quickly for optimization of viewing position or parameter setting with a high resolution image produced immediately after values have stopped changing. The most often used volume visualization algorithm for the production of high quality images is ray casting.

3.4. **3D Image Reconstruction**

Image reconstruction refers to the process of generating a 3D representation of the anatomy by first placing the acquired 2D images in their correct relative positions and orientations in the 3D image volume, and then using their pixel values to determine the voxel values in the 3D image. Two distinct reconstruction methods have been used: feature-based and voxel-based reconstruction [15].

3.4.1. Feature-Based Reconstruction

In this method, desired features or surfaces of anatomical structures are first determined and then reconstructed into a 3D image. For example, in echocardiographic or obstetric imaging, the ventricles or fetal structures may be outlined (either manually or automatically by computer) in the 2D images, and only the resulting boundary surfaces presented to the viewer in 3D. The surfaces of different structures may be assigned different colors and shaded, and some structures may be eliminated to enhance the visibility of others. This approach has been used extensively in 3D echocardiography to identify the surfaces of ventricles. Because this approach reduces the 3D image content to the surfaces of a few anatomical structures, the contrast of these structures can be optimized. Also, by using multiple images, their dynamic behavior can be easily appreciated. In addition, it also enables the 3D image to be manipulated efficiently using inexpensive computer display hardware (e.g. to produce 'fly-through' views). However, this approach also has major disadvantages. Because it represents anatomical structures by simple boundary surfaces, important image information, such as subtle anatomical features and tissue texture, is lost at the initial stage of the reconstruction process. Moreover, the artificial contrast between structures may misrepresent subtle features of the anatomy or pathology. Furthermore, the boundary identification step time consuming, and subject to errors if done automatically by a computer segmentation algorithm.

3.4.2. Voxel-Based Reconstruction

The more popular approach to 3D ultrasound image reconstruction is based on using a set of acquired 2D images to build a voxel-based image, i.e. a regular Cartesian grid of volume elements in three dimensions. Reconstruction is accomplished in two steps: first, the acquired images are embedded in the image volume, by placing each image pixel at its correct 3D coordinates (x, y, z), based on the 2D coordinates (x,y) of that pixel in its 2D image, and the position and orientation of that image with respect to the 3D coordinate axes. Then, for each 3D image point, the voxel value (color or grey scale image intensity) is calculated by interpolation, as the weighted average of the pixel values of its nearest neighbors among the embedded 2D image pixels. For mechanically scanned images, the interpolation weights may be precomputed and placed in a look-up table, allowing the 3D image to be rapidly reconstructed [16]. This voxel-based reconstruction approach preserves all the information originally present in the acquired 2D images. Thus, by taking suitable cross sections of the 3D image volume, the original 2D images can be recovered. Moreover, new views not found in the original set of images can be generated.

3.5. Medical Volume Visualization

Volume data can be visualized by directly projecting the data to the screen (direct volume rendering, DVR) or by generating an intermediate representation, which is subsequently rendered (indirect volume rendering) [17].

3.5.1. **Indirect Volume Visualization**

An intermediate representation is required by the rendering engine. Volumetric data are composed of large number of voxels datasets. A verity of approaches to extract information from this large set of information attempts to do so by focusing on a subset of that information. Such methods include plane-oriented visualization, surface-oriented visualization, Fourier Volume Rendering and Monte Carlo Volume Rendering [17]. We are concerned here with the most widely used method in ultrasound, namely; the surface-oriented visualization see section 3.3.2 and plane-oriented visualization see section 3.3.1.

3.5.2. **Direct Volume Rendering**

This method involves the direct visualization of volume data without any intermediate conversion of the volumetric dataset to a surface representation. The idea of Direct Volume Rendering (DVR) is to get a 3D representation of the volume data directly. The data is considered to represent a semitransparent light emitting medium. The approaches in DVR are based on the laws of physics (emission, absorption, scattering). The volume data is used as a whole (we can look inside the volume and see all interior structures). In DVR either backward or forward methods can be used. Backward methods use image space/image order algorithms. They are performed pixel by pixel. An example is raycasting which will be discussed in detail below.

All direct volume rendering algorithms can be classified into two groups; image space approaches, sometimes also called backward mapping approaches, and object space approaches, also called forward mapping approaches.

3.5.2.1. Volume Raycast

The classic direct volume rendering algorithm is the image space oriented ray casting, which is a variation of ray tracing. In both cases, rays are cast from the eye or viewpoint through the image-plane and through the dataset Figure 14. In the case of ray tracing, contributions are typically computed at the intersections of the rays with the objects of the scene. Depending on the material properties of the objects, the rays are refracted or reflected. If both effects take place, secondary rays are cast from the intersection point, while the primary rays proceed in the current or new directions. In contrast, standard ray casting does not traverse a scene containing various objects, but a volumetric dataset. Consequently, it does not compute intersections with objects, since all voxels of the dataset may be contributing. Instead, the ray samples the volumetric dataset along its path. Depending on the specified properties (specified by the transfer functions), the samples may be contributing (having opacity values larger than zero) or not contributing (having opacity values of zero). The positions of the samples on the ray within the data volume depend on the direction of the ray and the sampling rate, which governs the distance between neighboring samples on that ray. Since in most cases these positions will not be located directly at voxel positions, we need to compute the sample values by an interpolation method. The typical used method is trilinear interpolation. Next to the difference between computing intersections and sampling, the other big difference between ray casting and ray tracing is the lack of secondary rays, since here the rays are neither refracted nor reflected at the sampling points. Similar to ray tracing, ray casting expects at least one ray to be cast per pixel of the image plane (hence, it is classified as an image-space approach). However, the sampling theorem tells us that this is not enough to avoid sampling artifacts. Therefore, 2×2 or more rays per pixel should be cast. Since this increases the computational load fourfold, this is rarely done. Fortunately, only very few features are small enough to witness the sampling artifacts. Alternatively, a coarse sampling rate is used for a rapid exploration mode, while the full quality rate is used to refine the current image into the final image, once the viewing parameters are set by the rapid exploration mode. Note that also the distance of the sampling positions on the cast rays are subject to sampling issues, as the name already suggests. Hence the selected sampling rate should reflect the smallest voxel spacing with at least two samples per volume cell.

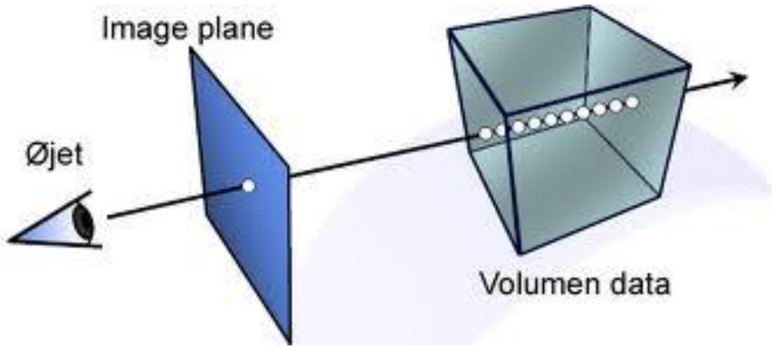


Figure 14. Volume raycasting

3.5.2.1.1. Rendering Quality

The main factor affect the rendering quality is the number of samples within the ray. Small features might be missed if the number of samples is insufficient or no samples represent them. The adaptive subsampling of the ray is the main method used to speed up the rendering time while does not drop any details.

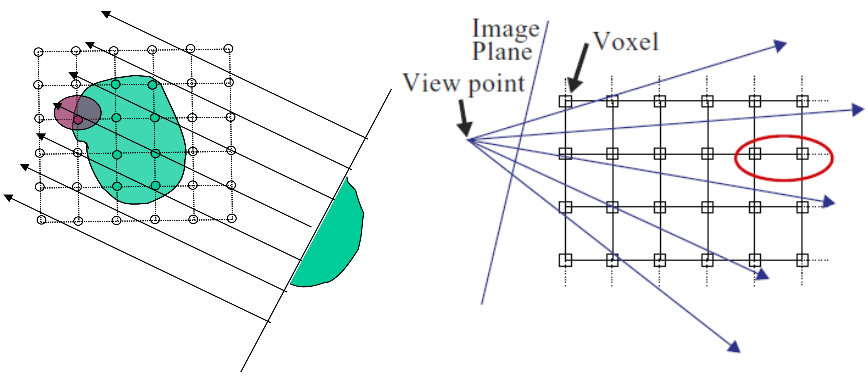


Figure 15. Raycast projection, (left) parallel projection , (right) perspictive projection the red ellipse indicates two voxels on the grid that are completely missed

For the parallel or orthographic projection, in which all rays remain parallel along their way through the volume dataset. Hence, the density of sampling points remains constant, if the sampling rate is not changed. This, however, is different with perspective projections, where the rays diverge from each other on their way through the volume dataset. This ray divergence reduces the sampling rate along the rays and causes aliasing see Figure 15.

3.5.2.1.2. Rendering Speed

Historically, most acceleration methods tried to reduce computational time by reducing the number of samples [18] [19]. The most known methods are Early ray termination and Empty space skipping.

3.5.2.1.3. Early Ray Termination

When tracing rays through a volume , many rays quickly accumulate full opacity. This means that features in the data set in the back are occluded and need not be considered for rendering the image. Thus, the ray can be terminated.

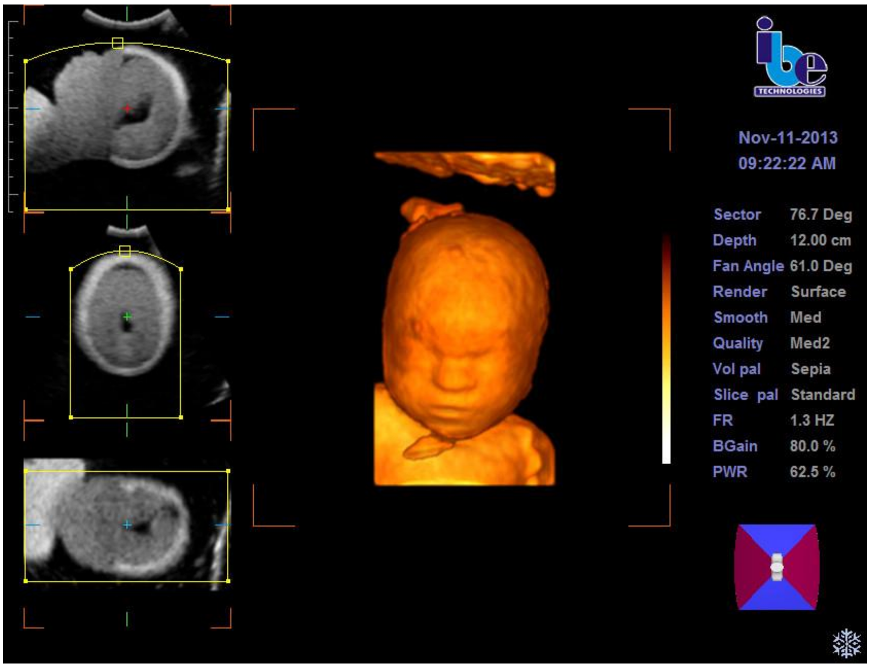


Figure 16. Fetal face with early ray termination and Empty space skipping

3.5.2.1.4. Empty space skipping

Volumetric data sets contain many features that are not required for that final rendering. Typically, they are removed by setting zero alpha values in the transfer function. In order to not waste time on features that have been removed by the transfer function, empty space skipping can be employed to reduce memory access and save computation power [16].

Empty space skipping is a problem dependent approach. For the fetal face, it is an ideal case where the face is always preceded by black area (amniotic fluid) as in Figure 16.

3.5.2.2. Shear Warp

Shear-warp is such an algorithm. It is considered to be the fastest software based volume rendering algorithm. It is based on a factorization of the viewing transformation into a shear and a warp transformation. The shear transformation has the property that all viewing rays are parallel to the principal viewing axis in sheared object space. This allows volume and image to be traversed simultaneously Figure 17. Compositing is performed into an intermediate image. A two dimensional warp transformation is then applied to the intermediate image, producing the final image [20]. This basic mechanism is illustrated in Figure 18.

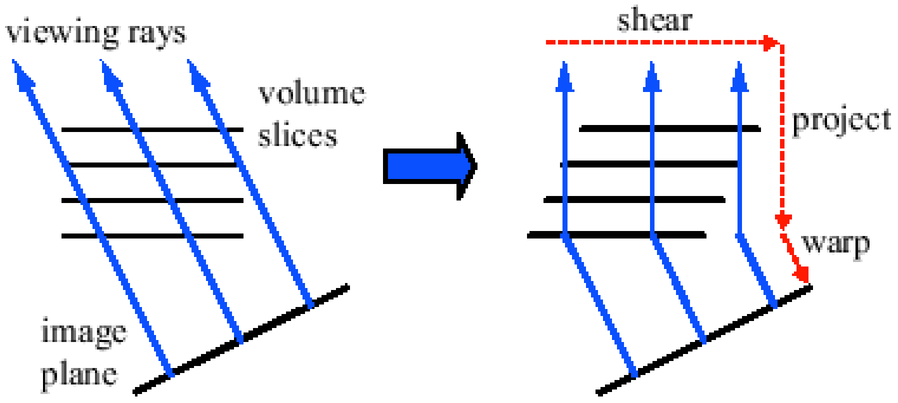


Figure 17. Illustration of the shear-warp mechanism. The volume slices are sheared so that all viewing rays are parallel to the major viewing axis. After the projection process has been performed, the distorted intermediate image is warped into the final image

The problem of shear-warp is the low image quality caused by using only bilinear interpolation for reconstruction, a varying sample rate that is dependent on the viewing direction, and the use of preclassification. The image quality is still inferior when compared to other methods, such as raycasting. The main drawback of the Shear-Warp algorithm is the need to rearrange the slices before sending to the GPU according to the viewing direction.

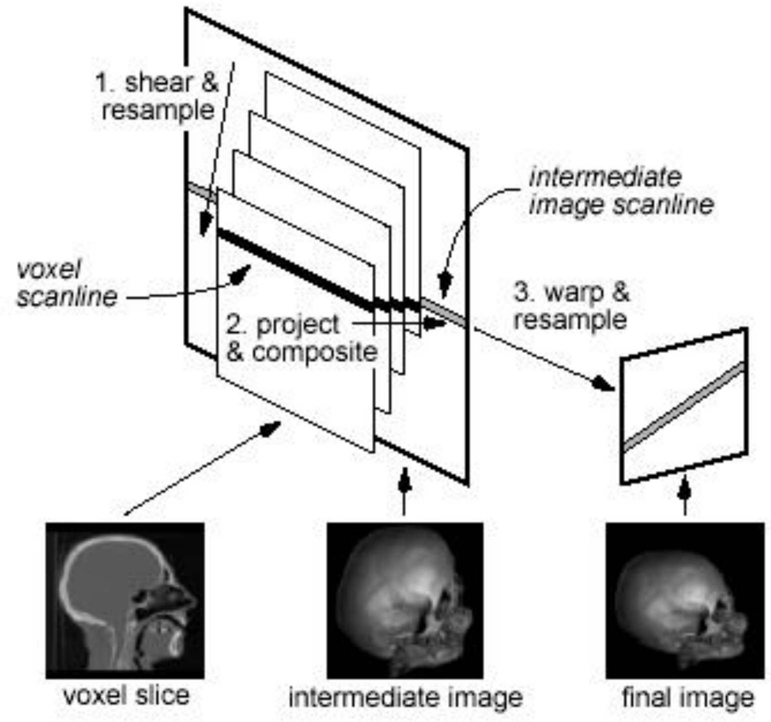


Figure 18. shear-warp conceptual algorithm.

3.5.2.3. Splatting

The most important algorithm of the object order family is the splatting algorithm [21] [22]. One way of achieving volume rendering is to try to reconstruct the image from the object space to the image space, by computing the contribution of every element in the dataset to the image. Each voxel's contribution to the image is computed and added to the other contributions. The algorithm works by virtually “throwing” the voxels onto the image plane. In this process every voxel in the object space leaves a footprint in the image space that will represent the object. The computation is processed by virtually slice by slice, and by accumulating the result in the image plane.

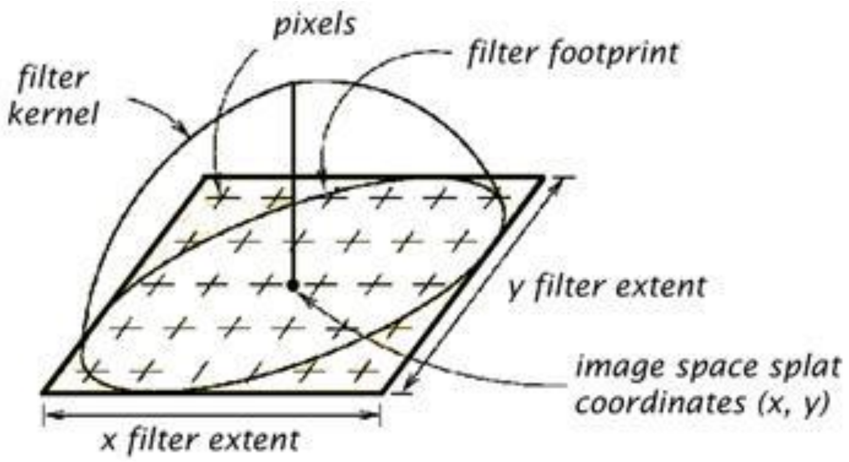


Figure 19. The splatting process reconstruction and resampling with the 2D filter kernel

The first step is to determine in what order to traverse the volume. The closest face (and corner) to the image plane is determined. Then the closest voxels are splatted first. Each voxel is given a color and opacity according to the look up tables set by the user. These values are modified according to the gradient.

Next, the voxel is projected into the image space. To compute the contribution for a particular voxel, a reconstruction kernel is used. For an orthographic projection a common kernel is a round Gaussian Figure 19. The projection of this kernel into the image space (called its footprint) is computed. The size is adjusted according to the relative sizes of the volume and the image plane so that the volume can fill the image. Then the center of the kernel is placed at the center of the voxel's projection in the image plane (note that this does not necessarily correspond to a pixel center). Then the resultant shade and opacity of a pixel is determined by the sum of all the voxel contributions for that pixel, weighted by the kernel.

One of the advantages of splatting is the voxel orientation of the approach. This way, it is less sensitive to changes of the image plane size, which are typical factors for increased costs in the ray casting approach. Instead, the filter kernel and hence the footprint size is adapted to the higher (or lower) image-plane resolution. Larger filter sizes can also be used to provide a lower fidelity representation of volume regions with high data coherence. This advantage, however, comes with several drawbacks, such as pre-classification volume rendering, illumination artifacts due to axis-aligned sampling, and potentially excessive blurring. In general, the size of the filter kernels (splats) should be chosen so that they slightly overlap to include all voxels and their respective volume cells [17].

3.5.2.4. Texture Mapping

Modern graphics hardware includes facility to draw a textured polygon. The texture is an image with color and alpha components is shown in Figure 20 so several overlapping polygons can be composited. Draw from back-to-front a set of rectangles, first rectangle drawn as an area of colored pixels, with associated opacity, as determined by transfer function and interpolation then merged with background in a compositing operation (supported by hardware).

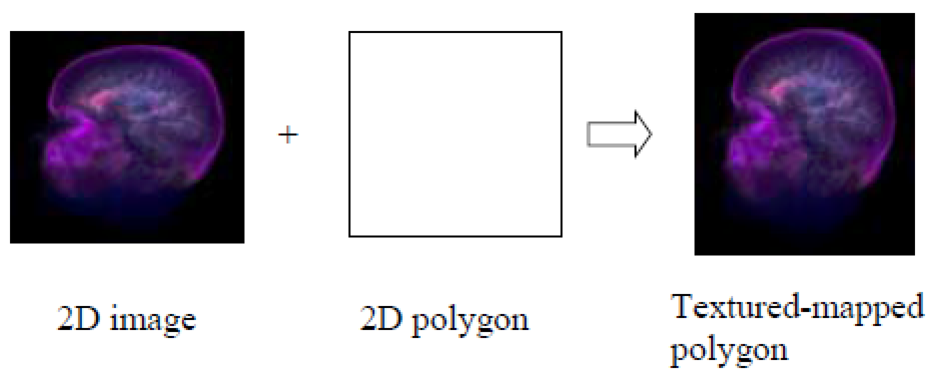


Figure 20. 2D Image mapped to 2D polygon forming Textured-mapped polygon

Successive rectangles drawn on top for a given viewing direction, we would need to select slices perpendicular to this direction. This requires interpolation to get the values on the slices, 3D texture hardware can do this fast enough. For a given viewing direction, we would need to select slices perpendicular to this direction Figure 21; this requires interpolation to get the values on the slices.

The main disadvantages of the texture mapping volume rendering, Need to compute offline the slicing planes for every view angle, which is time-consuming process, Low quality shading (not correct illumination).

The main advantage is the rendering speed is fast. While for real time application, it is not applicable, because the need to compute the slicing planes for every viewing angle, even though using the 3D texture mapping.

