

**THREE-DIMENSIONAL RECONSTRUCTION OF
ULTRASOUND FOETAL IMAGES USING VISUALISATION
TOOLKIT**

GRACE ANABELA HENRY DUSIM

**FACULTY OF ENGINEERING
UNIVERSITY OF MALAYA
KUALA LUMPUR**

2020

**THREE-DIMENSIONAL (3D) RECONSTRUCTION OF
ULTRASOUND FOETAL IMAGES USING
VISUALISATION TOOLKIT (VTK)**

GRACE ANABELA HENRY DUSIM

**DISSERTATION SUBMITTED IN PARTIAL
FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF BIOMEDICAL
ENGINEERING**

**FACULTY OF ENGINEERING
UNIVERSITY OF MALAYA
KUALA LUMPUR**

2020

UNIVERSITY OF MALAYA
ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Grace Anabela Henry Dusim

Matric No: KQB190006 @ 17036777/2

Name of Degree: Master of Biomedical Engineering

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

"Three-Dimensional (3d) Reconstruction of Ultrasound Foetal Images Using
Visualisation Toolkit (VTK)"

Field of Study: Biomedical Engineering

I do solemnly and sincerely declare that:

- (1) I am the sole author/writer of this work;
- (2) This work is original;
- (3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the work and its authorship have been acknowledged in this work;
- (4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
- (5) I hereby assign all and every rights in the copyright to this work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
- (6) I am fully aware that if in the course of making this work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature

Date: 09.09.2020

Subscribed and solemnly declared before,

Witness's Signature

Date:

Name:

Designation:

Acknowledgements

The completion of this undertaking could not have been possible without the participation and assistance of so many people whose names may not all be enumerated. Their contributions are sincerely appreciated and gratefully acknowledged. However, I would like to express my special thanks of gratitude to the following:

Mr Lai Khin Wee, my supervisor who still guides, and teach even when the world is going through a pandemic.

My parents, Henry E. Dusim and Alvernia S. Golokin, whom both has been helping me financially and I believe due to their prayers; I can finish this research project.

My partner, Calvin Sen, who has always been supporting and motivating me, and has gone through all the ups and downs that I faced during the entire project. You mean the world to me.

So with due regards, I express my gratitude to them and to all those that had helped me directly or indirectly.

Thank you.

TABLE OF CONTENTS

Acknowledgements	iii
Abstract	vi
List of Figures	vii
List of Symbols and Abbreviations.....	ix
CHAPTER 1: INTRODUCTION	1
1.1. Problem Statement	2
1.2. Aims and Objectives	3
CHAPTER 2: LITERATURE REVIEW	4
2.1. Two-dimensional (2D) Ultrasound Imaging of foetal	4
2.2. Three-dimensional (3D) Reconstruction.....	6
2.2.1. What is a three-dimensional (3D) Reconstruction?	6
2.2.2. History of three-dimensional (3D) Reconstruction and its Advantages	7
2.3. Application of 3D Ultrasound Reconstruction in Medical Visualisation	9
2.4. 3D Ultrasound Reconstruction Process	10
2.4.1. Data Acquisition.....	11
2.4.2. Data Pre-processing	11
2.4.3. Volume Reconstruction Stage.....	12
2.4.4. 3D Visualisation Stage.....	12
2.5. Visualisation Tool Kit (VTK).....	15
2.6. 3D Slicer	16
CHAPTER 3: METHODOLOGY	17

3.1. 2D Ultrasound Image Acquisitions.....	17
3.2. Volume Rendering Process.....	17
3.3. Coding.....	20
3.3.1. Volume Rendering of the 2D Ultrasound Images.....	20
3.3.2. Surface Rendering of the 2D Ultrasound Images	22
3.3.3. Multiplanar Rendering of the 2D Ultrasound Image	23
3.3.4. Measuring Length between two points	27
3.4. 3D Slicer	32
CHAPTER 4: RESULTS.....	33
4.1. 3D Reconstruction of Foetal Image	33
4.1.1. Surface Rendering Technique.....	33
4.1.2. Volume Rendering Technique	33
4.1.3. Multiplanar Rendering Technique	34
4.2. Measuring the Femur Length.....	34
4.3. Comparison between Visualisation Toolkit (VTK) and 3D Slicer	36
CHAPTER 5: DISCUSSION.....	39
CHAPTER 6: CONCLUSION	44
REFERENCES.....	45

THREE-DIMENSIONAL (3D) RECONSTRUCTION OF ULTRASOUND FOETAL IMAGES USING VISUALISATION TOOLKIT (VTK)

Abstract

Three-dimensional (3D) reconstruction of two-dimensional (2D) ultrasound images has been used for easier analysing of anatomy related to abnormalities. 3D image reconstruction is required as it helps doctors interpret results and assess any defects quickly. Not only does it improve the accuracy of diagnosis, but it also saves a lot of time. In this thesis, 295 slices of 2D ultrasound images of a foetal are used to do 3D reconstruction using surface rendering, volume rendering, and multiplanar rendering technique. A similar software known as 3D Slicer is also used to reconstruct the ultrasound image for comparison purpose. VTK can easily reconstruct the ultrasound image into a 3D image. It also allows for the measurement of foetal biometry, hence, reducing the risk of measurement inaccuracy. Other than that, multiplanar rendering allows the operator to view the 3D image in multiple planes without having to mentally imagine the plane as what has been done in a 2D ultrasound. In conclusion, VTK can clearly and accurately reconstruct a 3D image as well as measuring the length between two points. This software can help a lot of physicians in assessing any foetal abnormalities faster and accurately.

Keywords: 2D ultrasound, 3D reconstruction, visualisation toolkit (VTK), Volume Rendering technique, Surface Rendering technique

List of Figures

Figure 2.1: Sagittal Plane view of foetal	5
Figure 2.2: Coronal Plane view of the foetal	5
Figure 2.3: Classification of 3D Ultrasound Imaging System	6
Figure 2.4: Overall process of 3D Reconstruction	10
Figure 2.5: a) Planar cross-sectional images of reconstructed volume data and b) Cube view of reconstructed volume data	13
Figure 2.6: The difference of planar and nonplanar volume rendering in the assessment of scoliosis.....	13
Figure 2.7: The different volumes rendering visualisation of 3D ultrasound imaging system approaches where the left image shows the maximum intensity projection of a foetus and right image shows the minimum intensity projection of blood vessels in the liver	14
Figure 2.8: Left Image: The indirect surface rendering of cardiac structure; Right Image: The direct surface rendering of a magnetic resonance heart phantom	14
Figure 2.9: Basic Flow of Visualisation Toolkit (VTK) pipeline mechanism.....	15
Figure 3.1: Pipeline Mechanism for Volume Rendering	17
Figure 3.2: Inheritance Diagram for vtkJPEGReader	17
Figure 3.3: Smoother Surface due to filtering.....	18
Figure 3.4: Inheritance Diagram for vtkPolyDataMapper	18
Figure 4.1: 3D Reconstructed Image using Surface Rendering	33
Figure 4.2: 3D Reconstruction of the 2D Ultrasound Image using Volume Rendering ..	33
Figure 4.3: Multiplanar in Visualisation Toolkit; Sagittal Plane (Black); Axial Plane (Red); Coronal Plane (White)	34
Figure 4.4: 3D Reconstruction of 2D Ultrasound Image	34
Figure 4.5: Different Plane Positions	35

Figure 4.6: Cross-sectional View of the 3D Rendered Image	35
Figure 4.7: Measurement of Femur Length	35
Figure 4.8: Volume Rendering done using a) Visualisation Toolkit b) 3D Slicer.....	36
Figure 4.9: Viewing different plane in a single window using 3D Slicer.....	36
Figure 4.10: Pre-set for 3D Model Display.....	37
Figure 4.11: Advanced Settings in 3D Slicer for the Volume Properties	37
Figure 4.12: Measuring Femur Length in 3D Slicer	38
Figure 4.13: Available Modules in 3D Slicer	38

List of Symbols and Abbreviations

CT	: Computed Tomography
MRI	: Magnetic Resonance Imaging
2D	: Two-dimensional
3D	: Three-dimensional
VTK	: Visualisation Toolkit
A-mode	: Amplitude Mode
B-mode	: Brightness Mode
M-mode	: Motion Mode
ROI	: Region of Interest
MR	: Magnetic Resonance
PCA	: Principal Component Analysis
PBM	: Pixel-based method
VBM	: Voxel-based method
FBM	: Function-based method
DICOM	: Digital Imaging and Communications in Medicine
RGB	: Red Green Blue
Ob-gyn	: Obstetrics and Gynaecology

CHAPTER 1: INTRODUCTION

Medical imaging is a technique used in creating images of the human body for clinical purposes. It is used for analysing the anatomy of the human body that is related to abnormalities. Clinicians nowadays interpret the human body by looking at it layer by layer. Some of the techniques used are ultrasound, computed tomography (CT), and magnetic resonance imaging (MRI) (Brinkmann & Kline, 1998). In comparison with CT and MRI, ultrasound has a much lower risk; hence it is safer, cheaper, non-invasive, and non-traumatic. Due to these advantages, ultrasound is, therefore, the first choice and the most popular diagnostic tool. The entire image of internal organs such as the abdomen including the genitourinary system can be obtained by using diagnostic ultrasound. However, when a traditional two-dimensional (2D) ultrasound imaging is used, it is unable to quantify the volume of structures of interest in the body since the only view that can be accessed and seen is a 2D frame on a specific set of time. Hence, volume quantification is vital in assessing the progression of a particular disease as well as tracking the response rate of a treatment. Thus, three-dimensional (3D) ultrasound imaging has been getting a lot of attention in recent years, especially in hospitals and medical centres that offer very high-quality services.

The 3D ultrasound systems are consisting of four different types, i.e. tracked free-hand, untracked free-hand, mechanical assemblies, and 2D arrays. In a tracked free-hand system, the operator holds an assembly that has transducer and an attachment, then manipulates it over the anatomy until the 2D images are digitised as the sensor is moved. For the untracked free-hand system, the transducer is run by the operator in a steady and regular motion. At the same time, the 2D images are being digitised, and to reconstruct the 3D model, a linear or angular spacing between scanned images is assumed. In mechanical localiser system, the transducer is mechanically translated or rotated, while 2D ultrasound images are digitised at predefined spatial or angular intervals.

Lastly, for 2D arrays system, it generates a pyramidal pulse of ultrasound and processes the echoes to create 3D information in real-time.

A 3D reconstruction process refers to the generation of a 3D image from a digitised set of 2D images. Currently, two approaches can be used that is either a 3D surface model or voxel-based volume. In order to visualise information in the 3D image, the rendering technique is the most critical. Some of the primary methods are surface-based viewing technique, multi-plane viewing techniques, and volume-based rendering techniques.

In this thesis, a raw dataset of 295 multiple 2D images slices of images will be used. 3D reconstruction is done by using volume rendering techniques by utilising the visualisation toolkit (VTK) visualisation tool. This research aims to develop a program to visualise the medical object, i.e. foetal images in 3D formation.

The paper is structured as follow; chapter 2 describes some of the histories in 3D reconstruction, its application, and the process of reconstructing the 3D model. It also focuses on VTK, which is the software used in this thesis. Chapter 3 introduces the methodology used in obtaining the 3D model and the algorithms used. The results of the methods are shown in chapter 4, along with discussions in chapter 5. At last, a conclusion is drawn in chapter 6.

1.1. Problem Statement

The reconstruction of a 3D ultrasound image based on 2D images is a famous method used to analyse some anatomy related to abnormalities. Ideally, 2D ultrasound images are good enough for the patients to see foetal images as they can still see the baby. However, in reality, the conventional 2D ultrasound imaging for diagnosis purpose has limitations in quantifying the volume of structures of interest in the body since only a 2D frame is produced at a given time. Therefore, volume quantification is important in assessing the progression of the disease, or in this case, the foetal development and tracking of the progress of response to treatment are given.

Hence this study aims to develop a program to visualise the medical object in 3D formation. With this program, a doctor can diagnose and plan treatment more effectively and accurately. Since it is a program, the human cognitive dependent will be less and finally, less time required for human interpretation.

1.2. Aims and Objectives

The aim of this study is to develop a program that can visualise an anatomy, which in this thesis is foetal.

Objectives:

- a. To visualise the foetal by applying 3D reconstruction algorithm using Visualisation Toolkit (VTK)
- b. To measure the femur length of the foetal using Visualisation Toolkit (VTK)
- c. To compare the 3D Reconstruction between Visualisation Toolkit (VTK) and 3D Slicer

CHAPTER 2: LITERATURE REVIEW

This section explains about 2D and 3D ultrasound imaging system, past work recommending the usage of 3D reconstruction and the advantages, the process of reconstruction, and introduce both VTK and 3D Slicer software.

2.1. Two-dimensional (2D) Ultrasound Imaging of foetal

2D ultrasound imaging is also known as a scan. It is used mostly to get an image of the anatomy. It works by using ultrasound waves that are radiated from a transducer. The choice of frequency is vital as higher frequencies provide better spatial resolution but with a downside of having less depth (Moran & Thomson, 2020). Ultrasound images are essentially 2D cross-sectional slices through the body with the portion of anatomy closest to the transducer is displayed at the top of the screen and organs more distal to be presented at depth. In ultrasound, there are three modes of imaging technique; A-mode or amplitude mode, B-mode or brightness mode, and M-mode or motion mode (Abramowicz, Kossoff, Maršál, & Ter Haar, 2000; Meyer-Baese & Schmid, 2014). The A-mode is the simplest method and uses mostly in ophthalmology to determine the relative distances between different regions of the eye, and in localisation of the brain midline, or myocardium infarction. The B-mode can be used for both stationary and moving structures such as cardiac valve motion. This mode is the safest as the acoustic outputs are low (Abramowicz et al., 2000). Finally, for the M-mode, it is mostly used in conjunction with an electrocardiogram (ECG) to see the motion of the heart valves. Both the B- and M-mode are known to be safe for all stages of pregnancy.

In this thesis, the focus will be on the usage of ultrasound to create images of a developing foetus. During pregnancy, a 2D ultrasound scan is usually done during mid-pregnancy approximately when one is 20 to 24 weeks pregnant (“Ultrasound - 2D | imagingandultrasound,” n.d.). Ultrasound is performed as a matter of routine prenatal upkeep for all pregnant women.

In a 2D scan for pregnant women, it will focus on looking at the baby's bones, heart, spinal cord, face, kidneys, and abdomen. The scan will also check for the number of the foetus inside the womb and the position of the placenta. One of the purposes of doing an ultrasound is to see the development of the foetus. For example, to see whether there are any foetal abnormalities such as cleft pallet, heart abnormality, and Spina Bifida. With a 2D ultrasound, in order to diagnose the foetal properly, multiple planes must be used. For example, a sagittal plane view allows visualisation of the foetal profile as can be seen in fig. 2.1, which is the most typical view shown and given to the parents after a

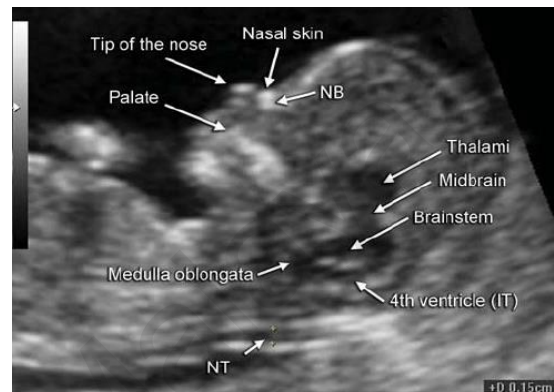


Figure 2.1: Sagittal Plane view of foetal

visit to the clinic. One of the planes used in order to assess the integrity of the eyes and lips of the foetal is axial or coronal planes as can be seen in fig. 2.2. As can be seen from these two figures, multiple planes are needed in order to view the image of the foetal. This requires a lot of time and human cognitive dependent.



Figure 2.2: Coronal Plane view of the foetal

During a 2D ultrasound, the clinicians will measure the femur length, biparietal diameter which is the diameter of the foetal head, head circumference, and abdominal circumference (Yang, Leung, Lee, Chan, & Tang, 2010). The measurements of this foetal biometry require the cognitive ability of the operator since the image of the foetal is only shown in 2D. All these measurements are the basis for accurate determination of gestational age of the foetal. It also allows for the clinicians to detect any foetal growth abnormalities. Hence, it is important for the operator to measure it accurately.

For beginners, using a real-time 2D ultrasound for the measurement of the anatomic segments of the foetal can take a long time since it can be challenging to obtain an appropriate image plane for the analyses. Some of the problems that may be encountered during the process of foetal biometry include unfavourable foetal position, unfamiliarity with the foetal anatomy, and manipulation of the ultrasound transducer may affect the quality of the foetal image.