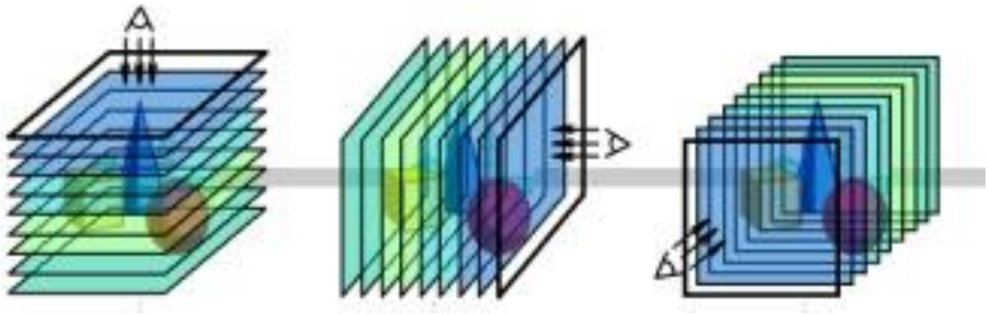


Figure 21. Slices parallel to coordinate planes

3.6. 3D Ultrasound Image Display

Over the past decade, many algorithms and software utilities have been developed to manipulate 3D images interactively using a variety of visualization tools. Although the quality of the 3D image depends critically on the method of image acquisition and the fidelity of the 3D image reconstruction, the viewing technique used to display the 3D image often plays a dominant role in determining the information transmitted to the operator. There are many approaches for displaying 3D images, which differ in the details of the display implementation. Many of these techniques have been investigated for use with 3D ultrasound images, and these can be divided into three broad classes: surface rendering (SR), multiplanar reformatting (MPR) and volume rendering (VR) [15].

3.6.1. Surface Rendering (SR)

A common 3D display technique used in medical imaging is based on visualizing the surfaces of organs. Because the 3D image must first be reduced to a description of surfaces, image classification and segmentation steps precede the rendering. In the first step, each voxel (or voxel group) in the 3D image must be either manually or automatically classified as to which structure it (or they) belongs. Organ boundaries are then identified using manual contouring [23], or computer based segmentation techniques [24]. Once the organs have been classified and segmented, the boundary surfaces are each represented by a mesh and then texture mapped with a color and texture which appropriately represents the corresponding anatomical structure.

Because early 3D ultrasound imaging techniques did not have the benefit of the high computational speed now available in inexpensive computers, only simple wireframe rendering techniques were used at first. These early approaches were used for manipulating the fetus volume [25]. More recent approaches use more complex surface representations, which are texture mapped, shaded and illuminated, so that both the surface topography and its 3D geometry are more easily comprehended. These approaches typically allow user controlled motion for viewing the anatomy from various perspectives. This approach has been used successfully by many investigators in the rendering of 3D echocardiographic and obstetric images [1].

3.6.2. **Multi Planar Reformatting (MPR)**

Rather than displaying only the surfaces of structures, as in the SR technique, the MPR technique provides the viewer with planar cross sectional images extracted from the 3D image. Thus, this approach requires that either a voxel based 3D image be first reconstructed, or that an algorithm be used that can extract an arbitrarily oriented plane from the original set of acquired 2D images [26]. Because the extracted plane will generally not coincide with one of these original image planes, interpolation is required to provide an image similar in appearance to an acquired 2D image. Thus, the extracted image resembles a conventional 2D ultrasound image, which is already familiar to the operator; the operator can then easily position and orient it to the optimal image plane for the examination. This technique is easy to learn and is therefore preferred by radiologists. Two MPR techniques have been used to view 3D ultrasound images: orthogonal planes and cube view.

3.6.3. **Orthogonal Planes.**

In this approach, computer–user interface tools provide three perpendicular planes which are displayed on the screen simultaneously, with graphical cues indicating their relative orientations. The operator can then select any single or multiple planes and move them within the 3D image volume to provide a cross sectional view at any desired location or orientation, including oblique views [1].

3.6.4. **Cube View**

In this approach, illustrated by Figure 38, the 3D image is presented as a polyhedron which represents the boundaries of the reconstructed volume. Each face of the multifaceted polyhedron is rendered with the appropriate ultrasound image for that plane, using a texture mapping technique [26]. With the computer–user interface tools provided, the operator can select any face, and move it in or out, parallel to the original face or reorient it obliquely, at any angle to the original face, while the appropriate ultrasound data are continuously texture mapped in real time onto the new face. In addition, the entire polyhedron may be arbitrarily rotated in any direction to obtain the desired orientation of the 3D image. In this way, the operator always has 3D image based cues relating the plane being manipulated to the anatomy being viewed [15].

3.6.5. **Volume rendering (VR)**

The volume rendering technique presents a display of the entire 3D image after it has been projected onto a 2D plane, with the image projection typically being accomplished via raycasting techniques [19]. The 3D image voxels that intersect each ray are weighted and summed to achieve the desired result in the rendered image. Although many VR algorithms have been developed, 3D ultrasound aging currently makes use of two main approaches: maximum (minimum) intensity projection and translucency rendering.

3.6.5.1. Maximum (minimum) intensity projection (MIP)

A common approach is to display only the maximum (minimum) voxel intensity along each ray. This approach is easy to implement and fast to compute, allowing real time manipulation of the MIP image on inexpensive computers. Excellent results are typically achieved when the image information is sparse, such as it is in Bone images Figure 38.

3.7. 4D Probe

The 4D probe used is from Prosonic Company. The 4D probe Is a mechanical sector probe with a convex 2D scanning. Digison include a control board to control the 4D probe stepper motor through SPI protocol. 4D interface with the 4D control board through the 2D Application. Full 4D probe specs included in Appendix I

3.8. Computing Platform

The used GPU for the current implementation is NVIDIA GT 630. It is one of the med scale GPUs by NVIDIA. This card is one of The Current generation of GPUs (2003 and on) includes NVIDIA's GeForce GT family. These GPUs provide both vertex level and pixel level programmability. This level of programmability opens up the possibility of offloading complex vertex transformation and pixel shading operations from the CPU to the GPU. Various OpenGL extensions expose the vertex level and pixel level programmability of these GPUs. This is the generation of GPUs where Cg gets really interesting. This GPU generation is designed for the PCI Express 2.0 bus architecture offering the highest data transfer speeds for the most bandwidth hungry games and 3D applications, while maintaining backwards compatibility with existing PCI Express motherboards for the broadest support.

3.8.1. The Graphics Hardware Pipeline

A pipeline is a sequence of stages operating in parallel and in a fixed order. Each stage receives its input from the prior stage and sends its output to the subsequent stage. Like an assembly line where dozens of automobiles are manufactured at the same time, with each automobile at a different stage of the line, a conventional graphics hardware pipeline processes a multitude of vertices, geometric primitives, and fragments in a pipelined fashion.

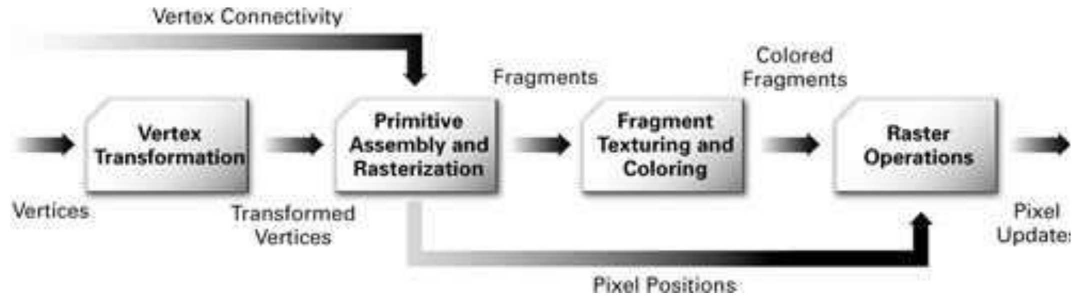


Figure 22. The Graphics hardware Pipeline

Figure 22 shows the graphics hardware pipeline used by today's GPUs. The 3D application sends the GPU a sequence of vertices batched into geometric primitives: typically polygons, lines, and points. There are many ways to specify geometric primitives.

Every vertex has a position but also usually has several other attributes such as a color, a secondary (or *specular*) color, one or multiple texture coordinate sets, and a normal vector. The normal vector indicates what direction the surface faces at the vertex, and is typically used in lighting calculations.

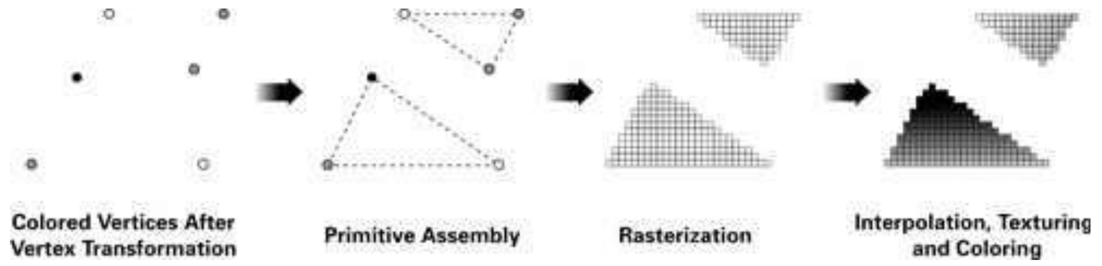


Figure 23. Visualizing the graphics pipeline

Figure 23 depicts the stages of the graphics pipeline. In the figure, two triangles are rasterized. The process starts with the transformation and coloring of vertices. Next, the primitive assembly step creates triangles from the vertices, as the dotted lines indicate. After this, the rasterizer "fills in" the triangles with fragments. Finally, the register values from the vertices are interpolated and used for texturing and coloring. Notice that many fragments are generated from just a few vertices.

3.8.2. Interpolation, Texturing, and Coloring

Once a primitive is rasterized into a collection of zero or more fragments, the interpolation, texturing, and coloring stage interpolates the fragment parameters as necessary, performs a sequence of texturing and math operations, and determines a final color for each fragment. In addition to determining the fragment's final color, this stage may also determine a new depth or may even discard the fragment to avoid updating the frame buffer's corresponding pixel. Allowing for the possibility that the stage may discard a fragment, this stage emits one or zero colored fragments for every input fragment it receives.

3.8.3. **The Programmable Graphics Pipeline**

The dominant trend in graphics hardware design today is the effort to expose more programmability within the GPU. Figure 24 The programmable graphics pipeline shows the vertex processing and fragment processing stages in the pipeline of a programmable GPU.

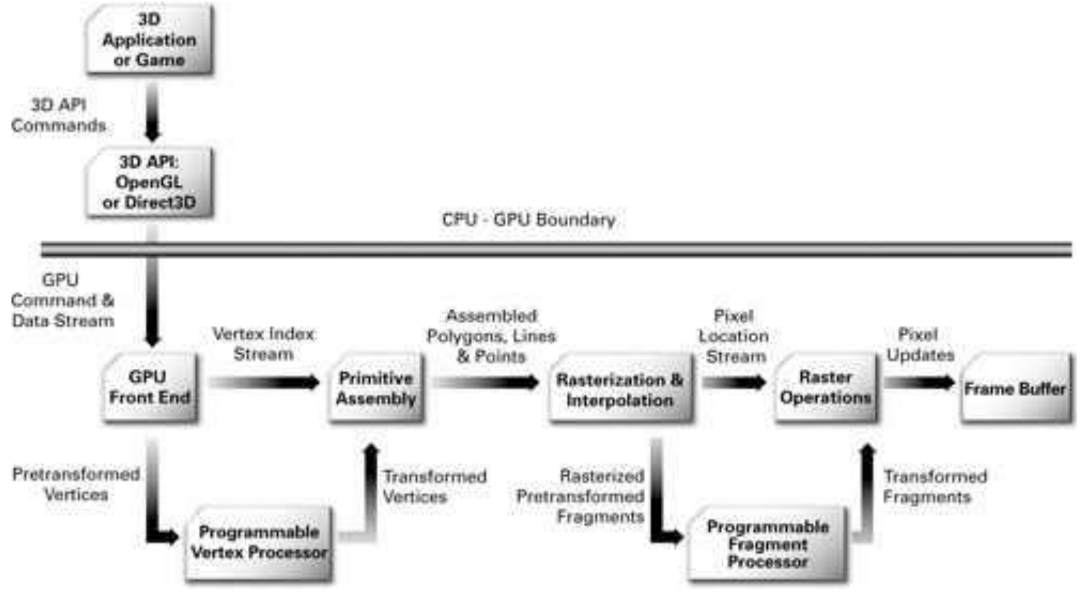


Figure 24 The programmable graphics pipeline

Figure 24 shows more detail than Figure 22, but more important, it shows the vertex and fragment processing broken out into programmable units. The programmable vertex processor is thhardware unit that runs your Cg vertex programs, whereas the programmable fragment processor is the unit that runs your Cg fragment programs.

As explained, GPU designs have evolved, and the vertex and fragment processors within the GPU have transitioned from being configurable to being programmable. The descriptions in the next two sections present the critical functional features of programmable vertex and fragment processors.

3.8.3.1. The Programmable Fragment Processor

Programmable fragment processors require many of the same math operations as programmable vertex processors do, but they also support texturing operations. Texturing operations enable the processor to access a texture image using a set of texture coordinates and then to return a filtered sample of the texture image.

Newer GPUs offer full support for floating-point values; older GPUs have more limited fixed-point data types. Even when floating-point operations are available, fragment operations are often more efficient when using lower precision data types. GPUs must process so many fragments at once that arbitrary branching is not available in current GPU generations, but this is likely to change over time as hardware evolves. Cg still allows you to write fragment programs that branch and iterate by simulating such constructs with conditional assignment operations or loop unrolling.

Figure 25 shows the flow chart for a current programmable fragment processor. As with a programmable vertex processor, the data flow involves executing a sequence of instructions until the program terminates. Again, there is a set of input registers. However, rather than vertex attributes, the fragment processor's read only input registers contain interpolated per-fragment parameters derived from the per-vertex parameters of the fragment's primitive. Read/write temporary registers store intermediate values. Write operations to write only output registers become the color and optionally the new depth of the fragment. Fragment program instructions include texture fetches.

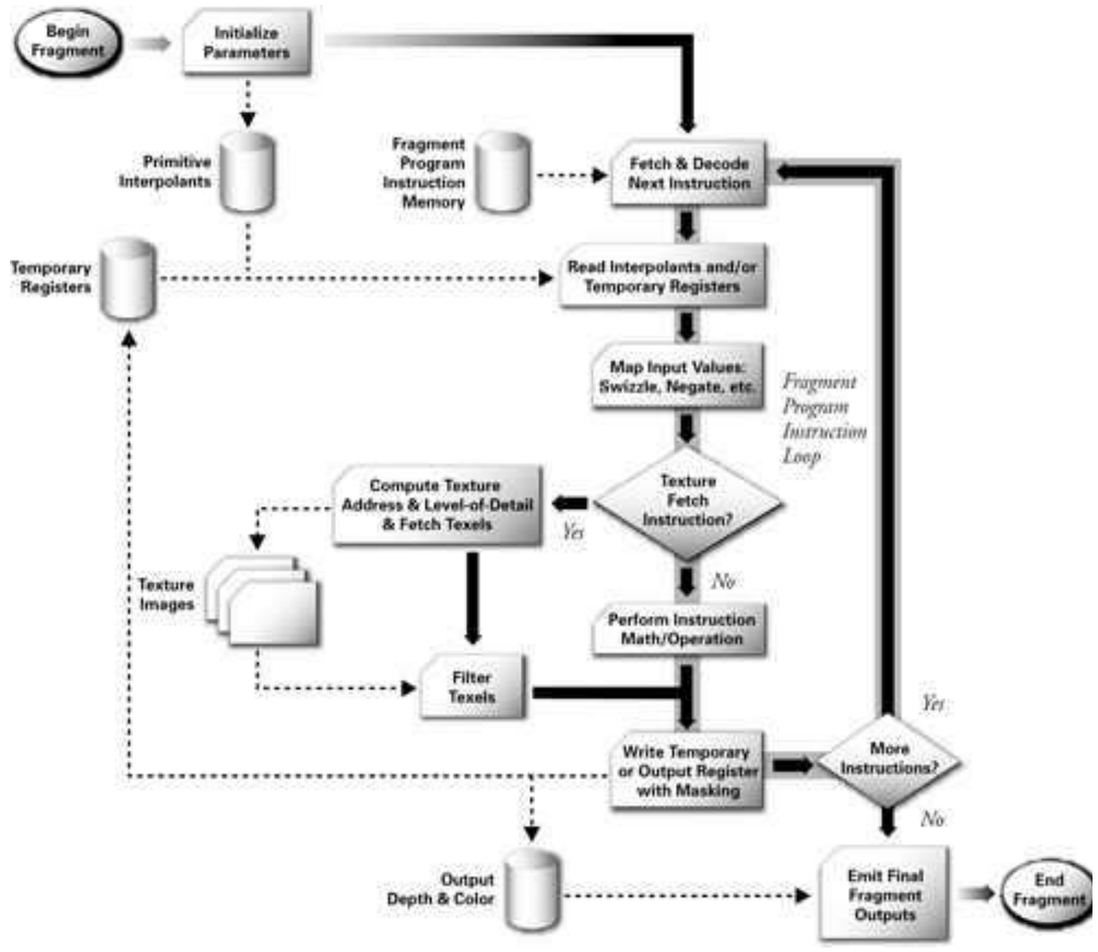


Figure 25. Programmable fragment processor flow chart

3.8.3.2. Cg Fragment Programmability

Cg's heritage comes from three sources, as shown in Figure 26. First, Cg bases its syntax and semantics on the general-purpose C programming language. Second, Cg incorporates many concepts from offline shading languages, as well as prior hardware shading languages developed by academia. Third, Cg bases its graphics functionality on the OpenGL and Direct3D programming interfaces for real time 3D.

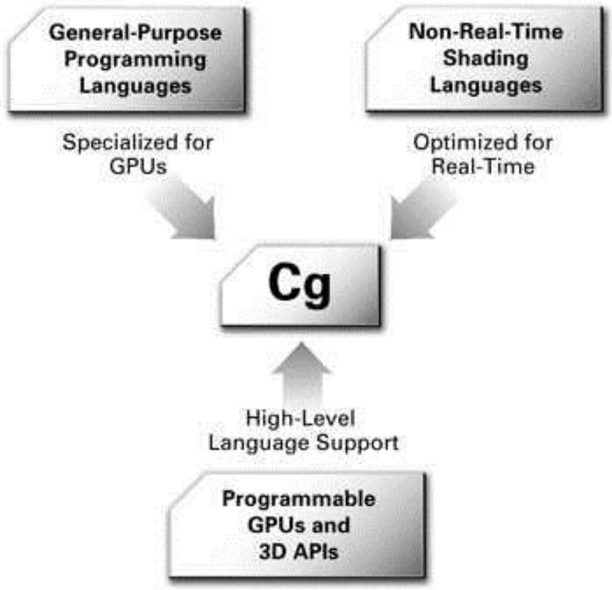


Figure 26. Sources of Cg's Technology Heritage

3.8.3.3. OpenGL

In the early 1990s, Silicon Graphics developed OpenGL in coordination with an organization called the OpenGL Architecture Review Board (ARB), which comprised all the major computer graphics system vendors. Originally, OpenGL run only on powerful UNIX graphics workstations. Microsoft, a founding member of the ARB, then implemented OpenGL as a way to support 3D graphics for its Windows NT operating system. Microsoft later added OpenGL support to Windows 95 and all of Microsoft's desktop operating systems.

OpenGL is not limited to a single operating or windowing system. In addition to supporting UNIX workstations and Windows PCs, OpenGL is supported by Apple for its Macintosh personal computers. Linux users can use either the Mesa open source implementation of OpenGL or a hardware-accelerated implementation such as NVIDIA's OpenGL driver for Linux. This flexibility makes OpenGL the industry's best cross platform programming interface for 3D graphics.

Over the last decade, OpenGL has evolved along with graphics hardware. OpenGL is extensible, meaning that OpenGL implementers can add new functionality to OpenGL in an incremental way. Today, scores of OpenGL extensions provide access to all the latest GPU features. This includes ARB standardized extensions for vertex and fragment programmability. As extensions are established, they are often rolled into the core OpenGL standard so that the standard as a whole advances. At the time of this writing, the current version of OpenGL is 1.4. Ongoing work to evolve OpenGL is underway in various OpenGL ARB working groups. This work includes both assembly level and high-level programmable interfaces. Because Cg operates as a layer above such interfaces, it will continue to function with future revisions of OpenGL in a compatible manner [27].

Chapter 4 : Visualization of Ultrasound Dataset

Four-dimensional (4D) ultrasound imaging extends the real-time capability of ultrasound to visualize a real-time volume that can be manipulated by the sonographer. Among the different visualization methods, surface rendering is a common mode for displaying volumetric datasets such as in obstetrical applications. A challenge in this mode is that surface shading is required to visualize the surface and enhances the surface contrast and this has very demanding computational requirements for 3D surfaces. Here, we present an optimized high performance rendering pipeline based on four stages for preprocessing, volume rendering, surface shading, and post-processing. The new approach is applied to render volumes acquired on a 4D commercial ultrasound imaging system with a research interface with results showing diagnostic quality and computational time suitable for real-time processing this suggesting potential for clinical utility.

4.1. Introduction

Volume rendering in medical imaging offers a display of a volumetric dataset after it is projected onto a plane using raycasting [19]. Many algorithms to extract surfaces of different structures inside the body such as bones and internal organs have been developed for imaging modalities with sufficient resolution and signal to noise ratio (SNR) such as magnetic resonance imaging and computed tomography [28]. However, surface extraction in ultrasound imaging applications is a much tougher problem given the lower resolution and SNR in addition to the presence of speckle as a characteristic texture in ultrasound images. Moreover, for volume scanning applications, reflected waves from specular reflectors vary with the direction of scanning unlike scattering. If we add to all that the real-time nature of ultrasound imaging that require such rendering to be done on the fly rather than offline as with other modalities, we realize how challenging this problem is.

Four-dimensional (4D) ultrasound surface rendering is a challenge for many reasons including noisy dataset, non-uniform volume sampling and processing time constraints. Two-dimensional (2D) cross sectional images of three-dimensional (3D) objects require realistic shading to create the illusion of depth [29]. Hence, the implementation of a complete system requires several steps. The first step is rendering the volume dataset considering the problem of polar sampling of ultrasound dataset, also surface detection must be take place while raycasting the volume to generate the Z-components of the rendered image, and due to the fuzzy nature of the ultrasound the volume rendering must include filtering of the data sets. The second step is Z-component filtration due to the noisy nature and surface discontinuities then uses the Z-components to shade the surface. Finally, the third step includes additional smoothing that enhances the rendered 2D image.

The used programming platforms for the implementation include OpenGL, which draws the volume cube and shading language that perform raycasting of the volume and coordinate transformation in addition to filtration. Alternatively, surface shading can be utilized where traditional surface rendering algorithms convert the volume data into geometric primitives in a process known as iso-surface extraction. The geometric primitives (polygon mesh) are then rendered to the screen by conventional computer graphics algorithms [30]. Another method is based on gradient calculation where the gradient vector is calculated first then used to represent the normal vector to be used for surface shading calculations [31].

Surface shading is an excellent method for allowing the operator to visualize the 3D shape of a structure [32]. The method is intuitive since humans perceive objects as solid structures that can be viewed from different perspectives to form an impression of their shape. The surface shading method aims to simulate this process in the digital world [16] and serves as a common tool for visualization in many imaging modalities. In ultrasound, the low SNR and parallel tissue boundary discontinuities make defining smooth surfaces difficult. Smoothing of a surface can be performed at the rendering stage. An approach was presented to render the fetal surface from the ultrasound 3D dataset [33]. First, the 3D dataset is smoothed using median and Gaussian filters. Then, a surface primitive is extracted and used for surface shading. Another approach renders smooth surfaces from 3D ultrasound based on the oriented splatting [34]. In this case, the surface is extracted based on the variational classification principle. Fuzzy surface rendering is done by oriented splatting whereby it creates triangles aligned with the gradient of the surface function. Such triangles are then colored with a Gaussian function and rendered in a back-to-front order. Another group proposed an improved surface rendering technique for 3D ultrasound data of fetuses [35] where modified anisotropic diffusion filtering is first applied to the dataset. In spite of the success of such methods in providing high quality rendering for their respective offline 3D visualization applications, such approaches were not intended for real-time processing in addition to not being optimized for implementation on the presently available GPU technology [36]. Several groups proposed GPU-based framework for the ultrasound data volume render [37] [36] with main focus on processing time. An approach that would combine the optimization methods used in offline 3D visualization and real-time processing on GPUs would offer an excellent platform for 4D imaging applications.

In this work, we are present an optimized framework for surface rendering of the 4D ultrasound volumetric data sets in real-time. This approach offers real-time performance while maintaining high quality of surface rendering as with offline 3D rendering methods. Image space Z-buffer shading is used [29], which postpones the surface shading to the last stage of the surface rendering. An additional task is added to the raycasting routine for edge detection, which can be implemented to get good results with modest computational effort. Working on a 2D texture to shade the surface offers optimized performance compared to working with 3D texture and allows more enhancements of surface smoothing and further additional post processing stages while still meeting real-time requirements. Such image space shading offers independent volume rendering and shading processing in the different stages. This allows pipelining for higher performance and makes it possible to process both the volume rendered and surface rendered images concurrently. The new approach is implemented and verified on a customized processing based on off-the-shelf components offering low cost of hardware.

4.2. **2D Ultrasound Acquisition System**

The imaging system used to conduct the experiments in this thesis was a DIGISON ultrasound imaging system (developed in Egypt by a team of engineers that include the author of this thesis). The system comprises 32 channels with 128 element probe, the signal from each channel is directly digitized after frontend preamplifier, followed by a Vertex 5 Field Programmable Gate Array (FPGA) (Xilinx, Inc.). The FPGA controls the firing sequence, handle the computer interface through the standard PCIe bus and buffer the data in its internal memory. The memory on this device is enough to store full 32 channel data of one complete ultrasound line. The FPGA supports up to 8 lanes PCIe interface protocol with up to 4 GB/s theoretical transfer rate (see Table 2), which offers a sufficient bandwidth to handle the huge data rates that result from digital ultrasound imaging analog front-end. The memory in the FPGA can hold data up to 4096 samples × 32 channels × 2 bytes/sample= 256 Kbytes. The system also utilizes 12bit analog-to-digital converters (ADC) with eight channels per chip and up to 50MHz sampling rate (Analog Devices, Inc.). Hence, the system uses 4 such chips to support the imaging system's 32-channel data acquisition. All the ADCs work synchronously using the same system clock and the FPGA then reads the 4 ADC chips concurrently and stores the data directly as raw data without any preprocessing or rearrangement to be ready for transfer to the reconstruction computer.

Table 2. PCI Express transfer rate.

Generation	Bit rate	Interconnect bandwidth	Bandwidth (per lane)	Maximum bandwidth (16 lanes)
PCI-Express 1.1	2.5GT/sec	2GB/sec	250MB/sec	8GB/sec
PCI-Express 2.0	5GT/sec	4GB/sec	500MB/sec	16GB/sec
PCI-Express 3.0	8GT/sec	8GB/sec	1GB/sec	32GB/sec

The computing platform used in this work was based on an Intel Core i3 processor. The Intel Core i3 is an x86-based multicore architecture which provides four cores (731M/1.17B transistors) on the same die. It features a super scalar out of-order core supporting 2-way hyper-threading and 4-wide SIMD parallelism. Each core is backed by 32 KB L1 and 256KB L2 caches, and all cores share an 8MB/12MB L3 cache. Quad and eight core CPUs provide 100 GFLOPS and 135 GFLOPS of peak single precision computation respectively, as well 32 GB/s of peak memory bandwidth.

4.3. 4D Acquisition and Rendering Hardware Description

The hardware setup used for the surface rendering is illustrated in Figure 27 and Figure 28. Imaging was performed using the DIGISON-Q ultrasound scanner. This scanner has independent 32-channel digital transmitters and receivers. The processing in the receive system filters and detects the digital RF lines which are subsequently converted and sent via fast PCIe interface to the digital processing computer. The processing platform is custom built from off-the-shelf hardware components comprising Intel® CoreTM i3 processor with 4 cores running at 3.3GHz each with 4 GB of DDR3-1600 RAM, a GA-B75M-D3H motherboard (GIGABYTE Technology Co., LTD) with PCI-E Gen3 expansion slots, and a NVidia GeForce GTX650 video card (NVIDIA Corporation, Santa Clara, USA) with 1027 MB of video RAM and 384 graphical processing unit (GPU) cores. The computer components were chosen to have sufficiently fast data transfer rates for real-time data acquisition, processing, and graphical display. The data are acquired directly from the ultrasound scanner through a single lane PCIe Gen3 with a maximum bandwidth of 1 GB/s per lane. The total cost of the hardware of the processing platform was less than 5000$, which is a very modest figure for such demanding application.

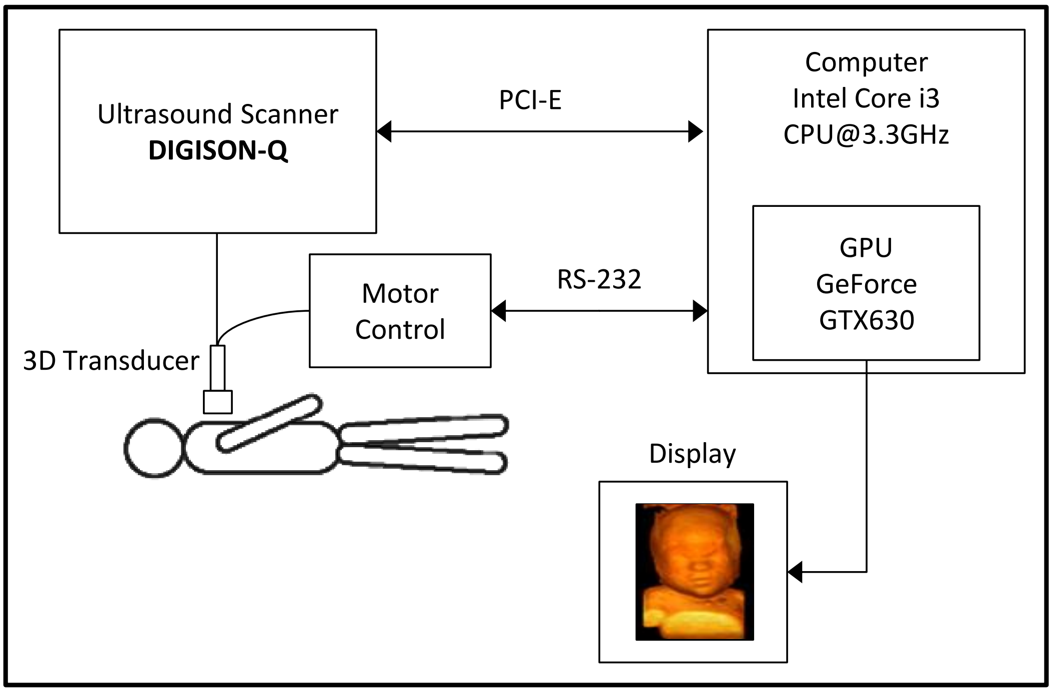


Figure 27. A block diagram of the system data acquisition hardware

The rendering is performed using the multicore GPUs on the programmable video card in real-time. The data acquisition, transfer, reconstruction and rendering processes are implemented in different layers with software developed using Microsoft Visual Studio 2010, the Microsoft DDK, Intel Parallel Studio, The Intel Math Kernel Library, OpenGL [38] and NVidia CG language [39]. Each detected ultrasound line of the imaging system is low-pass filtered then undersampled to 512 samples per line with quantization of 4 Bytes/sample. The ultrasound data from the acquisition hardware are copied directly to the computer memory as raw scan line data (image sticks) and then arranged in an ordered polar (θ,r) grid prior to transfer to the GPUs for further rendering processing.

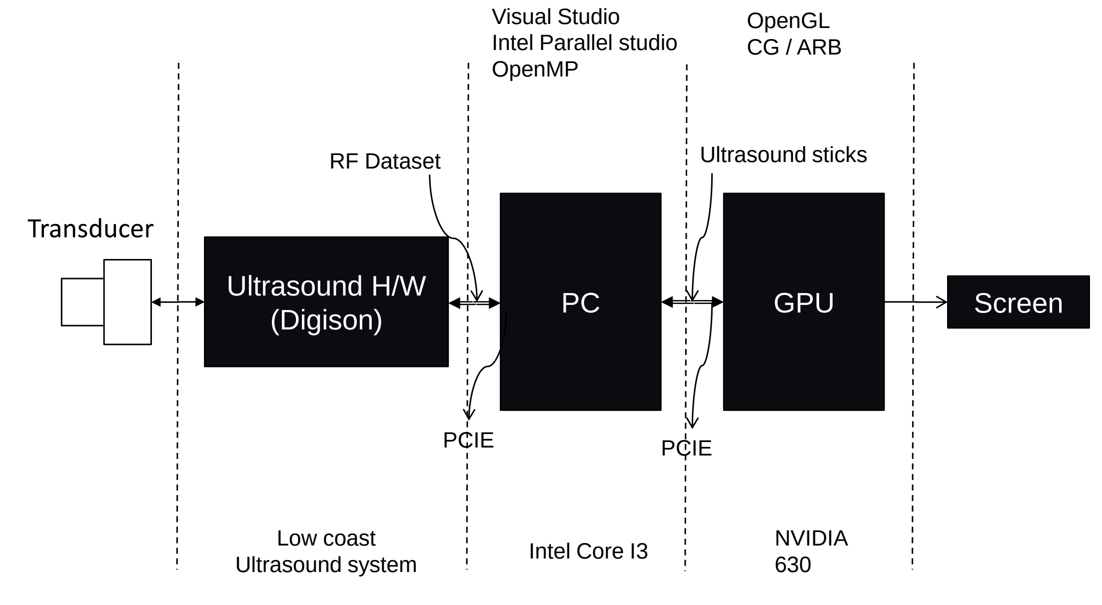


Figure 28. System blocks interconnection

4.4. Rendering Methodology

As the Frames are sent to the GPUs on the graphics card and before being integrated into the 3D texture memory, 2D image processing takes place in Pass (1). In Pass (1) after coordinate conversion, a homogenous filter is applied to smooth and slightly blur the scan-converted 2D image. The output from this stage is integrated directly into the 3D texture. In Pass (2), processing starts only after complete reception of all volume frames. Raycasting and Z-buffer calculation are applied in this stage. The Z-buffer is filtered using homogenous filter in the subsequent Pass (3) then used to shade the rendered 2D image. Finally, Pass (4) further applies a homogenous filter on the final image before display. The block diagram of the proposed processing methodology is shown in Figure 29.

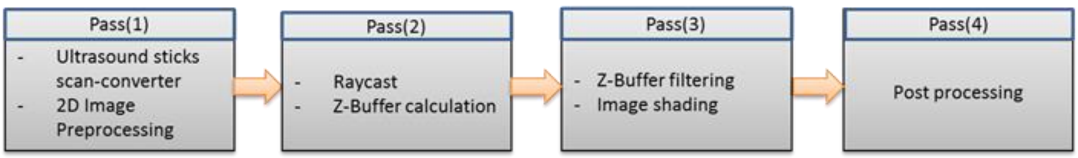


Figure 29. Rendering pipeline

4.4.1. **Pass (1): Preprocessing**

As the 4D ultrasound probe wobbles, ultrasound detected lines are sent to the GPUs synchronously. In this stage, we include a preprocessing step before integrating the ultrasound data to the 3D texture. The fragment shader uses a look-up table (LUT) to convert the image sticks from a polar (r,ϴ) to a Cartesian (x,y) coordinate system for each image, Figure 30 shows the ultrasound image before and after the scan conversion. Moreover, a homogenous filter is applied to the raw data. One of the major performance issues of this stage is skipping the empty spaces that result from the coordinate transformation process Figure 31. In this work, we use a third component flag in the LUT to skip the empty spaces in the scan-converted image. It should be noted that the parameters of the homogenous filter are not only selected to smooth the ultrasound images but also to add slight blurriness to the 2D image.

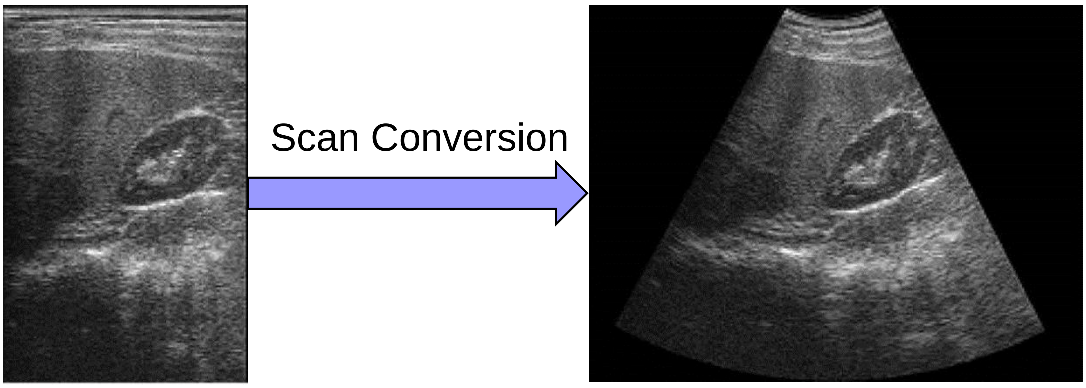


Figure 30. Ultrasound image before and after scan-conversion

The target of the first filter to smooth and add little blurriness, smoothing the 2D ultrasound image enhance the finale volume and the surface detection result, while if adding little blurriness to image data as shown in Figure 32 over-sharp surface silhouettes occasionally exhibited by volume renderings are softened [40] see Figure 33.

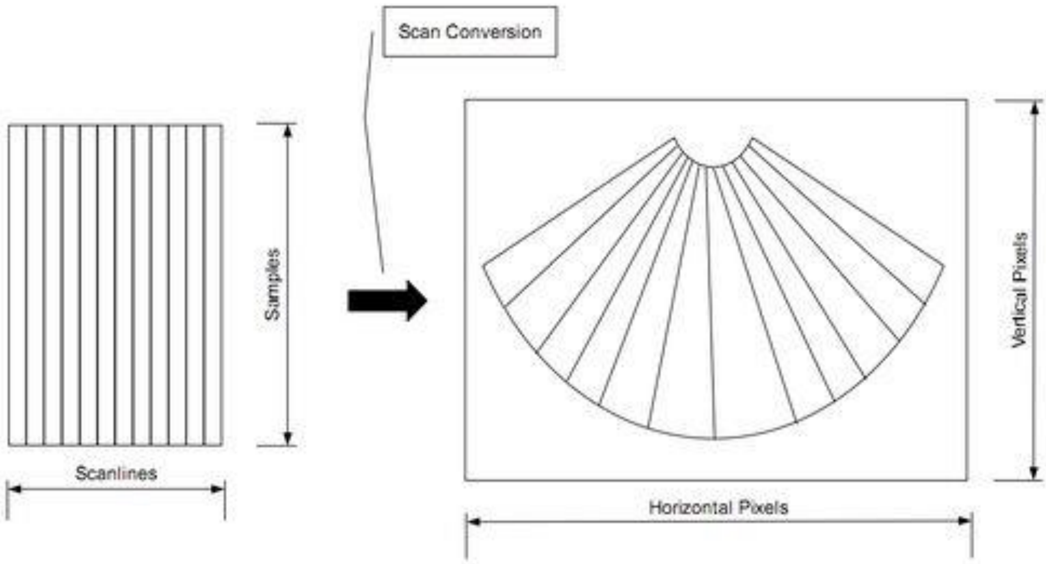
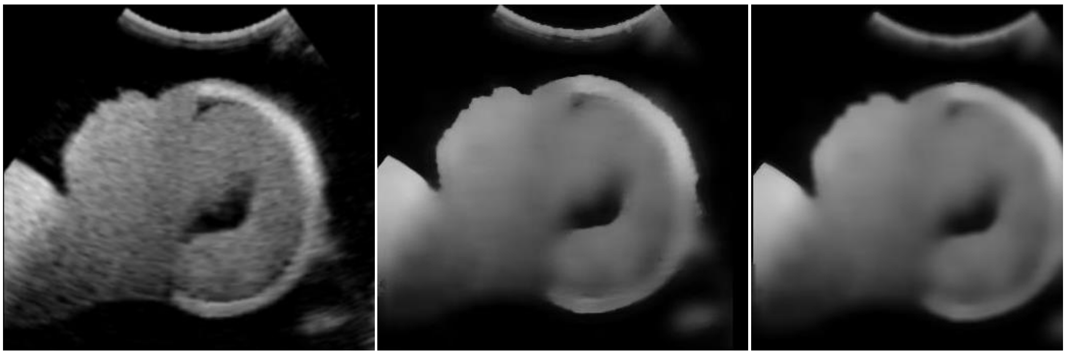


Figure 31. Scan conversion LUT and empty space skipping



a. after Scan-conversion b. after Speckle reduction c. after blurriness

Figure 32. 2D Scan-converted Image after speckle reduction with and without blurriness

Figure 33. Volume rendered image with wood-grain artifacts

4.4.2. Pass (2): Volume Rendering

This processing stage takes care of transforming the volume data from polar to Cartesian coordinate in the probe motion direction also using a LUT based method (since the in-plane direction transformation is done in Pass (1)) Figure 31. Then, volume raycasting (Figure 36) is performed to the transformed data where 3D Texture is used with a trilinear interpolation local illumination models [41] as follows. The Raycasting model is to directly evaluate the volume rendering integral along rays that are traversed from the viewing plan. For each pixel in the image, a single ray is cast into the volume. Then the volume data is resampled at discrete positions along the ray. By means of the transfer function, the scalar data values are mapped to optical properties that are the basis for accumulating light information along the ray. Typically, compositing can be performed in the same order as the ray traversal. Therefore, front-to-back compositing is applied as [42]:

$$C_{dst} = C_{dst} + (1 - \propto_{dst})C_{src} \ . \tag{4-1}$$

$$\propto_{dst} = \propto_{dst} + (1 - \propto_{dst}) \propto_{src} \tag{4-2}$$

C: Maps the ultrasound scalar value to color value(RGB)

$\propto$: Maps the ultrasound scalar value to opacity value(RGB)

Two curves for scalar value to color and alpha value are used. The two curves are stored in the same 1D texture memory. For each scalar value, RGB color and alpha value are stored see Figure 34 and Figure 35.