2.2. Three-dimensional (3D) Reconstruction

2.2.1. What is a three-dimensional (3D) Reconstruction?

A 3D reconstruction is created by using multiple images in order to create a model, which

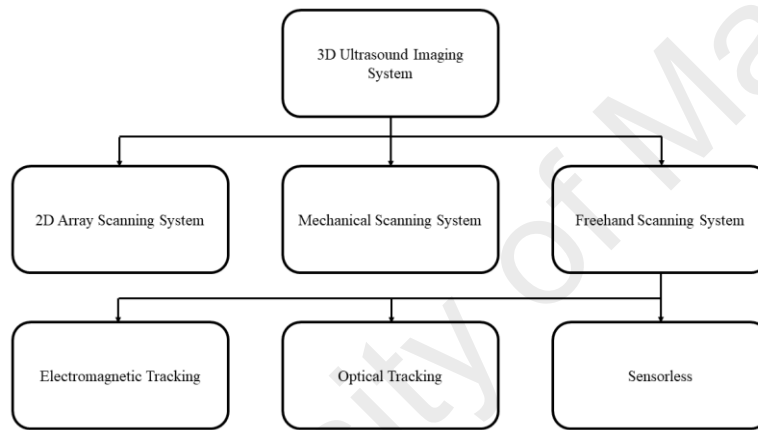


Figure 2.3: Classification of 3D Ultrasound Imaging System

in 3D by reconstructing and combining a set of 2D ultrasound frames, which can be viewed from different positions and angles of that ROI. Different scanning methods or techniques are used to capture the set of 2D ultrasound frames. Fig. 2.3 shows the classification of the 3D ultrasound imaging system (Mohamed & Vei Siang, 2019).

Since data acquisition plays a vital role in the accuracy and applicability of 3D reconstruction, selecting the most suitable 3D ultrasound imaging system is crucial. The choice is highly dependent on the application. For instance, a mechanical scanning system is ideal for transrectal ultrasound examination to evaluate the prostate gland in the human body.

2.2.2. History of three-dimensional (3D) Reconstruction and its Advantages

In the last decades, researchers have attempted to enhance medical images and develop methods that can reconstruct the 3D image. Bandyopadhyaya et al. (2016) have proposed an integrated tool that is used for the detection and evaluation of orthopaedic fractures in long bone by using a digital x-ray image. For the assessment and the diagnosis of a patient's case such as looking for abnormalities in foetal, it is much easier to be detected and interpreted by looking at 3D images (Bandyopadhyay, Biswas, & Bhattacharya, 2016). Knowing the 3D structure of anatomy can provide valuable information, especially when it is applied in a clinical setting. Çimen et al. (2016) have presented a review of the approaches to the reconstruction of high-contrast coronary arteries from x-ray angiography. They discussed the potential role of reconstruction in clinical decision making and interventional guidance (Çimen, Gooya, Grass, & Frangi, 2016). Other researchers that have shown interest in using 3D reconstruction are Arakeri and Reddy (2013). They proposed the usage of 3D reconstruction for a brain tumour which allows for the estimation of its volume from a set of 2D cross-sectional magnetic resonance (MR) images of the brain. It is done by using segmentation, inter-slice interpolation, mesh generation, and simplification methods (Arakeri & Reddy, 2013). These researchers have shown that it is possible to use 2D images to create a 3D model for assessment and diagnosis purpose.

Some of the advantages of using 3D reconstruction in comparison to 2D images can be summarised as follows (Fenster & Downey, 2000, 2003; Gebhard, Eubanks, & Meeks, 2015; Gunay, Shim, & Shimada, 2007; Hsu, Prager, Gee, & Treece, 2009);

- a. The entire structure of an organ can be visualised in 3D:** The methods used to visualise the structure and ROI are by using interpolation, segmentation, and registration methods in 2D or 3D space.

- b. Less human-cognitive dependency:** Since this is a program, the 3D visualisation of anatomy during the examination can be screened directly on the user interface instead of mentally transforming 2D B-scans to perceive a 3D view. Hence, this eliminates operator analysis dependence in the acquisition process. Furthermore, the usage of 2D transducer requires the operator to be experienced as well as knowing the human anatomy, whereas, in 3D reconstruction, the acquisition step is much easier and less user-dependent.
- c. Orientation-dependent Visualisation:** Working in a 3D domain, viewing any arbitrarily oriented planes, even those that are impossible to visualise by conventional ultrasound methods are possible. 3D reconstruction enables clinicians to discuss, re-examine, and depict the inside of a patient's body from different viewpoints after they left the clinic. Next, the sharing of data with other experts is possible. Hence, a better diagnosis can be made by referring to other experts in the field.
- d. Measurements of quantitative aspects in three-dimensions:** In clinical applications, most clinicians will need to measure the volume or other quantitative properties such as the size of the tumour, which is usually done by interpreting a 2D image of the anatomy. However, by using 3D volumes for measurement, it will present a more accurate measurement as the volume of the structure can be measured easily. It has been shown that the usage of 3D allows for an inexperienced operator to measure the foetal biometry accurately and with high quality results (Yang et al., 2010).
- e. Repeatability of ROI examination:** Physicians like to see the ROI from precisely the same perspective as in the previous review in order to monitor the progression of therapy, to perform a follow-up study of a disease, to observe the response of the body to the treatment and to plan treatment. By using 3D reconstruction, it facilitates these types of assessments and improves the accuracy of the results.

In a 2D ultrasound, to accomplish this task, it is arduous due to misregistration and difficulties in adjusting the transducer into a similar position.

- f. Fusion with other 3D image modalities:** 3D reconstruction can be easily used with other imaging modalities, such as MRI, which is more efficient and valuable, especially in diagnosing cancer.

2.3. Application of 3D Ultrasound Reconstruction in Medical Visualisation

The feasibility of 3D ultrasound imaging in clinical applications has increased significantly due to the improvement of data acquisition methods, 3D reconstruction algorithms, volume visualisations, and hardware capabilities. Hence, the usage of 3D ultrasound imaging has become more relevant in the medical field due to their flexibility, efficiency, and real-time applicability.

The second advantage is it can assist in the accurate volume measurement of the foetus size. During an ultrasound scan, the physicians need to measure the femur length, abdominal circumference, biparietal diameter, and head circumference (“Ultrasound scan—fetal growth scan - Patient Information Brochures - Mater Group,” n.d.).

The 3D ultrasound imaging used in obstetrics and gynaecology has two main advantages. Firstly, similar to 2D ultrasound imaging, it can be used to determine the number of foetuses, however, in 3D ultrasound, the foetus’s surface feature can be seen, and of course the placenta location. The volume rendering in 3D reconstruction can distinguish between tissues and surrounding amniotic fluids very well; hence it is suitable to view foetal ultrasound image for the physicians to examine the foetal presentation, as well as for the parents to see the foetus’s face. In 2D ultrasound, the physicians have to construct the image into a volumetric view mentally; hence it depends on the skills and experience of the physicians to interpret and give an accurate result. However, in 3D ultrasound, it can reduce the mental workload of the physicians as well as reducing the repeatability of them to relocate the placenta location.

This is done to determine whether the foetal is oversized or undersized (Mohamed & Vei Siang, 2019). By using 3D ultrasound, it is much easier to measure them quickly and accurately.

Other than these two main advantages, another advantage is the ability of the 3D ultrasound in assisting the clinicians in assessing any foetal abnormalities. 3D ultrasound enables a more detailed survey of the embryo and the foetus in comparison to 2D ultrasound. The availability of several display modes and standardised examinations permits the demonstration of both the normal and abnormal foetal anatomy in controlled planes and rendered images from different angles (Merz & Pashaj, 2017; Merz & Welter, 2005). This allows the demonstration of even subtle foetal defects to be viewed from the most optimal angle. When the physicians are counselling the parents, the rendered images can then be used to help the parents understand the severity of the existing malformation, or conversely, to ensure them of the absence of any foetal abnormality.

2.4. 3D Ultrasound Reconstruction Process

This section explained the reconstruction process. The standard workflow of 3D ultrasound reconstruction process according to Mohamed et al. (2019) and Mozaffari et al. (2017) is consist of data acquisition stage, pre-processing data stage, volume reconstruction method stage, and 3D visualisation stage. Fig. 2.4 shows the overall process of 3D reconstruction (Mozaffari & Lee, 2017).

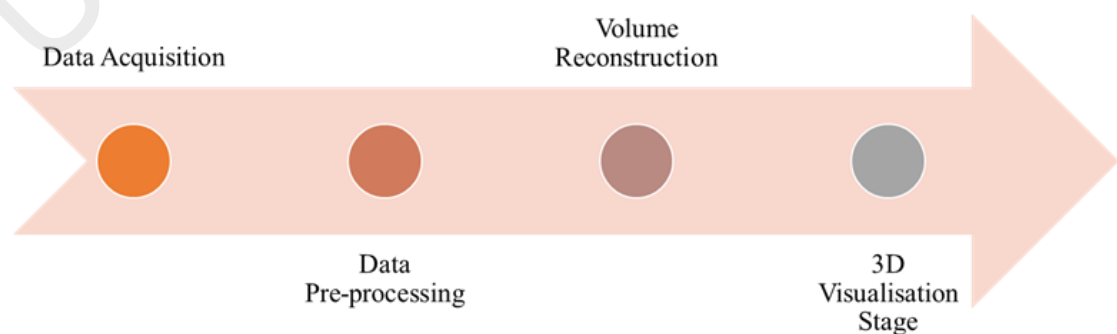


Figure 2.4: Overall process of 3D Reconstruction

2.4.1. Data Acquisition

The data can be directly obtained from any ultrasound scanning systems that are presented in fig. 2.3. The collected data were the 2D ultrasound frames and the orientation and position of the tracking sensor when a particular frame was taken. The B-scan image and its relative orientation and position must be synchronised. As for the real-time system, there is a need to synchronise the image captured, position, and the time. This process of synchronisation is also known as temporal calibration (Mohamed & Vei Siang, 2019; Mozaffari & Lee, 2017).

Next, in order to get the homogenous transformation, ultrasound probe calibration or also known as spatial calibration is used to convert each 2D coordinate pixel in 2D ultrasound frames into 3D coordinates voxel of ultrasound probe frame. This method is mostly used in real-time ultrasound 3D reconstruction system. Scan conversion is essential as well due to the possibility of using a different coordinate system of varying scanning devices.

2.4.2. Data Pre-processing

After obtaining the data, it will be sent to the workstation for further processing. Most image processing techniques are used during this stage to enhance the quality of the frames, remove any noise, and preserve the edge boundary. Since 2D structure have various types of noise and artefacts such as speckle noise, refraction, shadowing, the spatial resolution within a 2D ultrasound frame is not uniform due to the transducer, and signal characteristics vary with penetration depth. Some of the techniques used in enhancing the image are noise removing techniques, histogram equalisation, 2D Gaussian filter, and median filtering.

2.4.3. Volume Reconstruction Stage

The most important part of the 3D reconstruction process is the volume reconstruction stage. It involves the implementation of interpolation and approximation algorithm to get the 3D volume data and put them in a 3D volume grid based on the spatial information acquired from the tracking system. This stage aims to reduce computational requirements without damaging or losing the underlying shape of the data. Before any reconstruction of volume can occur, the coordinate system and volume grid of reconstructed volume need to be established. Some of these are volume size, the axis of volume, the origin of axis, and the size of the voxel.

The volume coordinate configuration uses principal component analysis (PCA). It is a statistical tool that estimates the most significant difference of collected data that the volume can enclose all the data values. Bounding box technique is configured by computing the volume size by filling all the voxel with pixels from a series of 2D ultrasound frames, and then the maximum point and the minimum point can be obtained. After volume coordinate has been configured, volume reconstruction can then be performed. Several standard methods of volume reconstruction are pixel-based method (PBM), voxel-based method (VBM), and function-based method (FBM). The visualisation toolkit (VTK) is the most common software used for volume reconstruction and visualisation, which is also used in this thesis.

2.4.4. 3D Visualisation Stage

After the third stage, i.e. volume reconstruction, the 3D visualisation stage is used to display the volume data from the volume grid for the physicians to see the results of ultrasound scanning. This is useful for the clinicians to analyse the scanned anatomy and assist in diagnosis.

For example, to see whether the foetal has fully formed hands and legs. This stage is the last step to complete the fully-functioning 3D ultrasound reconstruction. Some of the most common techniques or rendering algorithms are multiplanar rendering, volume rendering, and surface rendering (Mozaffari & Lee, 2017).

2.4.4.1. Multiplanar Rendering

This is a method where 2D ultrasound planes, also known as a resliced image, are extracted from the 3D sound data and displayed to the user with a 3D impression. The clinicians can then view the 3D ultrasound results on three

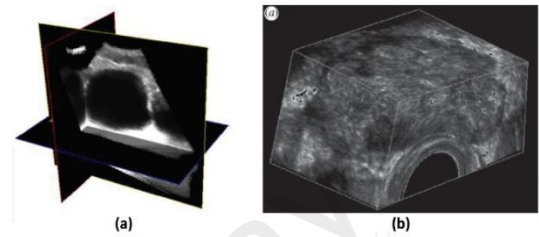


Figure 2.5: a) Planar cross-sectional images of reconstructed volume data and b) Cube view of reconstructed volume data

orthogonal slice views, i.e. transverse plane or also known as axial plane, coronal plane, and sagittal plane. The resliced image is presented together with texture-mapped 3D rendering. There are three approaches of the display, i.e. the planar cross-sectional images, the cube view, and the orthogonal planes. The limitation of this planar viewing is the possibility of a loss of information due to the complex shape of the anatomy, which usually happens in viewing spinal

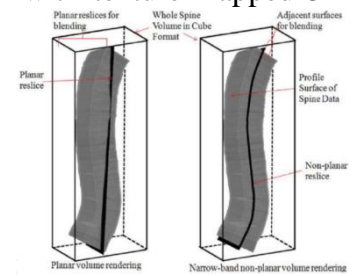


Figure 2.6: The difference of planar and nonplanar volume rendering in the assessment of scoliosis

curvature (Mohamed & Vei Siang, 2019). To overcome this limitation, the nonplanar volume rendering method can be used. Due to its simplicity and less computational power required, multiplanar rendering are the most favourable method among researchers and practitioners. An example of planar cross-sectional images and cube view can be seen in fig. 2.5 whereas fig. 2.6 shows the difference between planar and nonplanar volume rendering.

2.4.4.2. Volume Rendering

This technique involves ray-casting or also known as ray-marching techniques where the change of light that went through the 3D volume data is projected as the output visualisation results for the operator to view

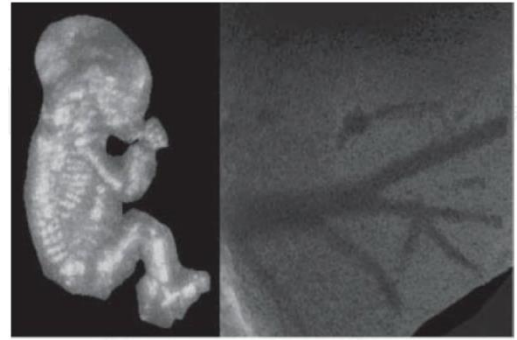


Figure 2.7: The different volumes rendering visualisation of 3D ultrasound imaging system approaches where the left image shows the maximum intensity projection of a fetus and right image shows the minimum intensity projection of blood vessels in the liver

(Mohamed & Vei Siang, 2019; Shahidi & Kaufman, 1996). Every voxel has attributes such as brightness, transparency, and colour. A voxel is the volume element which is defined in a 3D space (“What is voxel? - Definition from WhatIs.com,” n.d.). Volume rendering can differentiate between tissue and fluids very well; hence it is suitable to view 3D ultrasound foetal image. However, volume rendering uses a lot of power in the central processing unit (CPU) of a computer and is not suitable to view soft tissues details. Fig. 2.7 is an example of volume rendering visualisation for foetal and blood vessels in the liver.

2.4.4.3. Surface Rendering

Surface rendering is a technique that produces a 3D surface-based on segmented boundary data points by generating the surface triangles or polygons associated with standard surface-rendering methods being provided by interpolation. This technique can improve the interpretation of data sets, and it can be classified into indirect surface rendering and direct surface rendering. Direct surface rendering is a particular case of volume rendering technique, where the surface is rendered directly

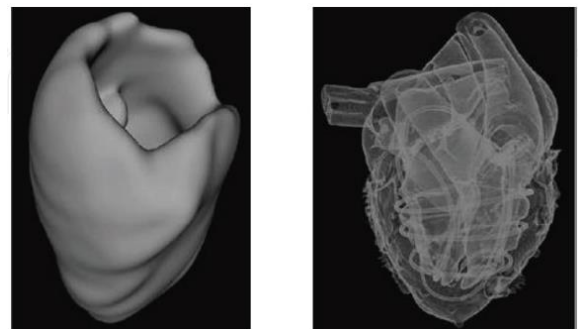


Figure 2.8: Left Image: The indirect surface rendering of cardiac structure; Right Image: The direct surface rendering of a magnetic resonance heart phantom

from the volume without intermediate geometric representations, setting thresholds or

using object labels to define a range of voxel intensities to be viewed; hence, why it is called direct (Mohamed & Vei Siang, 2019).