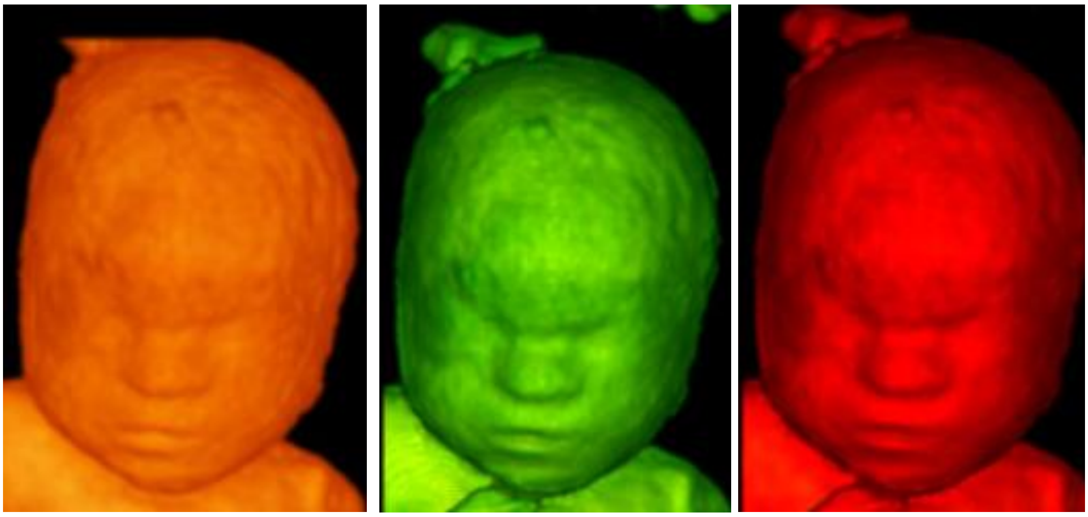


Figure 34. Deferent coloring transfer function

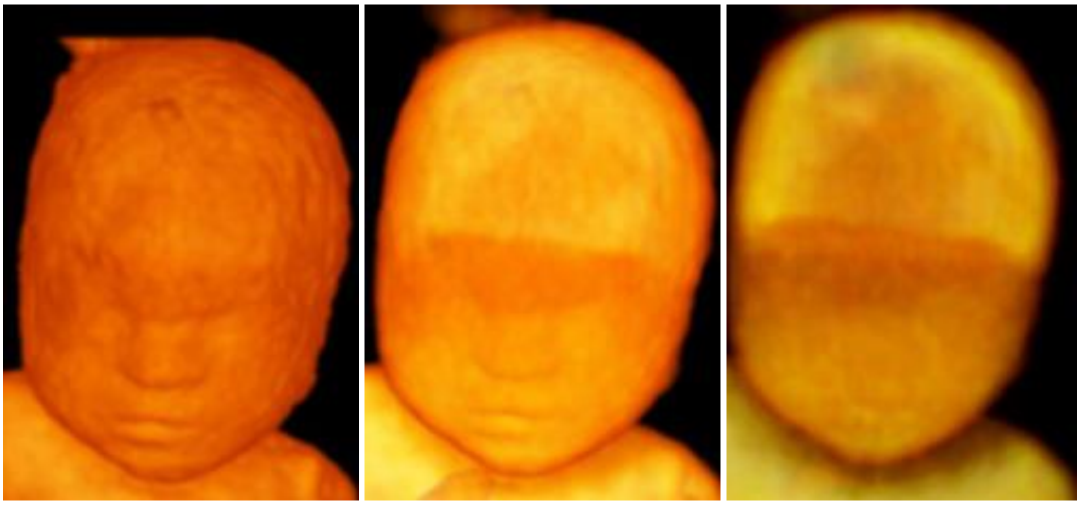


Figure 35 Deferent alpha transfer function

The implementation here uses a parallel projection raycasting model without loss of generality (current implementation is also applicable on the perspective projection with a slightly different point to plane distance calculation). Figure 37 illustrates raycasting geometry. Alpha-blending is subsequently applied to the resultant volume [19] followed by a 3D smoothing Gaussian filter. This 3D filter kernel is calculated in the CPU and sent to the fragment program as a float arra By pipelining the samples, there is no need to read the entire 3D rendered volume samples but rather it is possible to read the front slice of samples and discards the oldest slice and so on to optimize the data transfer [43].

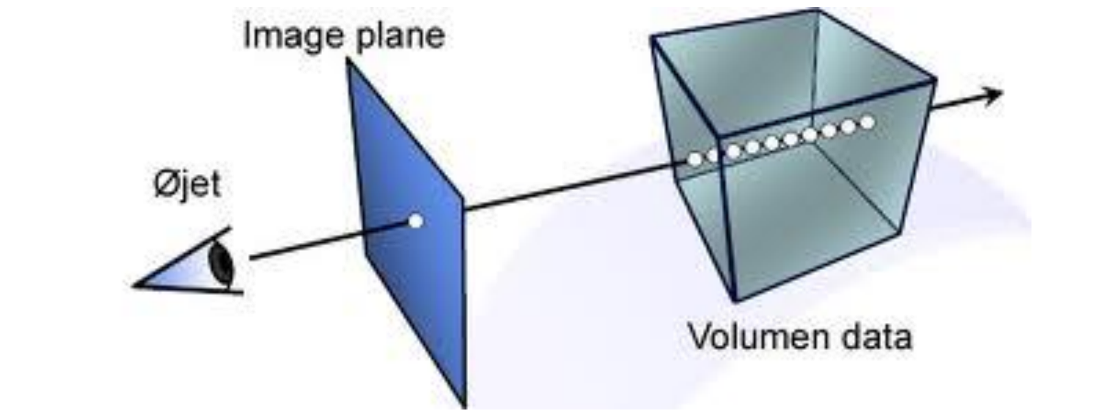


Figure 36. Volume raycast

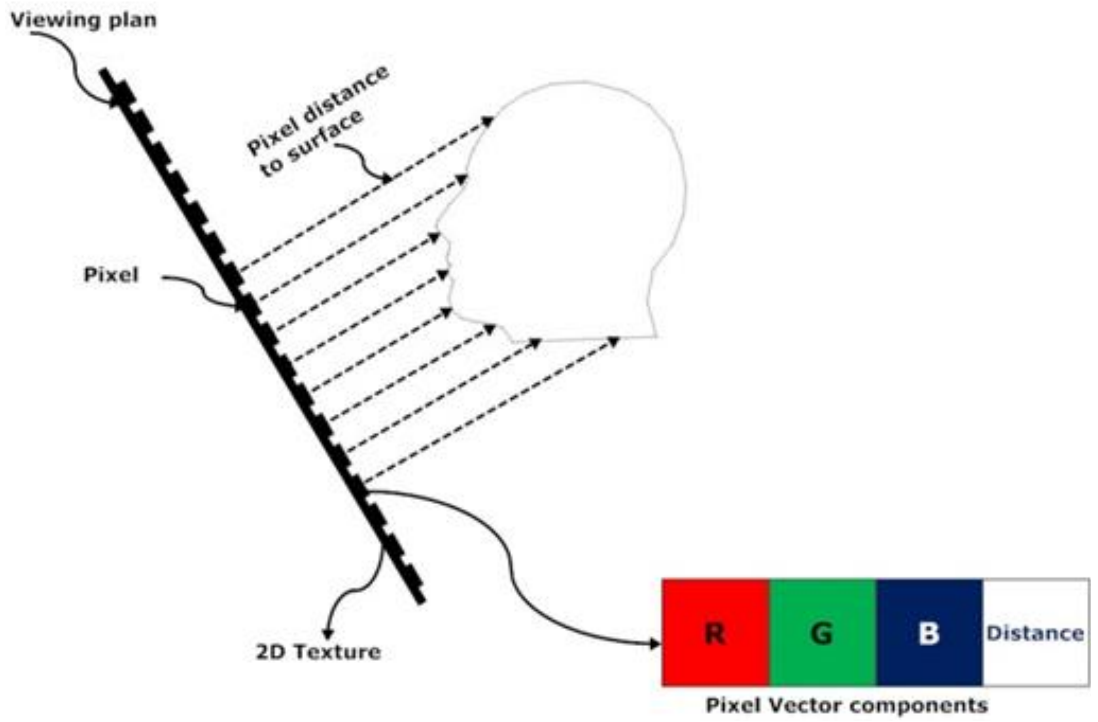


Figure 37. Viewing plan with respect to the volume data set and distance map calculation. Lower right show the pixel vector components

In this pass the deferent 3D data viewing techniques are implemented, volume Render(VR), Surface render(SR), Maximum Intensity Projection(MIR), Minimum Intensity projection, Average Intensity Projection and orthogonal views and Multiple Planner Reformation(MPR). See section 3.6 See Figure 38.

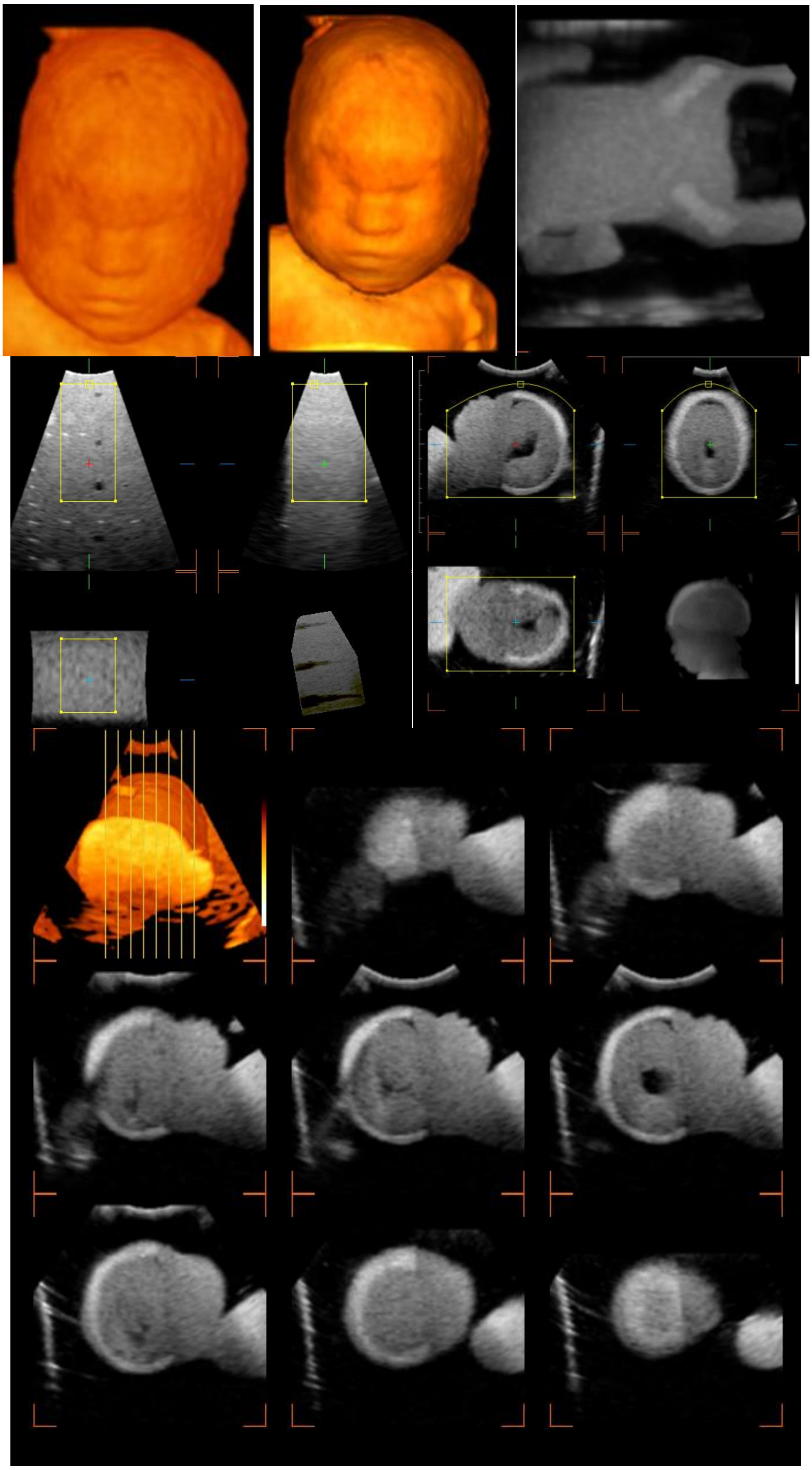


Figure 38. Deferent rendering modes from left to right and from top to bottom A. (VR)Volume render B. (SR)Surface render C. (MIP) Maximum intensity projection D. Minimum intensity projection E. (X-Ray) Average projection F. MPR Multiple planner reformation

4.4.2.1. Surface Detection

As the last step, edge detection and z-buffer formation are performed. Such steps take advantage of the special form of the ultrasound imaging data especially in such cases when imaging the fetus face. Since the fetus-face is preceded by amniotic fluid that produces very little reflected ultrasound signal compared to the baby face that acts close to a specular reflector, an excellent contrast between them is obtained. While raycasting the volume, the data samples are saved in a circular buffer with a reasonable length, (A buffer length of 8 was sufficient for the results of this work). We compute the following measure,

$$Difference = \sum_{i=0}^{i=\frac{n}{2}-1} data(i) - \sum_{i=\frac{n}{2}}^{i=n-1} data(i) \qquad \textbf{(4-3)}$$

Here, *n* is the buffer length. We compare the *Difference* value to a user selectable threshold to do the segmentation. In case of a surface point, we store the length of the ray at this point in the 4th element of the rendered pixel vector, this value will present the distance to the viewing plane (or the Z-Component).

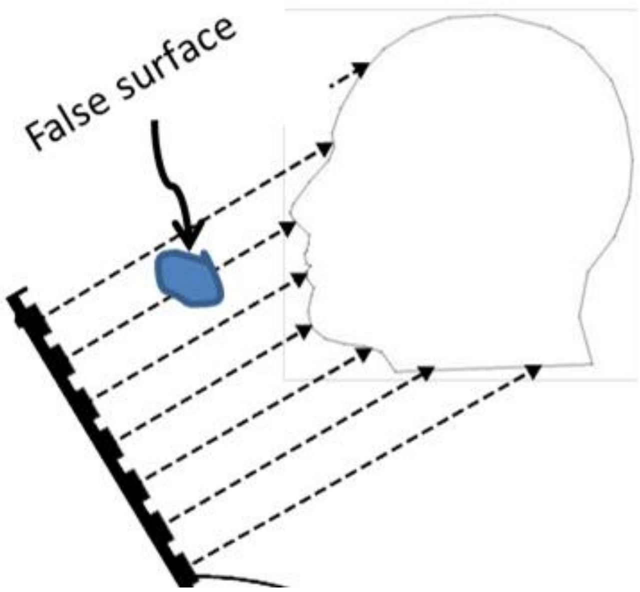


Figure 39. False Surface

Two deference kernels are used, the first one (Rough detector) have a larger step size to bypass the small volumes, which we consider as a false surface see Figure 39, after the rough detector detects the true surface a finer one is used to catch the surface at accurate position.

It should be emphasized that the threshold value is a user-controlled parameter that he/she can change from the user-interface to enable distinguishing different surfaces of the volume based on this selection. Also taking the sign of the *Difference* into consideration can be used to detect different surfaces of the volume such as detecting the surface of fetal face versus detecting the surface of a fluid-filled structure such as the uterus.

4.4.2.2. Motion Compensation

As described in Chapter 2 the ultrasound frame is formed from subsequent lines. Each line take a time depends on the flying depth(13uSec/cm), and while forming the frame the 2D probe is moving in wobbling direction, that leads to error in the 2D rendered frame position. a motion compensation is done, while raycasting the volume. A LUT is used for space correction due to the motion. Figure 40 shows how the correction is done, Speed ratio or Tan(Ɵ) is calculated in the CPU and sent to the GPU. The correction done real time and does not need significant calculation it is just one division operation for each X position substitute it in the equation to get the correct Y, use the corrected X,Y to Fitch the 3D texture.

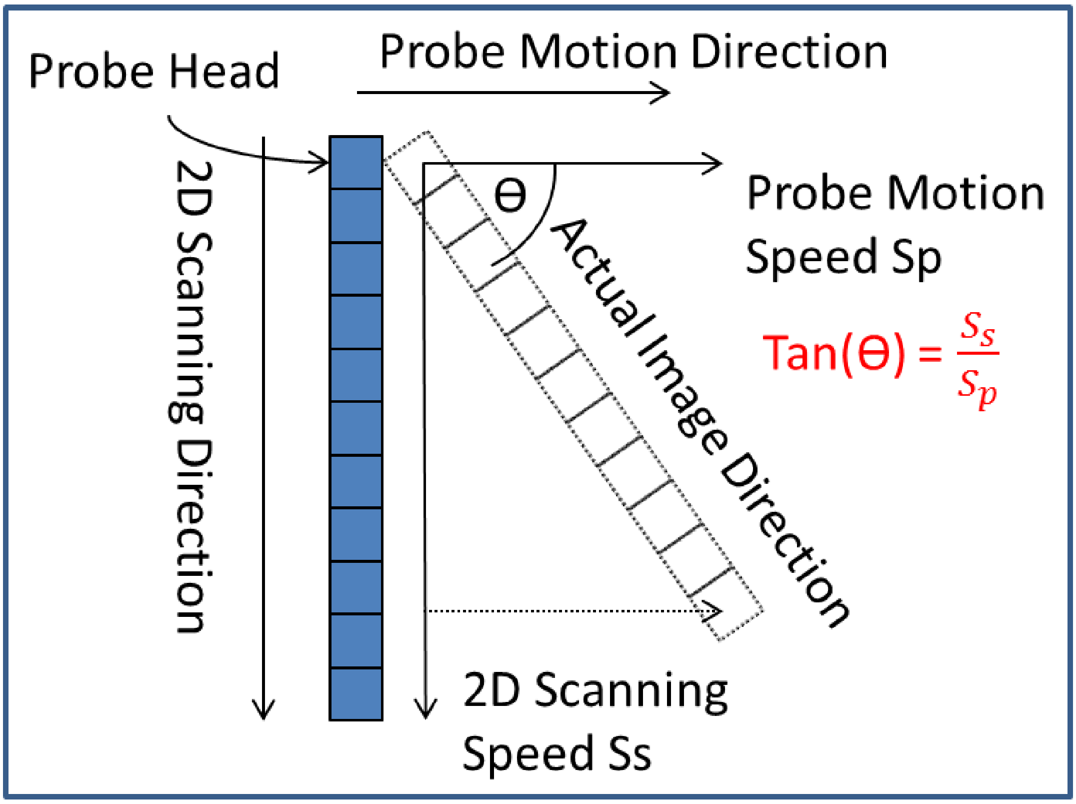


Figure 40. Probe scanning versus probe motion and the actual image direction

4.4.3. Pass (3): Surface Shading

This stage represents the lighting stage of the rendered volume. This stage begins by filtering the rendered Z-buffer components using a homogenous filter as in 4.5. Then, the 2D rendered image is appropriately shaded using image space shading [29] and local illumination models [41] as follows.

$$I_{Phong} = I_{ambient} + I_{diffuse} + I_{speculer} \quad (4\text{-}4)$$

$$I_{ambient} = color \tag{4-5}$$

$$I_{diffuse} = k_d I_{light} cos\theta = k_d I_{light}(N \, . \, L) \tag{4-6}$$

$$I_{specular} = k_s I_{light} \, (cos \; \phi)^{shininess} = k_s I_{light} \, (N.H)^{shininess} \tag{4-7}$$

Here,

N: Unit normal vector

L: Unit light vector

R: Unit reflection vector

V: Unit viewer vector

H: V+L

A point light source is used, while a directional light source is still applicable. Point light source enhances the contrast of the fetus face. The Blinn-Phong model is used for the surface illumination where it computes the light reflected by an object as a combination of three different terms, ambient, diffuse, and specular terms as [41]. Figure 43 shows the shaded volume image with deferent light source position. Because of the shading is done on a 2D image the moving of the light source done real-time based. Also shading the volume does not need to re-render the volume again.

The shadowing illumination components are first calculated by testing each surface pixel if it is shadowed or not. Using raytracing back from each pixel to the light source and if there is no other pixel obstructs the ray this pixel considered as lighted pixel and if not it is a shadowed pixel. If a shadowed pixel is found, reduction of the pixel value by the shadowing coefficient constant is done. The geometry of the process using the map of voxel heights from the viewing plane is illustrated in Figure 41. It should be noted that shadowing noticeably affects the overall performance due to the raytracing needed. However, it is still achievable in our system using a modest graphics card because it is still done on 2D texture data rather than 3D texture data.