Transparency and colours are used for better 3D visualisation of the volume of the ROI. For indirect surface rendering, the surfaces of the structure boundaries within the volume need to be identified prior to segmentation. An example of indirect surface rendering is contour filtering and marching cubes. Fig. 2.8 shows the 3D visualisation using surface rendering technique.

2.5. Visualisation Tool Kit (VTK)

VTK is an open-source, object-oriented software system for computer graphics, visualisation, and image processing. It is used by thousands of researcher and developers around the world. It aims mainly at medical imaging visualisation, hence the reason it is being used in the 3D reconstruction. VTK implements OpenGL rendering, recently updated to OpenGL Core, which can be used in external applications. Significant effort has been focused on VTK's volume rendering capabilities, with a major rebuild in 2014. Volume rendering functionality includes cropping, composite, and additive blending, maximum and minimum intensity projections, and so on (Wheeler et al., 2018).

In VTK, the image data is classified into different categories, i.e. structural point, line grid, structured grid, unstructured points, polygon data, and unstructured grids. The various filter will be allocated for each type of structure for processing. According to Lai et al. (2011), the most basic flow of VTK pipeline mechanism is similar to the one shown in fig. 2.9 below.

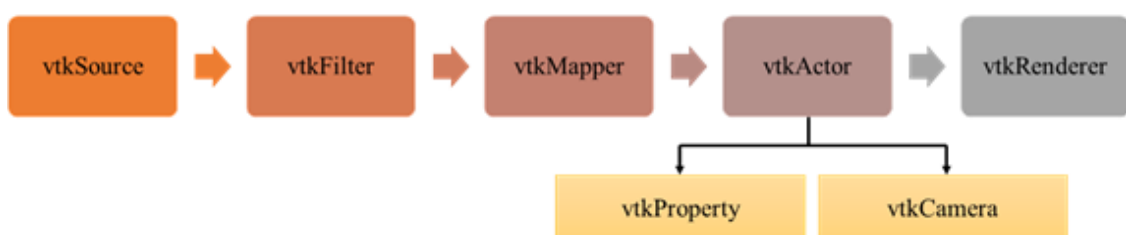


Figure 2.9: Basic Flow of Visualisation Toolkit (VTK) pipeline mechanism

vtkObject provides basic visualised method flow and is a basic class in the VTK library. The source data is used to generate the data layer. vtkSource is a derived class from vtkObject and is parent class of vtkFilter, which is a derived class from vtkSource. The vtkFilter implements data processing on raw data through filters, converting it into a module that can directly be processed using an algorithm model. vtkMapper is also derived from vtkObject. The vtkMapper will undergo a different filtering process that will be converted into geometric data as an interface in connecting original data and image data. It can achieve a mapping from data to graphics. The vtkActor is used to realise the visual using Mapper results to show the resultant image. vtkActor manipulates image display properties through the vtkProperty. For example, it can help to make the image more realistic. The final step is to display the image on a computer screen. Hence, vtkRendered is used to accomplish this task by using Render & Render Window. An interactor known as Render Window Interactor is used to achieve interoperability between the image and user. By controlling this method, the image can be easily rotated using a mouse to see images from all angles (Lai, Hum, & Supriyanto, 2011).

2.6. 3D Slicer

3D Slicer is open-source software that can reconstruct and visualise medical image data into a 3D model. Unlike VTK, 3D Slicer does not need for the operator to do any coding or algorithm. This software allows the user to directly import their data into the software in any format such as DICOM file or JPEG file format. 3D Slicer was first introduced by Ron Kikinis in 1999 (Zhang, Zhang, Pan, & Chang, 2019). Most of the development work of this software is led by professional engineers, algorithm developers, and scientist in the application field. 3D Slicer supports multiple operating systems, such as Windows, and Mac OS X. It has been used not only in the medical field but also in non-medical image analysis.

CHAPTER 3: METHODOLOGY

The reconstruction of 3D images consists of a few steps, i.e. 2D image acquisition and image processing. In this thesis, the software that will be used is visualisation toolkit (VTK).

3.1. 2D Ultrasound Image Acquisitions

The images used in this thesis is obtained from 2D ultrasound images. There are 295 layers of images used to construct the pictures into 3D. The output resolution of the 3D reconstructed image is 320 x 240.

3.2. Volume Rendering Process

All of the slices of images are to be read as a volume into the system by using the function `vtkJPEGReader` in VTK. The process of volume rendering in this thesis follow the pipeline shown in fig. 3.1 (“3D volume rendering using vtk toolkit. | Quppler,” n.d.).

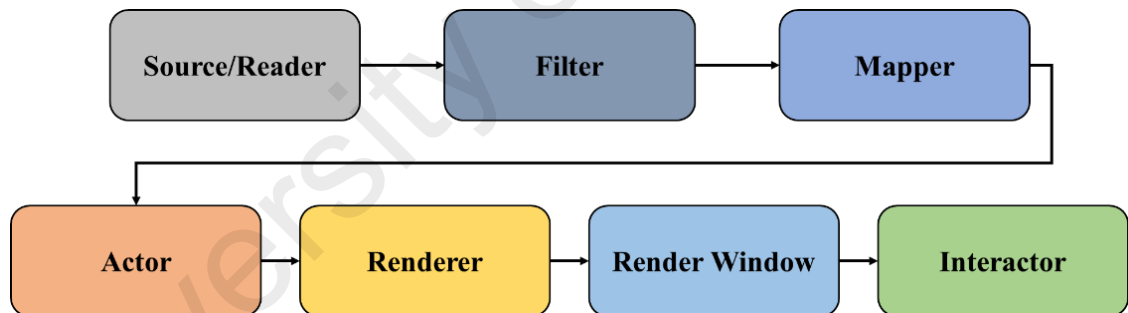


Figure 3.1: Pipeline Mechanism for Volume Rendering

a. Source/Reader:

Various reader can be used to construct the objects, which in this thesis ‘`vtkJPEGReader`’ is used since the source is from 2D ultrasound images. The inheritance diagram for `vtkJPEGReader` can be seen in fig. 3.2.

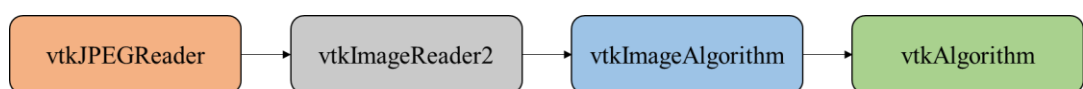


Figure 3.2: Inheritance Diagram for `vtkJPEGReader`

b. Filter:

This stage takes the data as input and modifies it in a certain way that is suitable for the anatomy and returns the modified data. In this thesis, a filter is used to process the 2D images into a 3D model that is clear and smoother.

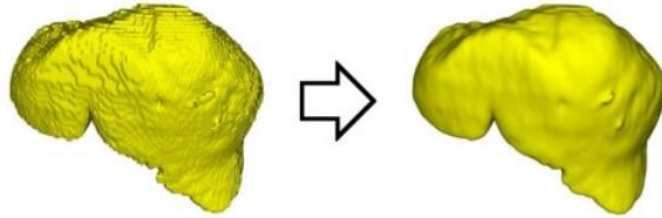


Figure 3.3: Smoother Surface due to filtering

c. Mapper

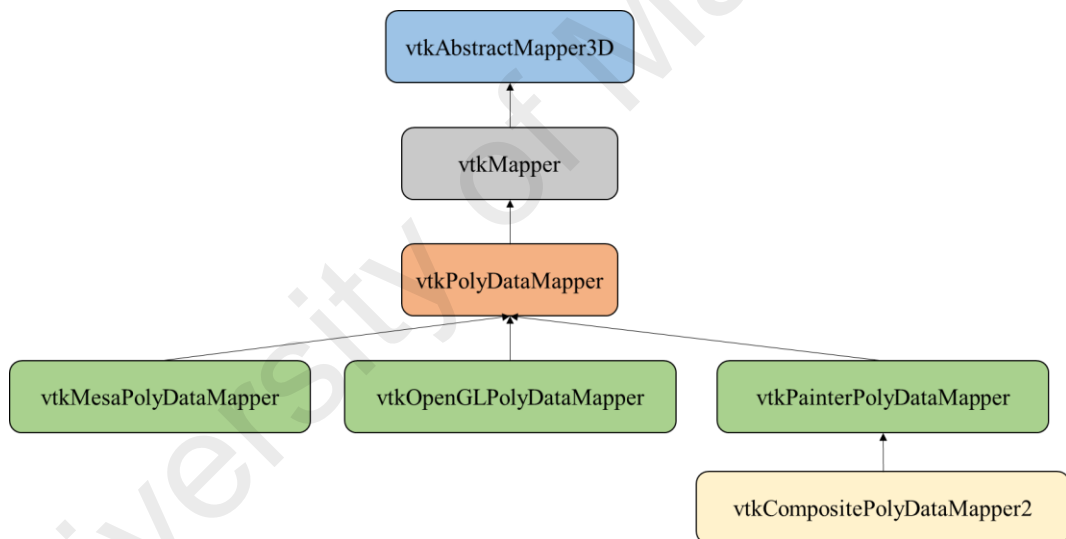


Figure 3.4: Inheritance Diagram for vtkPolyDataMapper

Mapper is used to map the data to graphics primitives that can be displayed by the renderer. vtkPolyDataMapper is used in this thesis, which maps polygonal data to the graphics primitives. Fig. 3.4 shows the inheritance diagram for vtkPolyDataMapper.

d. Actor:

vtkActor is used to represent the anatomy in a rendering scene. It has a position, scale, orientation, various rendering properties, and textures. It can adjust any visible features such as the opacity, colours, etc.

e. Renderer

vtkRenderer provides an abstract specification for renderers. It controls the rendering process for the objects. It is the process of converting geometry, a specification for lights, and a camera view into an image. It also performs coordinate transformation between world coordinates, view coordinates, and display coordinates. This stage can also set the colour of the render window background.

f. Render Window

vtkRenderWindow creates a window for renderers to view the final reconstructed 3D model.

g. Interactor

It provides platform-independent window interaction via the mouse and keyboard. It allows the users to rotate/zoom/pan the camera.

3.3. Coding

This section will be explaining the coding used and how it affects the reconstruction of the 3D foetal image. In this thesis, different coding is used to generate different visualisation results. The first coding reads a volume dataset and displays it via volume rendering technique, whereas the second coding uses surface rendering technique to construct the 3D image. The next coding creates three orthogonal planes which is also known as multiplanar rendering and finally, the last coding allows the reconstructed 3D image's cross-sectional views to be observed in different planes as well as measuring the length between two points, hence allowing for the measurement of femur length.

3.3.1. Volume Rendering of the 2D Ultrasound Images

Similar to the pipeline mechanism shown in the previous section, a reader should first be defined. In this program, vtkJPEGReader is used. The following line of coding allows the reader to read a series of 2D slices of images that compose the volume. The slice dimensions are set using the SetDataExtent where 319 is the size of the pixels in the x-direction, and 239 is the size of the pixels in the y-direction. The value 295 is the number of layers or the number of images used in reconstructing the 3D image.

```
vtkJPEGReader *v16 = vtkJPEGReader::New();

v16->SetDataExtent(0,319,0,239,1,295);
v16->SetFilePrefix("C:\\Users\\grace\\Desktop\\RESEARCH
PROJECT\\IMAGE1\\readyimageimg");

v16->SetFilePattern("%s%d.jpg");
v16->SetDataSpacing(1, 1, 1);
v16->SetDataOrigin(0.0, 0.0, 0.0);
v16->Update();
```

The volume will be displayed by using ray-cast alpha compositing technique where a ray-cast mapper is used.

```
vtkSmartPointer<vtkVolumeRayCastCompositeFunction> rayCastFunction =
vtkSmartPointer<vtkVolumeRayCastCompositeFunction>::New();

vtkSmartPointer<vtkVolumeRayCastMapper> volumeMapper =
vtkSmartPointer<vtkVolumeRayCastMapper>::New();
volumeMapper->SetInput(v16->GetOutput());
volumeMapper->SetVolumeRayCastFunction(rayCastFunction);
```

Next, filter is applied using `vtkColorTransferFunction`. It does this by mapping the voxel intensities to colours. This is modality-specific, as well as anatomy-specific. Hence, the value is often changed accordingly.

```
vtkSmartPointer<vtkColorTransferFunction>volumeColor =
vtkSmartPointer<vtkColorTransferFunction>::New();
volumeColor->AddRGBPoint(0, 0.0, 0.0, 0.0);
volumeColor->AddRGBPoint(50, 1.0, 0.3, 0.1);
volumeColor->AddRGBPoint(250, 1.2, 1.2, 1.1);
```

Next, opacity transfer function is used to control the opacity of different types of tissue using the `vtkPiecewiseFunction`. Increasing the value will increase the opacity of the tissues, whereas decreasing it would decrease the opacity of the tissues.

```
vtkSmartPointer<vtkPiecewiseFunction> volumeScalarOpacity =
vtkSmartPointer<vtkPiecewiseFunction>::New();
volumeScalarOpacity->AddPoint(0, 0.0);
volumeScalarOpacity->AddPoint(50, 0.2);
volumeScalarOpacity->AddPoint(250, 0.9);
```

Some filters such as colours is also applied using the `vtkVolumeProperty` where it attaches the colour and opacity functions to volume. In order for a high-quality rendering to be achieved, the interpolation is set to linear. `ShadeOn` is used to turn on directional lighting, which enhance the appearance of the 3D rendered view. However, it should be noted that the quality of the shading depends on how accurately the volume can be calculated. Hence, for noisy data, the gradient estimation will be poor.

The impact of the shading is influenced by the `SetAmbient`, `SetDiffuse`, and `SetSpecular`. By decreasing the `SetAmbient` value, and increasing the other two, the shading of the 3D image will decrease whereas increasing the `SetAmbient` value, and decreasing the `SetDiffuse` and `SetSpecular` would increase the impact of shading.

```
vtkSmartPointer<vtkVolumeProperty> volumeProperty =
vtkSmartPointer<vtkVolumeProperty>::New();
volumeProperty->SetColor(volumeColor);
volumeProperty->SetScalarOpacity(volumeScalarOpacity);
volumeProperty->SetGradientOpacity(volumeGradientOpacity);
volumeProperty->SetInterpolationTypeToLinear();
volumeProperty->ShadeOn();
volumeProperty->SetAmbient(0.2);
volumeProperty->SetDiffuse(0.8);
volumeProperty->SetSpecular(0.4);
```

3.3.2. Surface Rendering of the 2D Ultrasound Images

As mentioned in chapter 2, 3D reconstruction can be done using different methods. In this thesis, both volume rendering and surface rendering is used. The line of coding used in surface rendering is shown below. At first, the reader used is the same i.e. `vtkJPEGReader`.

```
vtkJPEGReader *v16 = vtkJPEGReader::New();

v16->SetDataExtent(0,99,0,99,1,295);
v16->SetFilePrefix("C:\\Users\\grace\\Desktop\\RESEARCH
PROJECT\\newimage\\readyimageimg");
v16->SetFilePattern("%s%d.jpg");
v16->SetDataSpacing(1, 1, 1);
v16->SetDataOrigin(0.0, 0.0, 0.0);
v16->Update();
```

Next, `vtkPolyDataNormals` is used to allow for smooth surface shading during the rendering process.

```
vtkSmartPointer<vtkContourFilter> skinExtractor =
vtkSmartPointer<vtkContourFilter>::New();
skinExtractor->SetInputConnection(v16->GetOutputPort());
skinExtractor->SetValue(0, 100);

vtkSmartPointer<vtkPolyDataNormals> skinNormals =
vtkSmartPointer<vtkPolyDataNormals>::New();
skinNormals->SetInputConnection(skinExtractor->GetOutputPort());
skinNormals->SetFeatureAngle(60.0);

vtkSmartPointer<vtkStripper> skinStripper =
vtkSmartPointer<vtkStripper>::New();
skinStripper->SetInputConnection(skinNormals->GetOutputPort());

vtkSmartPointer<vtkPolyDataMapper> skinMapper =
vtkSmartPointer<vtkPolyDataMapper>::New();
skinMapper->SetInputConnection(skinStripper->GetOutputPort());
skinMapper->ScalarVisibilityOff();
```

The colour of the skin can be changed by assigning different values to the `SetDiffuseColor` using RGB values.

```
vtkSmartPointer<vtkActor> skin =
vtkSmartPointer<vtkActor>::New();
skin->SetMapper(skinMapper);
skin->GetProperty()->SetDiffuseColor(1, .49, .25);
skin->GetProperty()->SetSpecular(.3);
skin->GetProperty()->SetSpecularPower(20);
skin->GetProperty()->SetOpacity(.5)
```