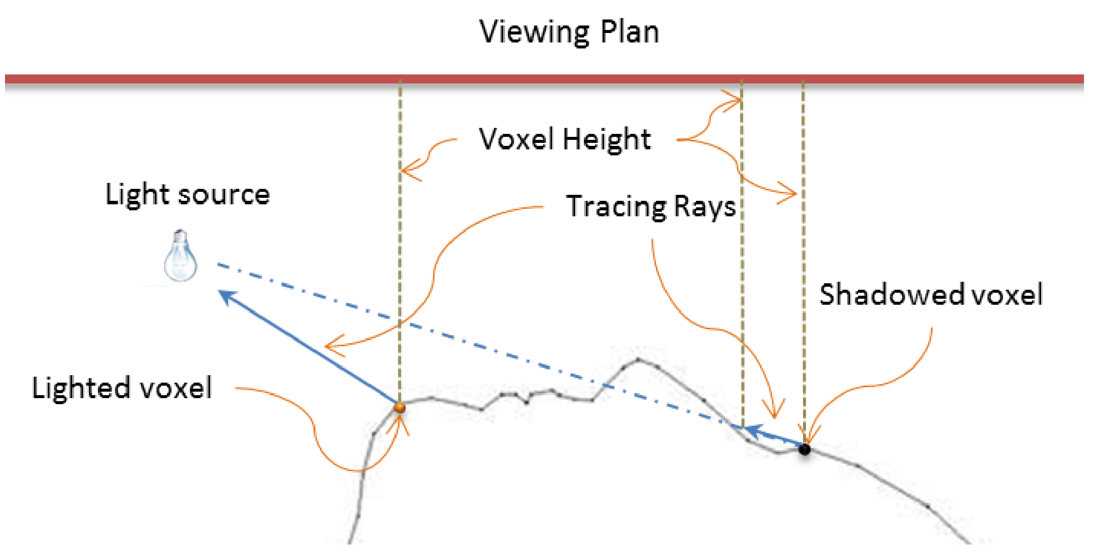


Figure 41. Shadow Ray tracing using Distance map

Filtration is done on the Z-Buffer component and works as a sand paper smoothing of the rendered surface. Surface filtration level is selectable by the user according to the current application.

As shown in Figure 42 if the design of the distance map filter did not preserve the edges, a new false surface edge will be generated. Therefore, that is one of the main targets of the filter in this stage significantly smooth the surface while preserving the edges.

The main advantage of this approach is that the filtration is performed on a 2D texture dataset, which makes the memory fetching performance better than working on a 3D texture.

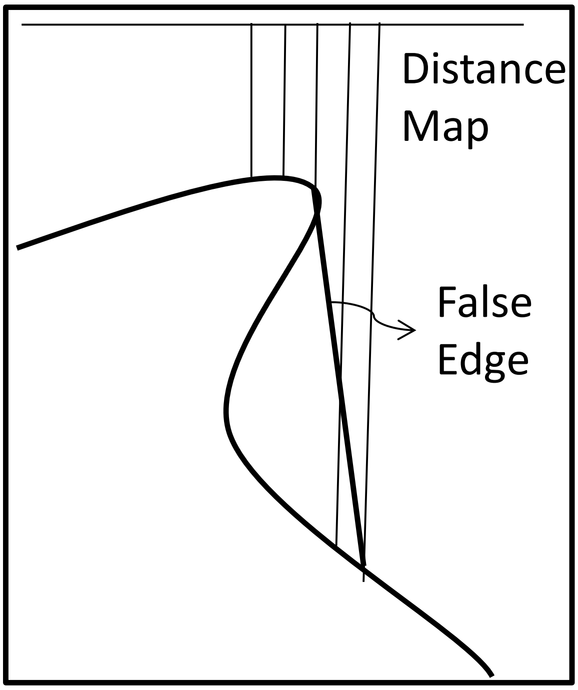


Figure 42. Distance map false edge generation due to undesired filtration

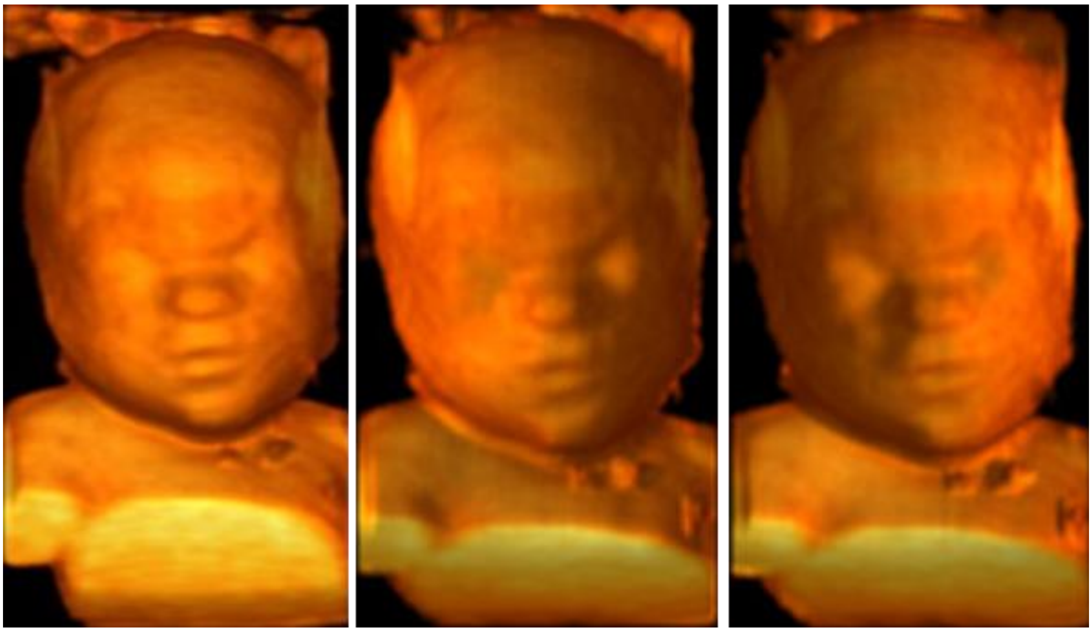


Figure 43. Deferent Shaded Image with deferent light position, left image with light source at the front, Middle Image Light source at front left, right Image with light source at the front right.

4.4.4. Pass (4): Post-processing

To enhance the output image, a homogenous Sigma filter is applied on the final shaded image [44]. Smoothing and preserving the edges is the target. Figure 44 and Figure 45 show the effect of the post-processing filtration.

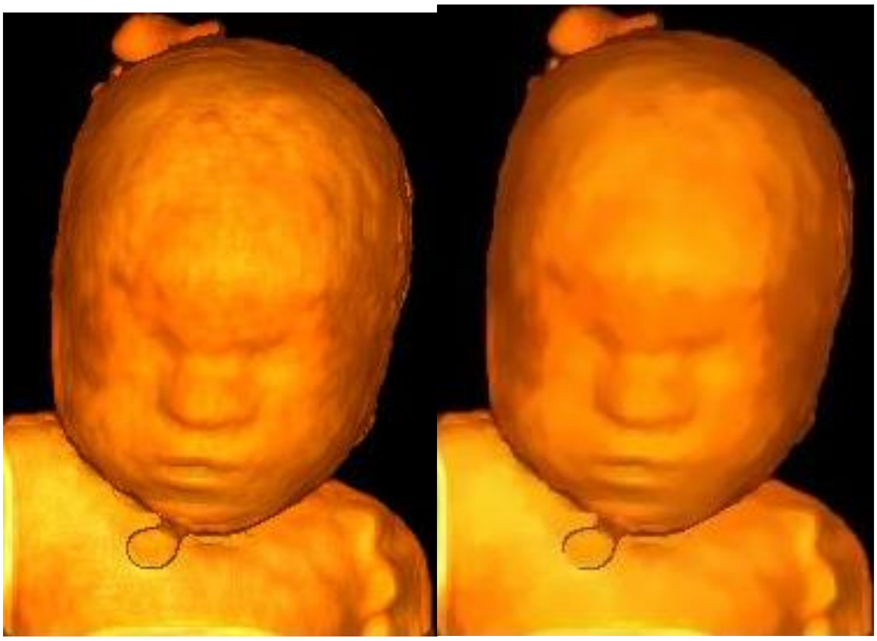


Figure 44. Post-Processing (Sigma filter) of surface rendered image, left image (before), right image (after)

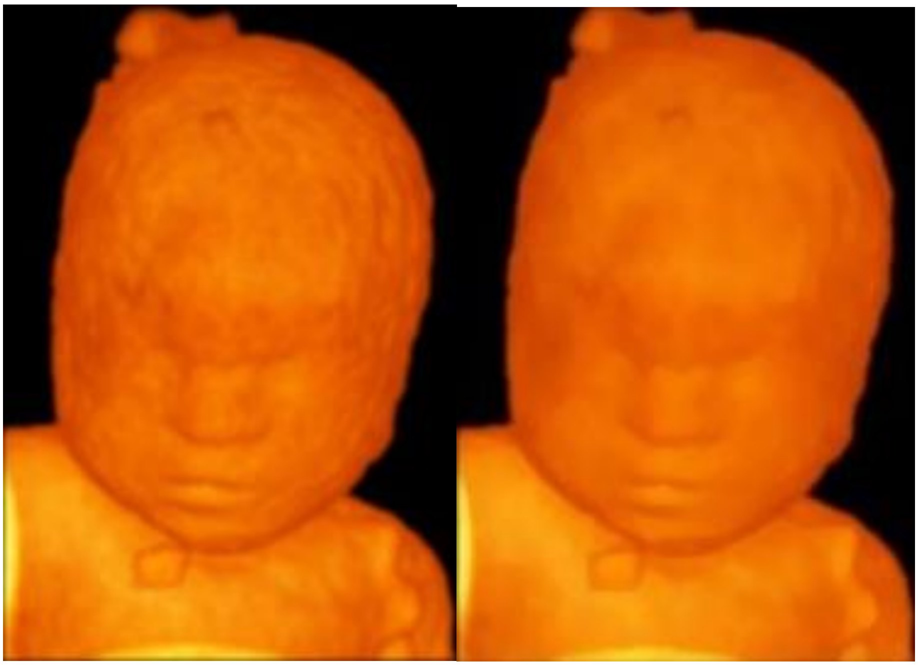


Figure 45. Post-processing (Sigma filter) of volume rendered image, left image (before), right image (after)

4.5. **Results and Discussion**

The time to collect an ultrasound volume assuming 35 frames per volume and a depth of 16 cm can be calculated from the echo-ranging theory to be around 466 ms (since the time to receive one line from a depth d is equal to 2d/c where c is the speed of ultrasound and the total time is this time multiplied by number of lines per image and number of images in the scanned volume) [8]. This time becomes shorter when the number of ultrasound frames per volume becomes less or the depth of scanning is decreased. Hence, this time maps directly to the quality of the rendered volume. Figure 43 shows sample 4D rendering results using data collected by scanning a 3D ultrasound training phantom (CIRS, Inc. Model068) with different light positions showing light source from the front and front-left directions. In addition, in Figure 44 and Figure 45 presents the results of rendering with (a) low filtration, and (b) high filtration kernel sizes. The results show good diagnostic quality of rendering which was performed in real-time.

To obtain quantitative results, a number of experiments were performed to compute the rendering time variation with the different imaging parameters to assess the practicality of the developed system. The results of the variation of the rendering time with number of frames within the rendered volume are shown in Table 3 (for surface rendering mode, 3D Kernel size of 3x3x3, 2D homogeneous filter kernel size of 9x9, image sector size of 77^o, 3D scanning Fan angle of 61^o and volume size of 512x512x512). It is clear from the results that such variations do not have a high order of complexity as suggested by the small differences in time for the whole range of values considered. Table 4 presents the results of the variation of the rendering time with size of rendered volume (for surface rendering mode, 3D Kernel size of 3x3x3, 2D homogeneous filter kernel size of 9x9, image sector size of 77^o, 3D scanning Fan angle of 61^o and 37 frames per volume). In this case, the variation show a linear O(n) variation with size of data in each dimension. Moreover, the study of the variation of the rendering time with size of 3D filter kernel is presented in (for surface rendering mode, 2D homogeneous filter kernel size of 9x9, image sector size of 77^o, 3D scanning Fan angle of 61^o, 37 frames per volume and volume size of 512x512x512) and shows a linear variation with number of points in the 3D filter kernel as expected. On the other hand, the variation of the rendering time with size of 2D homogeneous filter kernel presented in Table 5 suggest very small variation of rendering time with 2D homogeneous kernel size (for surface rendering mode, 3D filter kernel size of 3x3x3, image sector size of 77^o, 3D scanning Fan angle of 61^o, 37 frames per volume and volume size of 512x512x512). As a result, it is clear that the key issue in the rendering time is the choice of 3D filtration kernel size. Also, the results demonstrate that the performance offered by this midrange graphics card for different practical settings is acceptable and fulfill the timing constraints of real-time rendering. In other words, the frame rate in the developed system is limited only by the data collection speed not the reconstruction and visualization.

Table 3. Rendering time variation with number of frames per volume

Frames per volume	Render timer (mSec)
27	53.7
31	54.0
35	54.6
37	54.6

Table 4. Rendering time variation with volume size

Volume data size	Render time (mSec)
512x512x512	54.6
384x384x384	35.0
256x256x256	15.8
128x128x128	13.2

Table 5. Rendering Time variation with 3D filter kernel size

3D Filter Kernel size	Render time (mSec)
No Filtration	21.5
3x3x3	54.6
5x5x5	187

Table 6. Rendering time variation with 2D homogeneous filter size

Homogenous filter	Render time (mSec)
3x3	54.3
5x5	54.9
7x7	55.5
11x11	56.8

Chapter 5 : Ultrasound Image Processing

Introduction

The B-mode ultrasonic images exhibit a granular appearance, called speckle pattern, which is caused by the constructive and destructive interferences of the wavelets scattered by the tissue components as they arrive at the transducer surface [45]. This speckle appearance very much resembles the speckle pattern that results from laser scattering by a rough surface. If the incident ultrasound beam is totally coherent like the laser beam, the speckle carries no information about the microstructure of the tissues. Fortunately, ultrasonic scanners use partially coherent incident waves, i.e., pulses. Thus, the speckle patterns exhibited by tissues contain useful information about the structures of the tissues that can be used clinically for tissue differentiation.

The resemblance between laser and ultrasound speckles has been extensively analyzed [45]. The histogram of the video signals or echo amplitude returned from tissues or the number of occurrences plotted as a function of the amplitude of these echoes, Follows a Rician distribution similar to the distribution of the magnitude of a phasor $V = X + jY$ with a uniform phase. The symbol σ^2 denotes the variance of the real component ,X. or imaginary component, Y. This means that the signal contains random and ordered components. If there are no ordered components, the histogram should follow a Rayleigh distribution.

The question of whether the speckle is a friend or foe has been debated for many years. On the one hand, speckles provide diagnostic information for clinicians to make a diagnosis. A clear example is that different organs exhibit different speckle or textural patterns and tumors frequently exhibit different speckle patterns from normal tissues. On the other hand, speckles degrade spatial resolution of the imaging system. Smaller objects may be obscured by the speckles. The most optimal resolution appears to smooth out somewhat the speckle pattern while maintaining as much as possible the spatial resolution and the frame rate. Frame averaging via spatial compounding or frequency compounding has been studied and implemented in commercial scanners. Spatial or frequency compounding describes a signal processing method in which multiple frames are acquired at different imaging angles or spatial positions or at different frequencies and subsequently averaged to form one frame of image [9].

5.1. **Speckle Noise Characteristics**

In an ultrasound (US) imaging system, US waves are Transmitted/Received from the area of interest on the body. These waves pass through the skin of the patient and get reflected at the tissue interfaces. The reflected waves are envelope detected and displayed as a two-dimensional image.

Most of the biological tissues consist of cells smaller in size than the acoustic wavelength; the signal acquired within a resolution cell comprises reflections of many independent scatterers. These scatterers result in de-phased echoes causing interference (either constructive or destructive) and producing complex interference pattern termed as speckle [11]. Speckle is a type of noise which generally masks the fine details of the US image, thereby making the interpretation of an US image difficult for medical diagnosis. As a result, de-speckling are necessary for enhancing the image quality, increasing the diagnostic value of medical US images and prepare the image for 3D volume rendering process [46].

Speckle Noise is multiplicative in nature. This type of noise usually occurs in almost all coherent imaging systems such as laser, acoustics and SAR(Synthetic Aperture Radar) imagery. This type of noise is an inherent property of medical ultrasound imaging.

Speckle is present in both RF data and envelope-detected data. When a coherent component is introduced to the speckle, it adds a constant strong phasor to the diffuse scatterers echoes and shifts the mean of the complex echo signal away from the origin in the complex plane. Upon detection, this has the effect of changing the Rayleigh PDF into a Rician PDF.

5.2. Speckled Image Model

It is well established that the fully developed speckle is a multiplicative noise [44] [47] and can be modeled as

$$y_{i,j} = x_{i,j}\, n_{i,j} + a_{i,j} \qquad \textbf{(5-1)}$$

where, *y* and *x* are the observed noisy image and noise free image, respectively, *n* is the noise variable modeled as a stationary unity mean random variable independent of *x*. *a* is the additive noise which of negligible effect with respect to the multiplicative noise.

The de-speckling methods take the advantage of the logarithmic transformation that, when applied to both sides of (5-1), converts the multiplicative noise into an additive one. The logarithms of *x*, *y*, and *n* by $x1$, $y1$, and *n*l, respectively, Equation (5-1) can be rewritten as

$$y1_{i,j} = x1_{i,j} + n1_{i,j} \qquad \textbf{(5-2)}$$

Thus, the speckle reduction problem of Equation (5-2) becomes a conventional problem of additive noise removal.

5.3. Lee Sigma Filter

Lee Filter [48] use local statistics to effectively preserve edges. This filter is based on the approach that if the variance over an area is low or constant, then smoothing will be performed, otherwise smoothing will not be performed if variance is high(near edges).

Most image noise is Gaussian in distribution. The two-sigma probability is defined As the probability of a random variable being within two standard deviations of its mean. The two-sigma probability for a one-dimensional Gaussian distribution is 0.955.

This can be interpreted as meaning that 95.5% of random samples lie within the range of two standard deviations. In image smoothing, any pixel outside the two-sigma range most likely comes from a different population and, therefore, should be excluded from the average. If we assume that the a priori mean is the gray level of the pixel to be smoothed, we can establish a two-sigma range from the gray level and include in the average only those pixels within the two-sigma intensity. Let $x_{i,j}$ be the intensity or gray level of pixel (i, j), and $\hat{x}_{i,j}$ be the smoothed pixel (i, j). The sigma filter procedure is then described as follows:

1- Establish an intensity range ($x_{i,j} + \Delta$, $x_{i,j} - \Delta$) where $\Delta = 2\sigma$.
2- Sum all Pixels which lie within the intensity range in a ($2n + 1$, $2m + 1$) window.
3- Compute the average by dividing the sum by the number of pixels in the sum.
4- Then $\hat{x}_{i,j}$ = the average (to reduce sharp pot noise, step (4) will be modified later in this section.)

Or, Mathematically, let

$$\delta_{k,l} = 1 \qquad if\ (x_{i,j} - \Delta) \leq x_{k,l} \leq (x_{i,j} + \Delta) \qquad \textbf{(5-3)}$$
$$= 0 \qquad \text{otherwise.}$$

Then

$$\hat{x}_{i,j} = \sum_{k=i-n}^{n+i} \sum_{l=j-m}^{m+j} \delta_{k,l}\ x_{k,l} \Big/ \sum_{k=i-n}^{n+i} \sum_{l=j-m}^{m+j} \delta_{k,l} \qquad \textbf{(5-4)}$$

$$\hat{x}_{i,j} = \text{two-sigma average,} \quad \text{if } M > K \qquad \textbf{(5-5)}$$
$$= \text{immediate neighbor average,} \quad \text{if } M \leq K \ [48]$$

The two-sigma range is generally large enough to include 95.5% of the pixels from the same distribution in the window, yet in most cases, it is small enough to exclude pixels representing high contrast edges and subtle details. Linear features such as roads one or two pixels wide are retained, because only those pixels with intensity near that of the feature are included in the average. The main drawback is that sharp spot noise represented by clusters of one or two pixels will not be smoothed. This could be very annoying especially for a noisy image. To remedy this, we shall replace the two-sigma average with the center pixel's immediate neighbor average, if M, the number of pixels within the intensity range, is less than a prespecified value K. In other words, step (4) is replaced by equation (5-5) the value of K should be carefully chosen to remove isolated spot noise without destroying this features and subtle details. For a 7 × 7 window, K should be less than 4, and it should be less than 3 for a 5 × 5 window. It should note that subtle textures within the two-sigma range would be wiped out after a few iterations. If conservation of texture information is required, a small Δ range and one or two iterations should be used.

5.3.1. Locally Smoothing Algorithms

Numerous local image-smoothing algorithms have been developed recently. The straight local average method is known to blur edges and details. A template matching technique [49] to detect edges and lines and then replaced the pixel by a weighted average corresponding to the particular pattern detected. Twelve 3 × 3 masks are created and relatively complicated weighting schemes are proposed. This algorithm is not computationally efficient, nor is it very effective in smoothing noise, since the window size is small. Lee proposed algorithms for the local pixel calculation and Gradient inverse method [50]. The edge preserving smoothing scheme of Nagao and Matsuyama [51] and median filter were also used.

5.3.1.1. Gradient Inverse Method

The gradient inverse weighting scheme employs a 3 × 3 window and computes for each pixel its inverse gradient weighted average with its neighboring pixels. The idea is to put less weight on those pixels having greater absolute differences with their center pixel. The procedure for processing $x_{i,j}$ in a 3 × 3 window is given as follows:

(1) Compute the inverse gradients of the eight neighboring pixels:

$$g_{k,l} = 1/\left|x_{i+k,j+l} - x_{i,j}\right| \qquad if\ x_{i+k,j+l} \neq x_{i,j}$$
$$= ½ \qquad if\ x_{i+k,j+l} = x_{i,j}. \tag{5-6}$$

Where k, l = {-1,0,+1}

(2) Compute weights for the eight neighbors:

$$w_{k,l} = \frac{1}{2} \cdot \frac{g_{k,l}}{\sum g} \qquad and\ \ w_{i,j} = \frac{1}{2} \tag{5-7}$$

(3)

$$\hat{x}_{i,j} = \sum_k \sum_l w_{k,l}\ \ x_{i+k,j+l} \tag{5-8}$$

5.3.1.2. Scheme of Nagao and Matsuyama

Nagao and Matsuyama [51] proposed an algorithm which selects the most homogeneous neighborhood and replaces the pixel by its neighborhood average. They created nine overlapped subregions in a 5 × 5 window. The means and variances of the nine subregions are computed, and the center pixel is replaced by the mean of the subregion having the minimum variance.

5.3.1.3. Median Filter

A 3 × 3 window is used, and the median of the nine pixels in the window represents the smoothed pixel. A large window will smear details and edges, not to mention the higher computational load.

5.3.2. Advantages of the Sigma Filter

The gradient inverse filter is apparently the least efficient smoothing algorithm due to its small mask and the nature of its weighting scheme. The sigma filter is significantly superior in smoothing noise with a reduction of standard deviation by approximately a factor of ten. The Nagao and median filters are comparable in their ability to reduce noise.

5.3.3. Disadvantages of the Sigma Filter

The filter strength is good at the homogenous areas, while at the edges the data variation is greater than the homogenous areas stander deviation and so week filtration at the edges. Also, this method is quite time consuming. For the areas that have low frequency components, the selection of the filter windows size must be greater than that of the homogenous areas. That leads to selecting greater window size, and so more computation time.

5.4. Proposed Enhanced Lee Sigma Filter

We propose two major enhancements for the sigma filter. The first one intends to make the kernel window size expandable according to certain criteria, the second proposal to use the Perona-Malek anisotropic diffusion filter as the local smoother. These two proposals will be discussed in details in the next sections.

5.4.1. Expandable windows kernel

This is the proposed enhancement for the Lee filter. Use the filter with the minimum kernel size, in the case that the number of pixels shares in the kernel lower than a pre-specified level then expand the window kernel by one pixel in all kernel direction and recalculate the filter again. Repeat this process tell get the minimum required threshold of the pixels shared in the filter smoothing.

5.4.2. Anisotropic Diffusion Operator (Perona-Malik)

Anisotropic diffusion is an efficient nonlinear technique for simultaneously performing contrast enhancement and noise reduction. It smoothes homogeneous image regions and retains image edges. The main concept of anisotropic diffusion is the introduction of a function that inhabits smoothing at the image edges. This function is called diffusion coefficient. The diffusion coefficient is chosen to vary spatially in such a way to encourage intraregional smoothing in preference to inter-region smoothing [52]. Perona and Malik proposed the following nonlinear partial differential equation to smooth image on a continuous domain:

$$\frac{\partial I}{\partial t} = div[c(|\nabla I|) \,.\, \nabla I] \qquad (5\text{-}9)$$

$$I(t=0) = I_0.$$

Here, ∇ is the gradient operator, div is the divergence operator, | is the magnitude, c(x) is the diffusion coefficient, and I_0 is the initial image. For c(x), they have two coefficients options:

$$c(x) = \frac{1}{1 + (x/k)^2} \qquad (5\text{-}10)$$

$$c(x) = exp\left[-(x/k)^2\right], \qquad (5\text{-}11)$$

where k is the edge magnitude parameter. Physically, this model is like the thermo conduction. c(x) is the conduct coefficient along four directions. In practical design, the diffusion coefficient c(∇I) is anisotropic, and thus it is called anisotropic diffusion. The option 1 of the diffusion coefficient favors high contrast edges over low contrast ones. The option 2 of the diffusion coefficient favors wide regions over smaller ones. The edge magnitude parameter k controls conduction as a function of gradient. If k is low, then small intensity gradients are able to block conduction and hence diffusion across step edges. A large value of k can overcome the small intensity gradient barrels and reduces the influence of intensity gradients on conduction. Usually k ~ [20,100]. This method can be iteratively applied to the output image, and the iteration equation is:

$$I^{(n+1)} = I^{(n)} + \lambda \times \begin{bmatrix} c\left(\left|\nabla_{North} I^{(n)}\right|\right) \cdot \nabla_{North} I^{(n)} + c\left(\left|\nabla_{East} I^{(n)}\right|\right) \cdot \nabla_{East} I^{(n)} \\ + c\left(\left|\nabla_{West} I^{(n)}\right|\right) \cdot \nabla_{West} I^{(n)} + c\left(\left|\nabla_{South} I^{(n)}\right|\right) \cdot \nabla_{South} I^{(n)} \end{bmatrix}, \qquad (5\text{-}12)$$

where I(n) is the output image after n iterations. λ is the diffusion conducting speed, usually we set λ<=0.25.

5.5. Enhanced sigma filter procedure

First calculate the estimation of the center pixel using Perona-Malek operator, then start by kernel window size equal the minimum size, and expand the kernel size just in case low number of pixels shared in the Sum values, increase the windows size by two tell a Prespecified threshold of maximum window size.

1- Use Perona-Malek to calculate the center pixel $\hat{x}_{i,j}$ value using 3x3 window .
2- Establish an intensity range ($\hat{x}_{i,j} + \Delta$, $\hat{x}_{i,j} - \Delta$) *where* $\Delta = 2\sigma$.
3- Sum all Pixels which lie within the intensity range in a ($2n + 1$, $2m + 1$) window.
4- If the number of pixels shared in the pixels sum is less than the original window pixels size, increase the window size by two and go to step three. Else go to step 5.
5- Compute the average by dividing the sum by the number of pixels in the sum.

Or, Mathematically, let

$$\delta_{k,l} = 1 \qquad if\ (\hat{x}_{i,j} - \Delta) \leq x_{k,l} \leq (\hat{x}_{i,j} + \Delta)$$
$$= 0 \qquad \text{otherwise.}$$

Then

$$\hat{x}_{i,j} = \sum_{k=i-n}^{n+i} \sum_{l=j-m}^{m+j} \delta_{k,l}\ x_{k,l} \Big/ \sum_{k=i-n}^{n+i} \sum_{l=j-m}^{m+j} \delta_{k,l}$$

$$\hat{x}_{i,j} = \text{two-sigma average,} \quad \text{if } M > K$$

5.6. **Enhanced Sigma Filter Advantages**

The main disadvantage of the sigma filter is the week filtration at the edges, so making the window size expandable greatly solved this problem. Expanding the window in all direction make the filter work as the isotropic filter. For the local pixel estimation, Perona-Malek showed one of the most filters that enhance the SNR while keeping the edges of the image [53]. The Performance of the proposed filter depends on the details of the image, therefore for large areas of homogeneity the filter speed is greatly better that Lee filter while for less homogeneity areas the calculation will be greater the Lee filter while the filtration is better at the edges.

5.7. **Results and Discussion**

Quality of an image is a characteristic of an image that best measures the perceived image degradation. Digital images are subjected to wide variety of distortion during various processing, right from acquisition to processing. When it comes to image quality assessment there are two types of assessment: 1) Subjective Image Quality Assessment, 2) Objective Image Quality Assessment.

Subjective Image Quality Assessment is concerned with how image is perceived by a viewer and gives his or her opinion on a particular image [54]. Human eyes are the ultimate Viewer of an image. However this method of assessment is time consuming. The main goal of the image pre-processing is to get the most suitable image for the volume render, specially the surface rendering processing which need to minimize the image noise. So the first goal is to get a speckle free image. And the second to add some blurriness of the image at a specific processing stage in the 4D volume render, especially in the surface render stage to get rid of the wood-grain artifacts problem. From the previous sections of the Enhanced Sigma filter increasing the sigma value of the filter will increase the blurriness of the filtered image. Figure 46 shows the effect of increasing the Sigma value of the filter on the amount of blurriness added to the images. Figure 48 to Figure 50 Show the Enhanced sigma filter with different kernel window size. The speckle noise is greatly reduced with increasing the windows size. Also most of the image details and edges are preserved.

The expandable window enhances the filtration around the edges. Table 7 to Table 9 show the percentage of the pixels for each image that needs to be expanded, the first column is percentage of pixels that does not need to expand the window, the second column is the percentage of the pixels that expand the window by two and so on. The more details and edges in the image increase the need to expand the kernel window.

The expandable window shows good smoothing results at the edges, especially for low variance noise image. The expandable window reduced the need to start the kernel windows size by the worst case window size. Here we can start the kernel size with the minimum required window size and the window will expand in the case of edges. Figure 51 to Figure 53 show the comparison of the smoothing of the images with and without expandable window with deferent kernel window size.

The local Pixel estimation using median filter showed higher SNR Figure 48, Losing one pixel details and one pixel smearing is found, also blurring of the image is observed. While the local estimation using Perona-Malek operator showed lower SNR Figure 47, but the image perspective quality is better, blurriness is at minimum and no pixel smearing is observed. Due to these facts, we have used the Perona-Malek local pixel estimation for the 2D ultrasound image views like the MPR while we have used the median filter for the surface and volume rendered images.

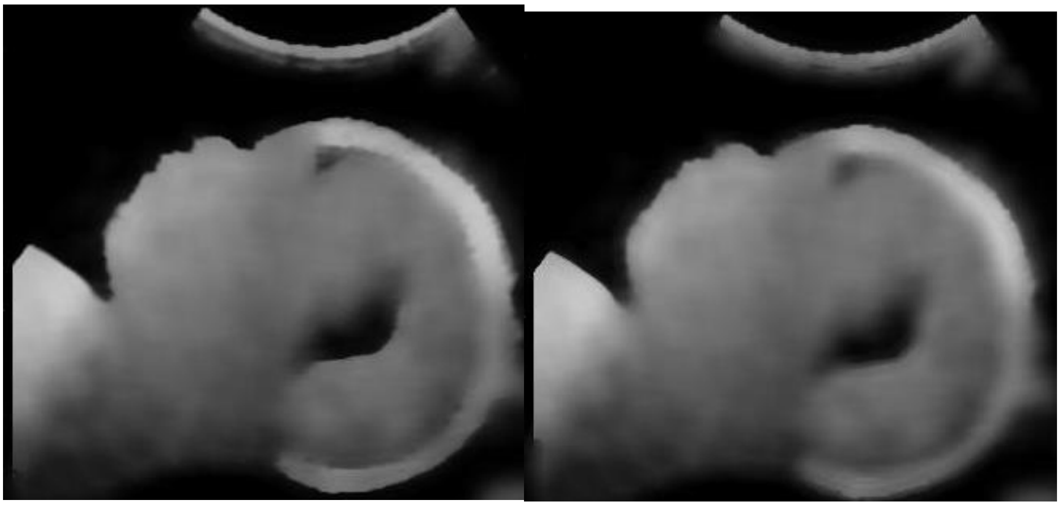


Figure 46. Effect of sigma value, Left image with Sigma = 0.1, The right image with sigma = 0.17, Local pixel calculated using median filter

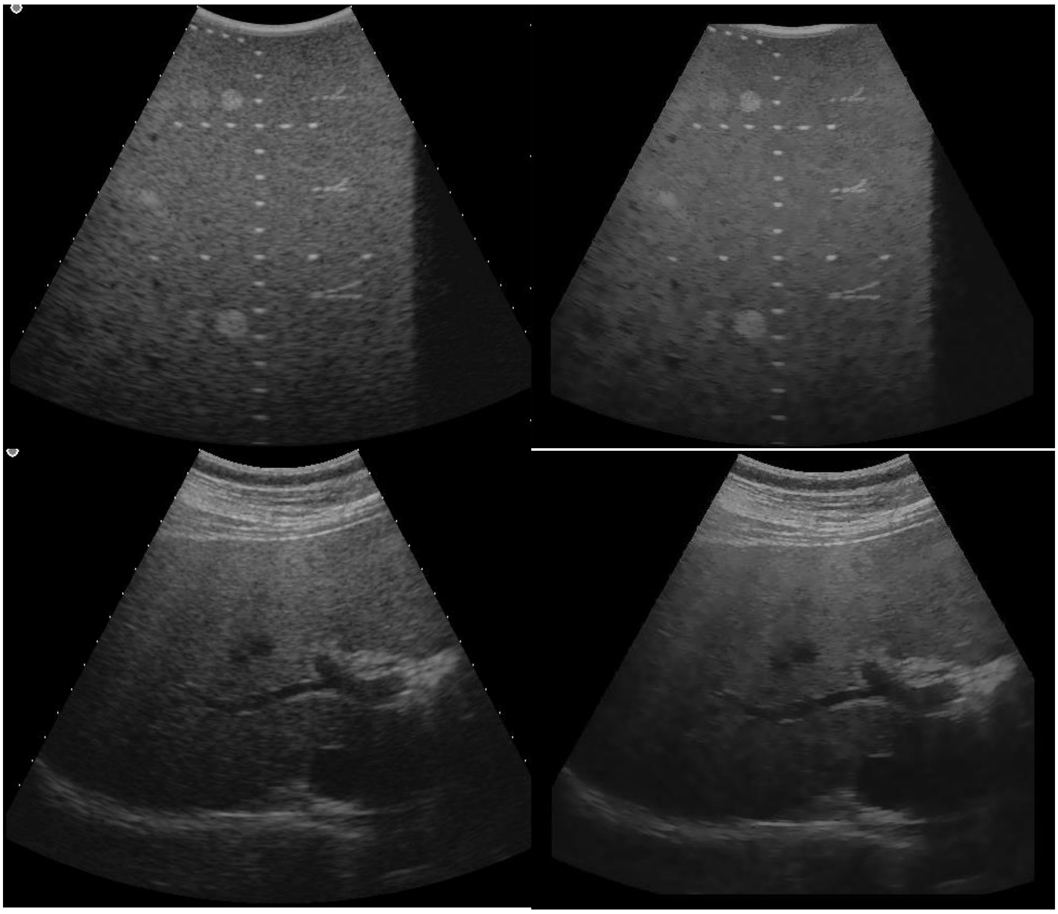


Figure 47. Enhanced sigma with Perona-Malek local pixel calculation, Left is the original image right is the processed image, with sigma = 0.03, window size 7×7.

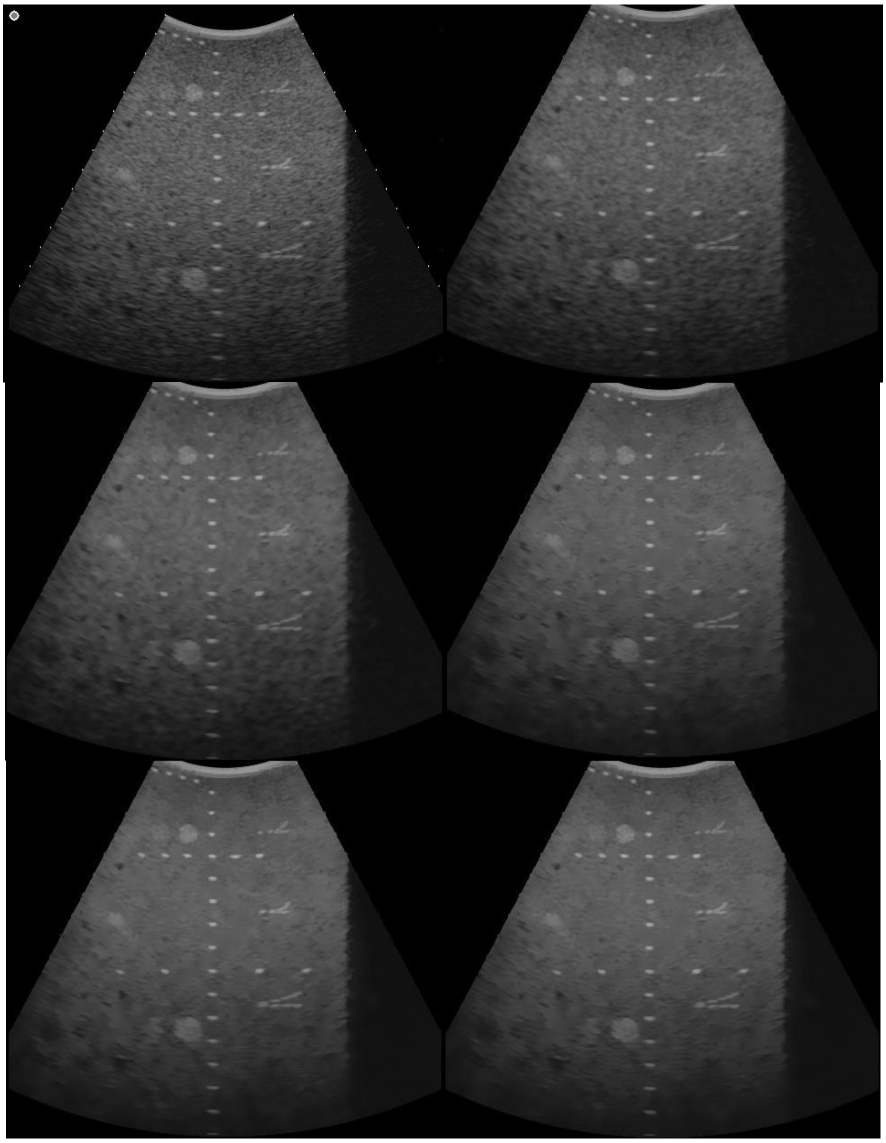


Figure 48. Deferent enhanced sigma filter with deferent windows size. Top left is the original image, and from left to right and from top to bottom and increasing the filter from 3×3 to 11×11. The Sigma value for the filter is constant = 0.03. The images for phantom CIRS Model040GSE)

Table 7. Shows the calculation for each kernel window size, the first column shows the percentage of calculation done without the need to expand the kernel window size and the next column is the increase of the kernel window size by two and so on.

	+0	+2	+4	+6	+8	+10	+12	+14
3x3	79.30	20.51	0.04	0.01	0.03	0.03	0.03	0.05
5x5	50.19	47.33	0.82	0.29	0.16	0.09	0.05	1.07
7x7	37.02	57.88	2.20	0.57	0.25	0.22	1.85	0.00
9x9	30.86	60.64	3.97	1.39	0.61	0.36	2.16	0.00
11x11	27.37	59.51	6.50	2.10	1.16	3.36	0.00	0.00
13x13	24.88	56.81	9.67	2.14	1.64	4.86	0.00	0.00

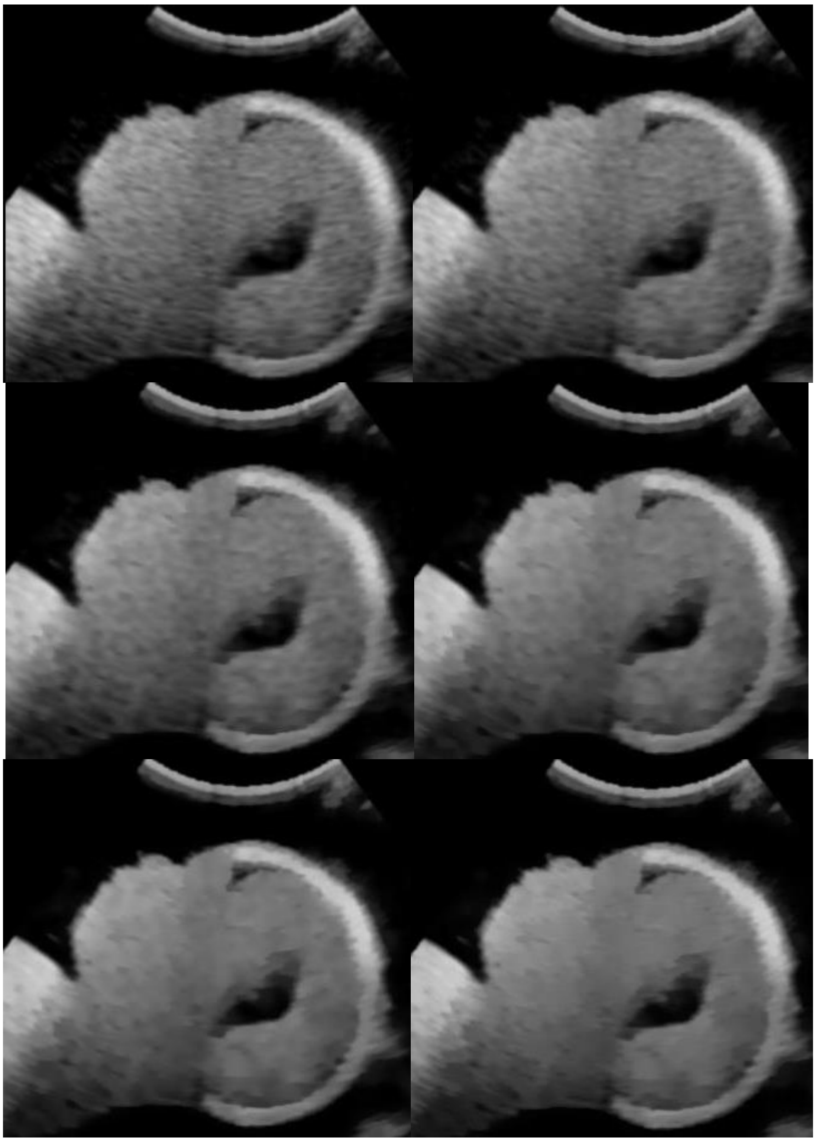


Figure 49. Deferent enhanced sigma filter with deferent windows size. Top left is the original image, and from left to right and from top to bottom and increasing the filter from 3×3 to 11×11. The sigma value for the filter is constant = 0.03. the images for phantom (CIRS Model 068-21week)

Table 8. Shows the calculation for each kernel window size, the first column shows the percentage of calculation done without the need to expand the kernel window size and the next column is the increase of the kernel window size by two and so on.