The same line of coding is used for bone. The colour of the bone can also be changed using the `SetDiffuseColor`.


```

vtkSmartPointer<vtkContourFilter> boneExtractor =
vtkSmartPointer<vtkContourFilter>::New();
boneExtractor->SetInputConnection(v16->GetOutputPort());
boneExtractor->SetValue(0, 200);

vtkSmartPointer<vtkPolyDataNormals> boneNormals =
vtkSmartPointer<vtkPolyDataNormals>::New();
boneNormals->SetInputConnection(boneExtractor->GetOutputPort());
boneNormals->SetFeatureAngle(60.0);

vtkSmartPointer<vtkStripper> boneStripper =
vtkSmartPointer<vtkStripper>::New();
boneStripper->SetInputConnection(boneNormals->GetOutputPort());

vtkSmartPointer<vtkPolyDataMapper> boneMapper =
vtkSmartPointer<vtkPolyDataMapper>::New();
boneMapper->SetInputConnection(boneStripper->GetOutputPort());
boneMapper->ScalarVisibilityOff();

vtkSmartPointer<vtkActor> bone =
vtkSmartPointer<vtkActor>::New();
bone->SetMapper(boneMapper);
bone->GetProperty()->SetDiffuseColor(1, 1, .9412);

```

3.3.3. Multiplanar Rendering of the 2D Ultrasound Image

Before the planes can be created, the renderer, render window, and interactor is created to allow the usage of mouse and keyboard-based interaction.

```

vtkSmartPointer<vtkRenderer> aRenderer =
vtkSmartPointer<vtkRenderer>::New();
vtkSmartPointer<vtkRenderWindow> renWin =
vtkSmartPointer<vtkRenderWindow>::New();
renWin->AddRenderer(aRenderer);
vtkSmartPointer<vtkRenderWindowInteractor> iren =
vtkSmartPointer<vtkRenderWindowInteractor>::New();
iren->SetRenderWindow(renWin);

```

Next, the type of reader used for the program needs to be declared. In this thesis, vtkJPEGReader is used, which allows the program to read a JPEG image format. SetDataExtent is used to declare the size of the pixels of the image, i.e. (x_{min} , x_{max} , y_{min} , y_{max} , z_{min} , z_{max}). Since the resolution of the image starts from 0, that will be the x_{min} whereas x_{max} will be 319. This is similar to the size of the pixels at y, where y_{min} is 0 and y_{max} is 239.

For z_{min} , the value is 1 since the first image is numbered as 1, and the z_{max} is 295 since there are 295 layers of 2D images used in this program. The location of the image will also need to be declared in the SetFilePrefix line. The SetDataSpacing is used to set the

spacing of the cubical cells that compose the dataset, i.e. the width, height, and length, whereas `SetDataOrigin` is used to set the origin of the dataset.

```
vtkJPEGReader *v16 = vtkJPEGReader::New();
v16->SetDataExtent(0,319,0,239,1,295);
v16->SetFilePrefix("C:\\Users\\grace\\Desktop\\RESEARCH
PROJECT\\IMAGE1\\readyimageimg");
v16->SetFilePattern("%s%d.jpg");
v16->SetDataSpacing(1, 1, 1);
v16->SetDataOrigin(0.0, 0.0, 0.0);
v16->Update();
```

Next, filter is applied. Once it is generated, `vtkPolyDataNormals` filter is used to create a smooth surface shading during the rendering process. By setting the value of skin extractor 'SetValue' (0, x) to a value larger than 255, no image will be shown in the render window. However, a value that is lower than 80 causes the image formed to be scattered around the window, hence, no clear image of the foetal can be seen.

```
vtkSmartPointer<vtkContourFilter> skinExtractor =
vtkSmartPointer<vtkContourFilter>::New();
skinExtractor->SetInputConnection(v16->GetOutputPort());
skinExtractor->SetValue(0, 85);
skinExtractor->Update();
```

The feature angle specifies the angle that defines a sharp edge.

```
vtkSmartPointer<vtkPolyDataNormals> skinNormals =
vtkSmartPointer<vtkPolyDataNormals>::New();
skinNormals->SetInputConnection(skinExtractor->GetOutputPort());
skinNormals->SetFeatureAngle(60.0);
skinNormals->Update();
```

`vtkStripper` is used to allow for a much faster rendering process.

```
vtkSmartPointer<vtkStripper> skinStripper =
vtkSmartPointer<vtkStripper>::New();
skinStripper->SetInputConnection(skinNormals->GetOutputPort());
skinStripper->Update();

vtkSmartPointer<vtkPolyDataMapper> skinMapper =
vtkSmartPointer<vtkPolyDataMapper>::New();
skinMapper->SetInputConnection(skinStripper->GetOutputPort());
skinMapper->ScalarVisibilityOff();
```

The coding lines below allows for the colour of the 3D reconstructed image to be change according to the preferences of the user.

```
vtkSmartPointer<vtkActor> skin =
vtkSmartPointer<vtkActor>::New();
skin->SetMapper(skinMapper);
skin->GetProperty()->SetDiffuseColor(1, .49, .25);
skin->GetProperty()->SetSpecular(1);
skin->GetProperty()->SetSpecularPower(80);
```

Next, the same line of coding is used for boneExtractor except the contour value for the it is changed to 100 instead of 85.

```
vtkSmartPointer<vtkContourFilter> boneExtractor =
vtkSmartPointer<vtkContourFilter>::New();
boneExtractor->SetInputConnection(v16->GetOutputPort());
boneExtractor->SetValue(0, 100);

vtkSmartPointer<vtkPolyDataNormals> boneNormals =
vtkSmartPointer<vtkPolyDataNormals>::New();
boneNormals->SetInputConnection(boneExtractor->GetOutputPort());
boneNormals->SetFeatureAngle(60.0);

vtkSmartPointer<vtkStripper> boneStripper =
vtkSmartPointer<vtkStripper>::New();
boneStripper->SetInputConnection(boneNormals->GetOutputPort());
vtkSmartPointer<vtkPolyDataMapper> boneMapper =
vtkSmartPointer<vtkPolyDataMapper>::New();
boneMapper->SetInputConnection(boneStripper->GetOutputPort());
boneMapper->ScalarVisibilityOff();

vtkSmartPointer<vtkActor> bone =
vtkSmartPointer<vtkActor>::New();
bone->SetMapper(boneMapper);
bone->GetProperty()->SetDiffuseColor(1, 1, .9412);
```

Next, three orthogonal planes, i.e. sagittal plane, coronal plane, and axial plane is created using the line of coding shown below. Each plane has different colours. Firstly, a black/white lookup table is created.

```
vtkSmartPointer<vtkLookupTable> bwLut =
vtkSmartPointer<vtkLookupTable>::New();
bwLut->SetTableRange(0, 2000);
bwLut->SetSaturationRange(0, 0);
bwLut->SetHueRange(0, 0);
bwLut->SetValueRange(0, 1);
bwLut->Build();
```

Then, another lookup table consists of full hue circle obtained from the hue-saturation-value (HSV) which ranges from 0 – 255. Next, the first plane is created. vtkImageMapToColors is a filter used to map the data through the corresponding lookup table created above. vtkImageActor is a type of vtkProp that displays an image on a single quadrilateral plane. The process is done using texture mapping, hence, it allows for a faster rendering process.

```
vtkSmartPointer<vtkImageMapToColors> sagittalColors =
vtkSmartPointer<vtkImageMapToColors>::New();
sagittalColors->SetInputConnection(v16->GetOutputPort());
sagittalColors->SetLookupTable(bwLut);
sagittalColors->Update();
```

VTK allows for the plane to have their own particular colour that has been set in the lookup table. For sagittal plane, the colour is white, red for axial plane, and black for coronal plane. The `SetDisplayExtent` is used to specify the position of each of the plane;

$(x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max})$.

```
vtkSmartPointer<vtkImageActor> sagittal =
vtkSmartPointer<vtkImageActor>::New();
sagittal->SetInput(sagittalColors->GetOutput());
sagittal->SetDisplayExtent(178,178, 0,180, 1,295);
```

Same approach is used to create the axial and coronal plane except the `SetDisplayExtent` is changed according to where it will be positioned in the image.

```
vtkSmartPointer<vtkImageMapToColors> axialColors =
vtkSmartPointer<vtkImageMapToColors>::New();
axialColors->SetInputConnection(v16->GetOutputPort());
axialColors->SetLookupTable(hueLut);
axialColors->Update();

vtkSmartPointer<vtkImageActor> axial =
vtkSmartPointer<vtkImageActor>::New();
axial->SetInput(axialColors->GetOutput());
axial->SetDisplayExtent(180,180, 0,180, 200,200);

vtkSmartPointer<vtkImageMapToColors> coronalColors =
vtkSmartPointer<vtkImageMapToColors>::New();
coronalColors->SetInputConnection(v16->GetOutputPort());
coronalColors->SetLookupTable(satLut);
coronalColors->Update();

vtkSmartPointer<vtkImageActor> coronal =
vtkSmartPointer<vtkImageActor>::New();
coronal->SetInput(coronalColors->GetOutput());
coronal->SetDisplayExtent(0,120, 200,200, 1,295);
```