	+0	+2	+4	+6	+8	+10	+12	+14
3x3	90.30	8.10	1.28	0.22	0.03	0.05	0.00	0.01
5x5	72.17	21.20	2.77	1.47	1.12	0.84	0.27	0.18
7x7	56.03	28.51	3.16	3.16	2.15	2.32	1.44	3.22
9x9	48.52	32.03	5.20	2.85	2.24	1.87	7.28	0.00
11x11	42.92	32.43	6.52	3.87	3.26	11.00	0.00	0.00
13x13	38.55	30.99	7.95	4.37	3.53	14.62	0.00	0.00

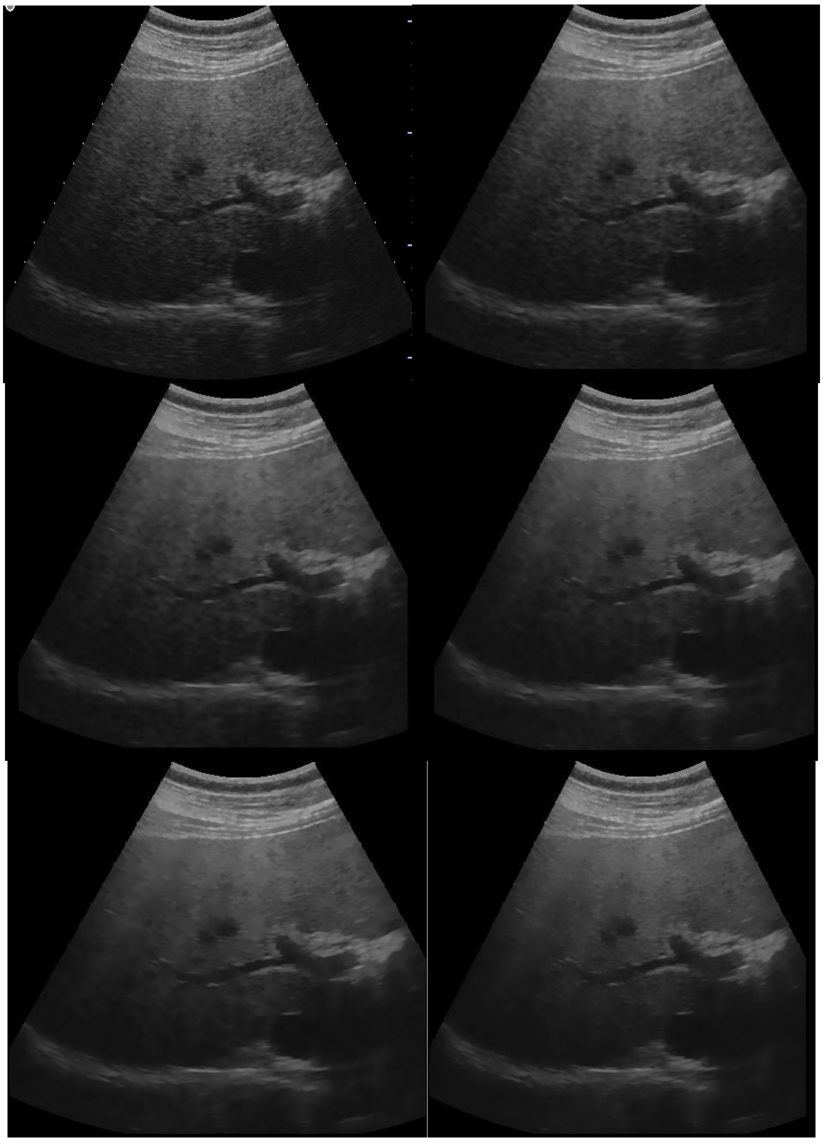


Figure 50. Deferent enhanced sigma filter with deferent windows size. Top left is the original image, and from left to right and from top to bottom and increasing the filter from 3×3 to 11×11. The sigma value for the filter is constant = 0.03. The images for lever

Table 9. Shows the calculation for each kernel window size, the first column shows the percentage of calculation done without the need to expand the kernel window size and the next column is the increase of the kernel window size by two and so on.

	+0	+2	+4	+6	+8	+10	+12	+14
3x3	77.95	21.83	0.20	0.01	0.00	0.00	0.00	0.01
5x5	53.79	42.68	2.28	0.51	0.17	0.08	0.29	0.19
7x7	41.51	50.11	4.91	1.82	0.76	0.30	0.58	0.00
9x9	34.11	50.64	8.30	3.11	1.46	0.87	1.50	0.00
11x11	29.27	48.67	10.83	4.58	2.64	4.00	0.00	0.00
13x13	24.00	43.27	11.56	11.56	3.22	6.38	0.00	0.00

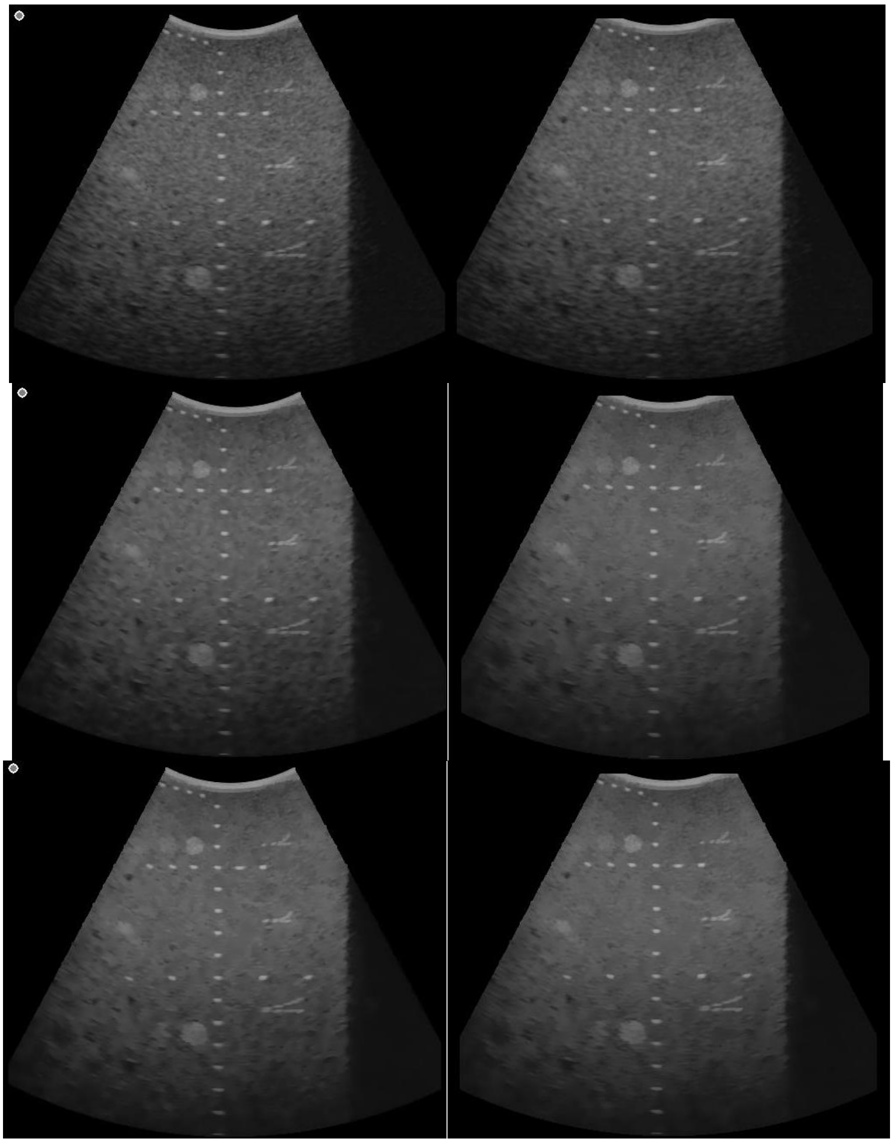


Figure 51. Fixed and expandable filter kernel window size. Left images with fixed kernel and the right images with expandable window size. From top to bottom the kernel size are 3×3, 7×7 and 11×11. Phantom image (CIRS Model 040GSE)

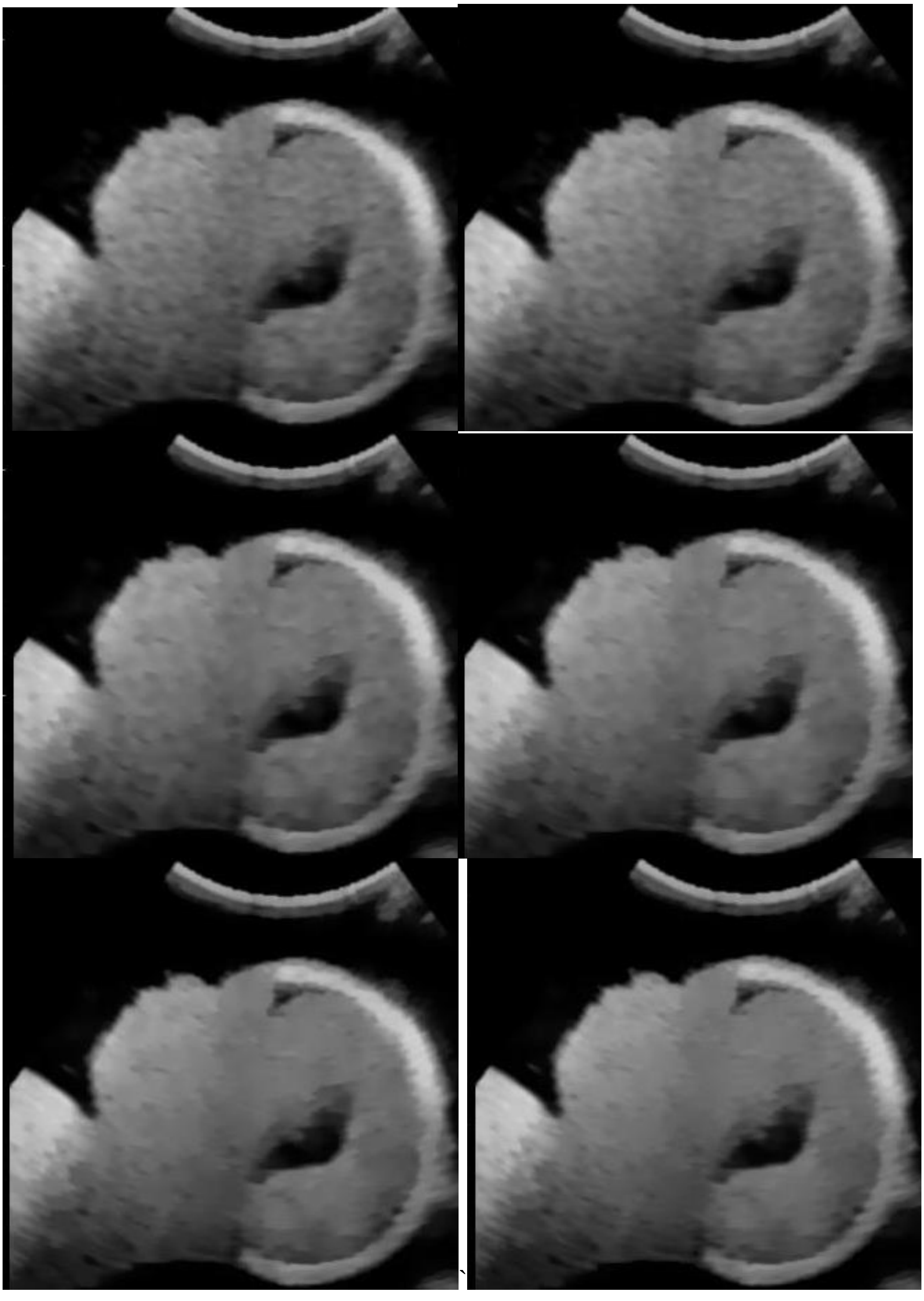


Figure 52. Fixed and expandable filter kernel window size. Left images with fixed kernel and the right images with expandable window size. From top to bottom the kernel size are 5×5, 9×9 and 15×15. Phantom Image (CIRS Model 068-21week)

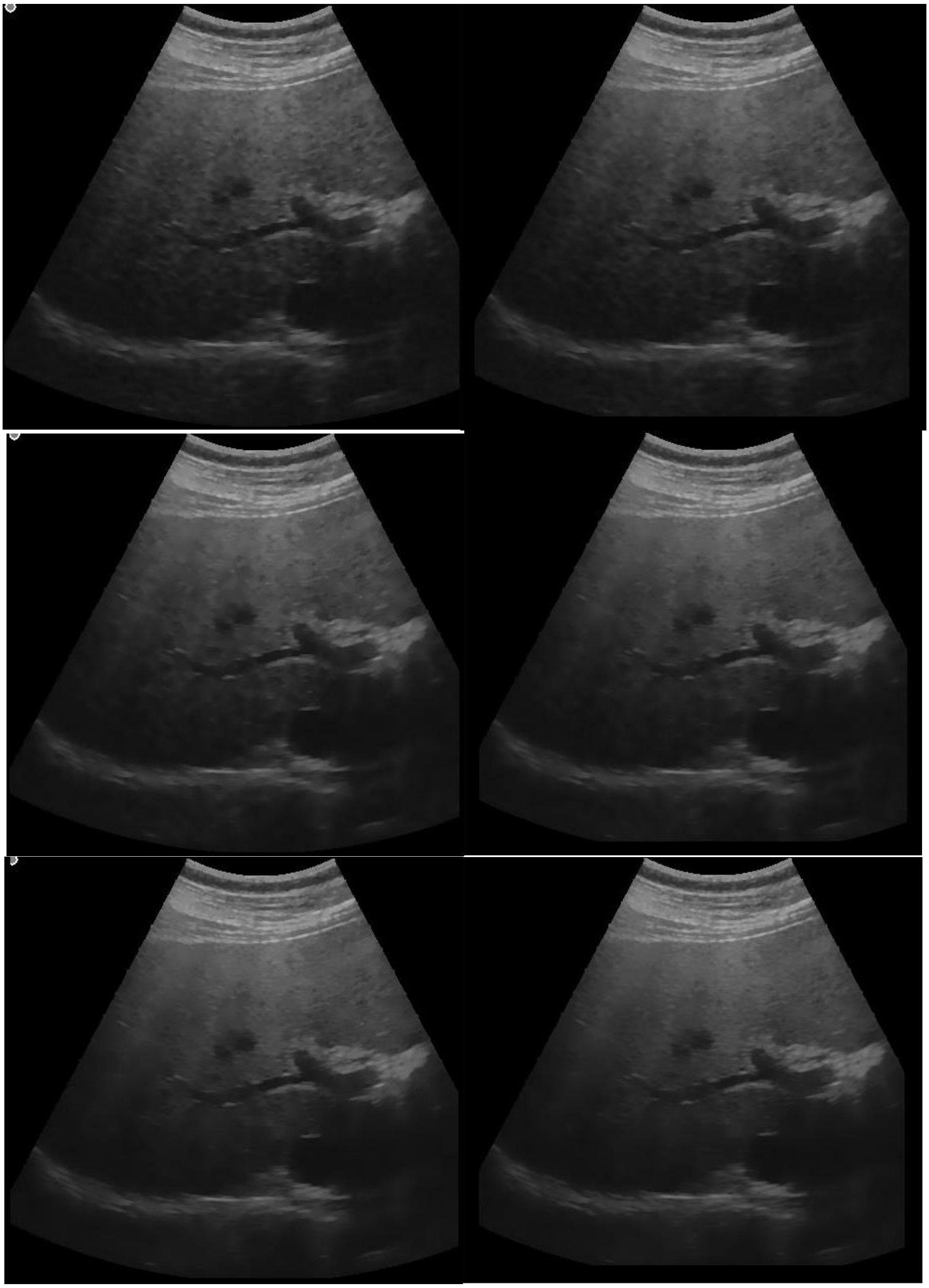


Figure 53. Fixed and expandable filter kernel window size. Left images without fixed kernel and the right images with expandable window size. From top to bottom the kernel size are 5×5, 9×9 and 15×15. Liver image.

Chapter 6 : Conclusion and Future Work

In this work, we presented a high-performance rendering pipeline implemented on a customized platform made up of low-cost commercial components. This approach combines the optimization methods used in offline 3D visualization and real-time processing on GPUs to offer an optimized platform for 4D imaging applications. The processing system is verified by rendering volumes acquired on a 4D commercial ultrasound imaging system with a research interface with results showing diagnostic quality while maintaining real-time performance suggesting potential for practical utility.

A new 2D filter is proposed, the new filter is optimized for the 4D rendering pipeline. The new filter is used in three stages of the rendering pipeline and for each stags the filter showed a good and suitable results for the 4D rendering problem. Due to the good performance and image quality of the proposed filter we tested it for regular ultrasound images and the subjective study showed a good image quality and enhanced the SNR of the result image.

Surface detection and surface segmentation is one of the challenges of the surface rendering, so a more segmentation techniques can be tested for better surface rendering. Also more enhancements can be done in the Raycast step size, in our implementation we used the worst case step value for the raycasting, while due to the convexity shape of the 4D probe and 2D probe the sampling rate can be reduced according to the image depth.

As higher performance GPU technology becomes available, advanced memory textures and throughput rate are expected to grow significantly. Here, we used a mid-range graphics card to confirm that the current implementation does not significantly affect the raycasting performance. The two major specifications of the graphics card that affect performance are the core and memory specifications. For the core, the number of cores used and the texture filling rate are the most important feature. On the other hand, the performance of the memory part is determined by its bandwidth.

References

[1] T. R. Nelson and D. H. Pretorius, "Three-dimensional ultrasound of fetal surface features," *Ultrasound in Obstetrics and Gynecology,* vol. 2, no. 3, pp. 166-174, 2003.

[2] A. Gee, R. Prager, G. Treece and L. Berman, "Engineering a freehand 3D ultrasound system," *Pattern Recognition Letters,* vol. 24, no. 4-5, pp. 757-777, 2003.

[3] F. Dunny, "Ultrasound," *Education, IEEE Transactions on,* pp. 266-268, 1991.

[4] . J. P. McGahan and B. B. Goldberg , Diagnostic Ultrasound, Second Edition, CRC Press, 2008.

[5] J. Carr, *Surface Reconstruction in 3D Medical Imaging,* University of Canterbury, 1996.

[6] S. N. Narouze, Atlas of Ultrasound-Guided Procedures in Interventional Pain Management, New York: Springer, 2011.

[7] T. L. Szabo, Diagnostic ultrasound imaging : inside out, Burlington, MA: Elsevier Academic Press, 2004.

[8] P. Hoskins, K. Martin and A. Thrush, Diagnostic ultrasound: physics and equipment, New York: Cambridge University Press, 2010.

[9] K. K. Shung, Diagnostic Ultrasound : imaging and blood flow measurements, Boca Raton: CRC / Taylor & Francis Group, 2006.

[10] P. N. T. Wells, "Ultrasonics in medicine and biology," *Physics in Medicine and Biology,* vol. 22, no. 4, p. 629, 1977.

[11] J. W. Goodman, "Some fundamental properties of speckle," *Journal of the Optical Society*, vol. 66, no. 11, p. 1145–1150, 1976.

[12] J.-S. Lee and E. Pottier, Polarimetric Radar Imaging From Basics to Applications, NW: CRC Press, 2009.

[13] M. A. H. Khan, *3D Reconstruction of Ultrasound Images*, University of Burgundy, University of Girona and University of Heriot Watt, 2008.

[14] P. Hoskins, A. Thrush and K. Martin, Diagnostic ultrasound: physics and equipment, second ed., Cambrige University Press, 2010.

[15] A. Fenste, B. D. Downey and H. N. Cardinal, "Three-dimensional ultrasound imaging," *Physics In Medicine And Bioglogy,* vol. 46, no. 5, 6 11 2000.

[16] M. Hadwiger, J. M. Kniss, C. Rezk-salama, D. Weiskopf and K. Engel, Real-Time Volume Graphics, Eurographics Association, 2006.

[17] B. Preim and D. Bartz, Visualization in Medicine: Theory, Algorithms, and Applications, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007.

[18] J. Kruger and R. Westermann, "Acceleration Techniques for GPU-based Volume Rendering," in *Visualization, 2003. VIS 2003. IEEE,* 2003, pp. 287-292.

[19] M. Levoy, "Display of surfaces from volume data," *IEEE Computer Graphics and Applications*, vol. 8, no. 3, pp. 29-37, 1988.

[20] G. P. Lacroute, "Fast volume rendering using a shear-warp factorization of the

viewing transformation," Computer Systems Laboratory, Stanford University, 1995.

[21] *Splatting: A Parallel Feed-Forward Volume Rendering Algorithm,* Ph.D. Dissertation, Department of Computer Science, The University of North Carolina, 1991.

[22] Z. Matthias, H. Pfister, J. v. Baar and M. Gross, "EWA Splatting," *IEEE transactions on visualization and computer graphics,* vol. 8, no. 3, pp. 223-238, 2002.

[23] G. Coppini, R. Poli and G. Valli, "Recovery of the 3-D shape of the left ventricle from echocardiographic images," *Medical Imaging, IEEE Transactions on,* vol. 14, no. 2, pp. 301-317, 1995.

[24] J. Gill, L. O. C. John P. Robarts Res. Inst., H. Ladak, D. Steinman and A. Fenster, *Engineering in Medicine and Biology Society, 2000. Proceedings of the 22nd Annual International Conference of the IEEE,* vol. 1, pp. 669-672, 2000.

[25] C. Sohn, "A new diagnostic technique. Three-dimensional ultrasound imaging," *Ultrasonies Int 89 Conference Proc. Butterworths,* pp. 1148 -1153, 1989.

[26] R. Robb, "Virtual endoscopy: development and evaluation using the visible human data sets," *Computerized Medical Imaging and Graphics,* vol. 24, no. 3, pp. 133-151, 1995.

[27] R. Fernando and M. J. Kilgard, The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics, Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003.

[28] R. A. Drebin, L. Carpenter and P. Hanrahan, "Volume rendering," *Computer Graphics,* vol. 22, no. 4, pp. 51-58, 1988.

[29] R. A. Reynolds and D. Gordon, "Image space shading of 3-dimentional objects," *Computer Vision, Graphics, And Image Processing,* vol. 29, pp. 361-376, 1985.

[30] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *Computer Graphics,* vol. 21, no. 4, pp. 163-169, 1987.

[31] K. H. Hohne and R. Bernstein, "Shading 3D-Images from CT Using Gray-Level Gradients," *IEEE Transactions on Medical Imaging,* vol. 5, no. 1, pp. 45-47, 1986.

[32] J. K. Udupa and G. T. Herman, 3D Imaging in Medicine, second edition, CRC PRESS, 2000.

[33] T. R. Nelson and T. T. Elvins, "Visualization of 3D ultrasound data," *IEEE Computer Graphics and Applications,* vol. 13, no. 6, pp. 50-57, 1993.

[34] R. Fattal and D. Lischinski, "Variational Classification for Visualization of 3D Ultrasound Data," in *VIS '01 Proceedings of the conference on Visualization '01*, Washington, DC, USA, 2001.

[35] D. K. Pai, R. Rohling and Y. Zhang, "Direct surface extraction from 3d freehand ultrasound images," *IEEE Visualization,* vol. 30, no. 11, pp. 1461-1473, 2002.

[36] L. Sukhyun, K. Kwon and B.-S. Shin, "GPU-based interactive visualization framework for ultrasound datasets," *Comput. Animat. Virtual Worlds,* vol. 20, no. 1, pp. 11-23, 2009.

[37] A. F. Elnokrashy, A. A. Elmalky, T. M. Hosny, M. A. Ellah, A. Megawer, A. Elsebai, A.-B. Youssef and Y. Kadah, "GPU-based reconstruction and display for 4D ultrasound data," in *Ultrasonics Symposium (IUS), 2009 IEEE International*, Rome, 2009.

[38] D. Shreiner, OpenGL Reference Manual: The Official Reference Document to OpenGL, Version 1.4, Boston: Addison-Wesley, 2004.

[39] "developer.nvidia," NVidia, 2012. [Online]. Available: http://developer.nvidia.com/object/nvidia_opengl_specs.html. [Accessed 1 10 2013].

[40] C. D. Hansen and C. R. Johnson, "Isosurfaces and Level-Sets," in *Visualization Handbook*, Elsevier, 2005, pp. 97-123.

[41] J. F. Blinn, "Models of Light Reflection for Computer Synthesized Pictures," *Computer Graphics,* vol. 11, no. 2, pp. 192-198, 1977.

[42] J. F. Blinn, "Compositing, Part I: Theory," *IEEE Computer Graphics and Application*, vol. 14, no. 5, 1994.

[43] A. Elnokrashy, M. Hassan, T. Hosny, A. Ali, A. Megawer and Y. Kadah, "Multipass GPU surface rendering in 4D ultrasound," in *Cairo International Biomedical Engineering Conference (CIBEC)*, Cairo, 2012.

[44] C. P. Loizou and C. S. Pattichis, Despeckle Filtering of Ultrasound Images, Springer New York, 2011, pp. 1-166.

[45] R. F. Wagner, S. W. Smith and J. M. Sandrik, "Statistics of speckle in ultrasound B-scans," *IEEE Trans. Sonics and Ultrasonics,* vol. 30, p. 156–163, 1983.

[46] B. Deka and P. Bora, "Wavelet-based Despeckling of Medical Ultrasound Images," *IETE Journal of Research,* vol. 59, no. 2, pp. 97-108, 2013.

[47] K. Z. AbdElmoniem, Y. M. Kadah and A. M. Youssef, "Real time adaptive ultrasound speckle reduction and coherence enhancement," *Image Processing, 2000. Proceedings. 2000 International Conference on,* vol. 1, pp. 1106 -1109, 2000.

[48] J.-S. Lee, "Digital image smoothing and the sigma filter," *Computer Vision, Graphics, and Image Processing,* vol. 24, no. 2, pp. 255-269, 1983.

[49] A. Lev, S. W. Zucker and A. Rosenfeld, "Interactive enhancement of noisy images," *IEEE Trans. On Systems, Man and Cybernetics,* vol. 7, no. 6, pp. 435-422, 1977.

[50] D. C. Wang, A. H. Vagnucci and C. C. Li, "Image enhancement by gradient inverse weighted smoothing scheme," vol. 4, no. 16, pp. 9-16, 1979.

[51] M. Nagao and T. Matsuyama, "Edge preserving smoothing," in *Proc. of 4th International Joint Conference on Pattern Recognition*, 1979.

[52] Perona, Pietro and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions, Pattern Analysis and Machine Intelligence,* vol. 12, no. 7, pp. 629-639, 1990.

[53] G. Farage, S. Foucher and G. Benie, "Comparison of PolSAR Speckle Filtering Techniques," in *Geoscience and Remote Sensing Symposium, 2006. IGARSS 2006. IEEE International Conference on,* 2006.

[54] M. Christian, M. o. Petersen and M. B. Nielsen, "Ultrasound Image Quality Assessment, a Framework for Evaluation of Clinical Image Quality," vol. 7629, pp. 7629-76290, 2010.

[55] J. Kniss, C. Hansen, P. Shirley and A. Mcpherson, "A model for volume lighting and modeling," *IEEE Transactions on Visualization and Computer Graphics,* vol. 9, pp. 150-162, 2003.

[56] K. H. Hoehne and R. Bernstein, "Shading 3D-Images from CT Using Gray-Level

Gradients," *IEEE Transactions on Medical Imaging,* vol. 5, no. 1, pp. 45-47, 1986.

[57] S. K. Narayanan and R. S. Wahidabanu, "A View on Despeckling in Ultrasound Imaging," *International Journal of Signal Processing, Image Processing and Pattern Recognition,* vol. 2, no. 3, pp. 85-98, 1990.

[58] Kniss, Joe, G. Kindlmann and C. Hansen, "Multidimensional transfer functions for interactive volume rendering," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 8, no. 3, pp. 270-285, 2002.

[59] W. Li, K. Mueller and A. Kaufman, "Empty space skipping and occlusion clipping for texture-based volume rendering," in *Proceedings of the 14th IEEE Visualization 2003 (VIS'03),* Washington, DC, USA, IEEE Computer Society, 2003, pp. 42--.

[60] S. M. Ali and R. E. Burge, "New automatic techniques for smoothing and segmenting SAR images," *Signal Processing,* vol. 14, no. 4, pp. 335-346, 1988.

[61] Cg 3.1 Refrence Manual Release 3.1, NVIDIA Corporation, 2012.

[62] Å. Birkeland, V. ,. H. D. Solteszova, O. G. Helge, S. Brekke, T. Ropinski and I. Viola, The Ultrasound Visualization Pipeline - A Survey, arXiv preprint arXiv:1206.3975, 2012.

[63] L. Grønvold, *Implementing ultrasound beamforming on the GPU using CUDA,* Norwegian University of Science and Technology, 2008.

[64] E. Brunner, "Ultrasound System Considerations and their Impact on Front-End Components," 2002. [Online]. Available: http://www.analog.com/library/analogdialogue/archives/36-03/ultrasound/ultrasound.pdf.

[65] M. Grandolfo and P. Vecchia, Fundamentals of Acoustic Wave Theory, Springer US, 1987.

[66] Lorensen, E. William, Cline and E. Harvey, "Marching cubes: A high resolution 3D surface construction algorithm," *Computer Graphics,* vol. 21, no. 4, pp. 163-169, 1987.

[67] G. Sakas and S. Walter, "Extracting surfaces from fuzzy 3D ultrasound data," *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques,* vol. 701, no. 4, pp. 465-474, 1995.

[68] Kruger, Jens and R. Westermann, "Acceleration techniques for GPU-based volume rendering," in *In Proceedings of the 14th IEEE Visualization 2003 (VIS'03),* Washington, DC, USA, 2003.

[69] Klein, Thomas and M. Strengert, "Exploiting frame-to-frame coherence for accelerating high-quality volume raycasting on graphics hardware," in *Procedding Of IEEE Visualizaion '05,* IEEE, 2005, pp. 223-230.

[70] Y. Yu and S. Acton, "Speckle Reducing Anisotropic Diffusion," *IEEE Transactions on image porcessing,* vol. 11, no. 11, pp. 1260-1270, 2002.

[71] K. Engel, M. Kraus and T. Ertl, "High-quality pre-integrated volume rendering using hardware-accelerated pixel shading," in *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, Los Angeles, California, USA, ACM, 2001, pp. 9-16.

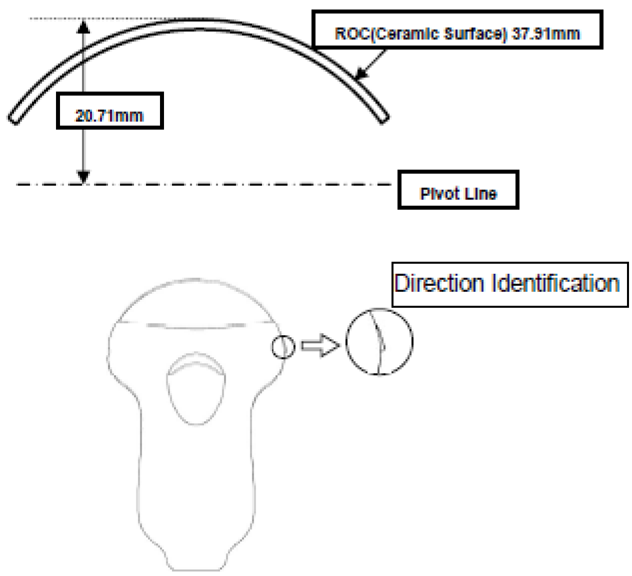
Appendix I 4D Probe Specification

Table 10. 4D Probe general specifications

Array Geometry Type	**Curved Linear Array**		
Nominal Center Frequency		4.5	MHz
Pitch	Front of PZT@ Patient side	0.405	mm
Number of Elements		128	ea
Radius of Curvature	Front of PZT@ Patient side	37.91	mm
Elevation Aperture		12.5	mm
Elevation Focus		55	mm
Field of View		77.86	deg
Radius of Acoustic Lens(Concave type)		25	mm
Radius of Mechanical Scanning		20.71 ,24.3(cap surface)	mm
Ztt(Ltt)		1.6	mm

Table 11. 4D Probe mechanical specification

Parameters	Description	
Weight	Total weight not to exceed TBD g (including cable). Weight of the probe head (excluding cable and connector): ≤ 240 g	
Reliability	4 volumes/sec at volume angle 25°, 10 million cycles.	
Motor	Two-phase Step Motor(Bipolar)	Step motor (200step/rev)
		Winding resistance 8.5Ω/Phase
		Inductance 3.4mH/Phase
		Rated Voltage：3V/Phase
		Rated Current：0.35A/Phase
Transducer Drive Deceleration Ration	Variable (0.579 ~ 1.428) * Refer to APPENDIX	
Wobbling Parameter	Wobbling Angle	Max. 86°
	Maximum Frame Rate	8 frame/sec at 60° 25 frame/sec at 10°
Positional Accuracy	1. Encoded relative position accurate within +/- 0.2 degree 2. Encoded absolute position accurate within +/- 1.40 degree 3. Encoded center position accurate within +/- 0.5 degree	
Direction Identification		



الملخص

إن التصوير باستخدام الموجات الصوتية رباعية الأبعاد يضيف الى قدرة الموجات الصوتية فى التصوير المقطعي اللحظى قدرة التصوير المجسم اللحظى أيضا و هو الذى يتم تحقيقه بواسطة المصور الطبى. هناك عدة طرق لعرض المجسمات الطبية. يعتبر العرض السطحى لمجسمات أعضاء الجسم البشرى من الطرق المعروفة و المهمة لعرض البيانات الحجمية مثل تطبيقات الولادة و طب النساء و الأجنة. هناك عدة تحديات فى هذا الأسلوب من العرض للصور الطبية و على رأسها هو تلوين السطح بطريقة تعطي الإحساس بالبعد الثالث للصورة. تضيف بعض المعلومات للطبيب مثل البعد عن السطح. بالإضافة لذلك تحديد موقع النقط السطحية للصور الطبية و هى من أكثر الأمور تعقيدا فى هذة الطريقة.

و نظرا لقلة جودة صور الموجات فوق الصوتية و ذلك بسبب وجود ترقيط مميز بالصورة, تم تطوير طريقة جديدة لتحسين و معالجة صور الموجات الفوق صوتية قبل أن تتم عملية استنتاج المجسمات ثلاثية الأبعاد من تلك الصورة. و نظرا لجودة هذه الطريقة تم تطبيقها على صور الموجات الصوتية ثنائية الأبعاد و اعطت نتائج جيدة سواء من حيث جودة الصورة و ايضا من حيث فاعلية التنفيذ و خاصة قدرتها على أن تتم مع التصوير اللحظى في الزمن الحقيقي.

أيضا تقدم الرسالة طريقة مختصرة و عالية الأداء لعرض و استنتاج الصور الطبية ثلاثية الأبعاد من البيانات الطبية ثلاثية الأبعاد. و ذلك من خلال استخدام أجهزة موجات صوتية متداولة و ذات أسعار رخيصة, أيضا استخدام وحدات معالجة رسوم متداولة و ذات قدرات متوسطة حتى تكون زهيدة التكلفة و ذلك لاثبات أن الطرق المقدمة فى البحث ذات كفاءة حسابية عالية و جودة فى عرض الصور الطبية أيضا. هذا و قد تم انجاز هذه الرسالة على جهاز تصوير بالموجات فوق الصوتية تم تطويره في مصر من فريق من الباحثين يضم مقدم هذه الرسالة و الجدير بالذكر أن التصوير رباعي الأبعاد على هذا المنتج المحلي قد استفاد من نتائج هذه الرسالة في أن يكون أقل أجهزة هذا النوع من التصوير تكلفة في حين يتميز بجودته التشخيصية مما يعطي أمل في أن يتم استعماله في تحسين الرعاية الصحية في المناطق الفقيرة و النائية في مصر و الدول النامية.

مهندس: أحمد فتحى مسعد النقراشى

تاريخ الميلاد: 1975\02\10

الجنسية: مصري

تاريخ التسجيل: 2010\10\1

تاريخ المنح: 2014\01\19

القسم: الهندسة الطبية الحيوية و المنظومات

الدرجة: الدكتوراه

المشرفون:

أ.د أبو بكر محمد يوسف (المشرف الرئيسي)

أ.د ياسر مصطفى قدح (المشرف الرئيسي)

الممتحنون:

أ. د أبو بكر محمد يوسف (المشرف الرئيسي)

أ. د ياسر مصطفى قدح (المشرف الرئيسي)

ا.د. ناهد حسين سلومة (الممتحن الداخلي)

أ.د محمد إبراهيم العدوي (الممتحن الخارجي)

عنوان الرسالة:

طرق جديدة لمعالجة الصورة الطبية

الكلمات الدالة:

التصوير الطبى رباعى الأبعاد , معالجة الصورة الطبية , معالجة صورة الموجات فوق الصوتية , تحسين صور الموجات فوق الصوتية

ملخص الرسالة:

إن التصوير باستخدام الموجات الصوتية رباعية الأبعاد يضيف الى قدرة الموجات الصوتية فى التصوير المقطعي اللحظى قدرة التصوير المجسم اللحظى أيضا و هو الذى يتم تحقيقه بواسطة المصور الطبى. هناك عدة طرق لعرض المجسمات الطبية. يعتبر العرض السطحى لمجسمات أعضاء الجسم البشرى من الطرق المعروفة و المهمة لعرض البيانات الحجمية مثل تطبيقات الولادة و طب النساء و الأجنة. هناك عدة تحديات فى هذا الأسلوب من العرض للصور الطبية و على رأسها هو تلوين السطح بطريقة تعطي الإحساس بالبعد الثالث للصورة. تضيف بعض المعلومات للطبيب مثل البعد عن السطح. بالإضافة لذلك تحديد موقع النقط السطحية للصور الطبية و هى من أكثر الأمور تعقيدا فى هذة الطريقة .

و نظرا لقلة جودة صور الموجات فوق الصوتية و ذلك بسبب وجود ترقيط مميز بالصورة, تم تطوير طريقة جديدة لتحسين و معالجة صور الموجات الفوق صوتية قبل أن تتم عملية استنتاج المجسمات ثلاثية الأبعاد من تلك الصورة. و نظرا لجودة هذه الطريقة تم تطبيقها على صور الموجات الصوتية ثنائية الأبعاد و اعطت نتائج جيدة سواء من حيث جودة الصورة و ايضا من حيث فاعلية التنفيذ و

وخاصة قدرتها على أن تتم مع التصوير اللحظى في الزمن الحقيقي.

أيضا تقدم الرسالة طريقة مختصرة و عالية الأداء لعرض و استنتاج الصور الطبية ثلاثية الأبعاد من البيانات الطبية ثلاثية الأبعاد. و ذلك من خلال استخدام أجهزة موجات صوتية متداولة و ذات أسعار رخيصة, أيضا استخدام وحدات معالجة رسوم متداولة و ذات قدرات متوسطة حتى تكون زهيدة التكلفة و ذلك لاثبات أن الطرق المقدمة فى البحث ذات كفاءة حسابية عالية و جودة فى عرض الصور الطبية أيضا. هذا و قد تم انجاز هذه الرسالة على جهاز تصوير بالموجات فوق الصوتية تم تطويره في مصر من فريق من الباحثين يضم مقدم هذه الرسالة و الجدير بالذكر أن التصوير رباعي الأبعاد على هذا المنتج المحلي قد استفاد من نتائج هذه الرسالة في أن يكون أقل أجهزة هذا النوع من التصوير تكلفة في حين يتميز بجودته التشخيصية مما يعطي أمل في أن يتم استعماله في تحسين الرعاية الصحية في المناطق الفقيرة و النائية في مصر و الدول النامية.

عنوان الرسالة
طرق جديدة لمعالجة الصور الطبية

اعداد
أحمد فتحى مسعد النقراشى

رسالة مقدمة إلى كلية الهندسة – جامعة القاهرة
كجزء من متطلبات الحصول على درجة الدكتوراه
في
الهندسة الطبية الحيوية والمنظومات

يعتمد من لجنة الممتحنين:

الاستاذ الدكتور: أبو بكر محمد يوسف المشرف الرئيسى

الاستاذ الدكتور: ياسر مصطفى قدح المشرف الرئيسى

الاستاذ الدكتور: ناهد حسين سلومه الممتحن الداخلي

الاستاذ الدكتور: محمد إبراهيم العدوي الممتحن الخارجي

كليــة الهندســة - جامعــة القاهــرة
الجيـزة - جمهوريـة مصـرالعربيــة

2014

عنوان الرسالة
طرق جديدة لمعالجة الصور الطبية

اعداد
أحمد فتحى مسعد النقراشى

رسالة مقدمة إلى كلية الهندسة – جامعة القاهرة
كجزء من متطلبات الحصول على درجة الدكتوراه
في
الهندسة الحيوية الطبية والمنظومات

تحت اشراف

ياسر مصطفى إبراهيم قدح
أستاذ بقسم الهندسة الحيوية الطبية
والمنظومات

أبو بكر محمد يوسف
أستاذ بقسم الهندسة الحيوية الطبية
والمنظومات

كليـة الهندسـة - جامعـة القاهـرة
الجيـزة - جمهوريـة مصرالعربيـة

2014

عنوان الرسالة
طرق جديدة لمعالجة الصور الطبية

اعداد

أحمد فتحى مسعد النقراشى

رسالة مقدمة إلى كلية الهندسة – جامعة القاهرة
كجزء من متطلبات الحصول على درجة الدكتوراه
في
الهندسة الحيوية الطبية والمنظومات

كليــةالهندســة-جامعــةالقاهــرة
الجيـزة-جمهوريـة مصـرالعربيــة

2014