Now, after creating all the three planes, it is best to create an initial view of the data. The

FocalPoint and **Position** form a vector direction.

```
vtkSmartPointer<vtkCamera> aCamera =
vtkSmartPointer<vtkCamera>::New();
aCamera->SetViewUp (0, 0, -1);
aCamera->SetPosition (0, 1, 0);
aCamera->SetFocalPoint (0, 0, 0);
aCamera->ComputeViewPlaneNormal();
aCamera->Azimuth(30.0);
aCamera->Elevation(30.0);
```

`addActor` is then added to present the entity in a rendering scene. It inherits functions related to the actors position, an orientation from the `vtkProp` stated in the line of coding above.

```

aRenderer->AddActor(outline);
aRenderer->AddActor(sagittal);
aRenderer->AddActor(axial);
aRenderer->AddActor(coronal);
aRenderer->AddActor(skin);
aRenderer->AddActor(bone);

```

In this set of coding, the bone visibility will be turned off to allow for the foetal to be seen as a whole.

```
bone->VisibilityOff();
```

The opacity is set to 0.8. A value lower than this causes the rendered image to be slightly transparent, hence not a clear image can be seen.

```
skin->GetProperty()->SetOpacity(0.8);
```

Initial camera view is created. Dolly is used to set the camera's distance from the focal point.

```

aRenderer->SetActiveCamera(aCamera);
renWin->Render();
aRenderer->ResetCamera();
aCamera->Dolly(1);

```

3.3.4. Measuring Length between two points

Since this program allows the measurement between two points, the length of one pixel need to be defined. However, since the 2D ultrasound images is obtained in the format of JPEG and not DICOM, the size of one pixel could not be identified. Hence, for the purpose of showing the ability of VTK to allow for the measurement of distance between any two points, the pixel length is assigned to 0.2645. The value 0.2645 is used in this thesis as it is equivalent to one pixel ("Convert Pixel (X) to Millimeter," n.d.). However, it should be noted that this is just an estimation of the pixel size of the image.

```

double millimeter;
millimeter=length * 0.2645 * 2;

char text[120];
sprintf( text, "The length of two points is %5.5f cm", millimeter);
textMapper->SetInput( text );
textActor->SetPosition(10, 10);
textActor->VisibilityOn();

style->OnRightButtonDown();
break;
case vtkCommand::LeftButtonReleaseEvent:
    if (MouseMotion == 0)
    {

```

```

    int *pick = iren->GetEventPosition();
    picker->Pick((double)pick[0], (double)pick[1], 0.0, ren1);
    }
    style->OnLeftButtonUp();
    break;
case vtkCommand::MouseMoveEvent:
    MouseMotion = 1;
    style->OnMouseMove();
    break;

```

A similar line of coding is used for the vtkJPEGReader. It is used to allow the input of the coding to be the 2D ultrasound image which is in the format of JPEG.

```

vtkJPEGReader *v16 = vtkJPEGReader::New();
v16->SetDataExtent(0,319,0,239,1,295);
v16->SetFilePrefix("C:\\Users\\grace\\Desktop\\RESEARCH
PROJECT\\IMAGE1\\readyimageimg");
v16->SetFilePattern("%s%d.jpg");
v16->SetDataSpacing(1, 1, 1);
v16->SetDataOrigin(0.0, 0.0, 0.0);
v16->Update();

```

Next, filter is applied to allow for the rendered 3D images to be as smooth and clear as it can. 1024 is the pixels value + the desired opacity value of the foetal. In this thesis, 20 is used as it gives a better rendered 3D image. A value higher than 20 causes the image to be brighter, hence less details can be seen on the foetal image.

```

vtkPiecewiseFunction *opacityTransferFunction =
vtkPiecewiseFunction::New();
opacityTransferFunction->AddPoint(1024+20, 0.0);
opacityTransferFunction->AddPoint(1024+295,1);

```

vtkColorTransferFunction add colours to the rendered image. AddRGBPoint is used to set the value of (R , G , B) of the foetal image.

```

vtkColorTransferFunction *colorTransferFunction =
vtkColorTransferFunction::New();
colorTransferFunction->AddRGBPoint(1024+295.0, 1, 0.6, 0.3);

```

vtkVolumeProperty is used to represent the properties of the volume rendering.

SetAmbient allows for the surroundings of the foetal image to be brighter or darker. A value less than 8 causes it to be too dark for the viewers to see. Next, SetDiffuse does the same function as SetDiffuse. SetSpecular is use control the lighting in the render window. A larger value gives off a much brighter lighting. Finally, the SetSpecularPower act as a mirror, where a larger value gives off a darker rendered image but with a much better contrast.

```

vtkVolumeProperty *volumeProperty = vtkVolumeProperty::New();
volumeProperty->SetColor(colorTransferFunction);
volumeProperty->SetScalarOpacity(opacityTransferFunction);
volumeProperty->ShadeOn();
volumeProperty->SetInterpolationTypeToLinear();
volumeProperty->SetAmbient(8);
volumeProperty->SetDiffuse(8);
volumeProperty->SetSpecular(1);
volumeProperty->SetSpecularPower(80);

```

In this coding, the volume rendering process of the 3D image is generated using **vtkVolumeRayCastMapper** where the change of light that went through the 3D volume data is projected as the output visualisation results for the operator to view.

```

vtkVolumeRayCastCompositeFunction *compositeFunction =
vtkVolumeRayCastCompositeFunction::New();

```

vtkPlane is used to display a plane that can do cross-sectional view of the foetal image.

```

vtkPlane *plane=vtkPlane::New();

vtkVolumeRayCastMapper *volumeMapper = vtkVolumeRayCastMapper::New();
volumeMapper->SetVolumeRayCastFunction(compositeFunction);
volumeMapper->SetInputConnection(readerImageCast->GetOutputPort());
volumeMapper->AddClippingPlane(plane);
volume = vtkVolume::New();
volume->SetMapper(volumeMapper);
volume->SetProperty(volumeProperty);
volume->PickableOff();

```

Next, filter is applied for the outline of the rendered image where it will create a box in which the 3D rendered image will be position.

```

vtkOutlineFilter *outline = vtkOutlineFilter::New();
outline->SetInputConnection( readerImageCast->GetOutputPort() );
vtkPolyDataMapper *outlinemapper = vtkPolyDataMapper::New();
outlinemapper->SetInputConnection( outline->GetOutputPort() );
vtkActor * outlineactor = vtkActor::New();
outlineactor->SetMapper( outlinemapper );
outlineactor->GetProperty()->SetColor( 0.0, 1, 1);

```

In order to show when a point is click **vtkSphereSource** is used to create a sphere with a radius of 5 mm which will be viewed as a point in the rendering window. The colour of the sphere can be changed using the **SetDiffuseColor** where using RGB value. Currently, $R = 1$, hence it will be creating a red sphere.

```

vtkSphereSource *sphere1 = vtkSphereSource::New();
sphere1->SetRadius(5);
vtkPolyDataMapper *sphereMapper1 = vtkPolyDataMapper::New();
sphereMapper1->SetInput(sphere1->GetOutput());
sphereMapper1->GlobalImmediateModeRenderingOn();
sphereActor1 = vtkActor::New();
sphereActor1->SetMapper(sphereMapper1);

```

```
sphereActor1->GetProperty()->SetDiffuseColor(1,0,0);
```

The second sphere for the second point is also created using the same line of coding shown below.

```
vtkSphereSource *sphere2 = vtkSphereSource::New();
sphere2->SetRadius(5);
vtkPolyDataMapper *sphereMapper2 = vtkPolyDataMapper::New();
sphereMapper2->SetInput(sphere2->GetOutput());
sphereMapper2->GlobalImmediateModeRenderingOn();
sphereActor2 = vtkActor::New();
sphereActor2->SetMapper(sphereMapper2);
sphereActor2->GetProperty()->SetDiffuseColor(1,0,0);
```

Then, after creating the two points, a line is created between the two sphere that has a resolution value of 200. The colour again can be set to either red, green, or blue by setting it in the SetDiffuseColor (R,G,B) where in this case a green line will be used.

```
line=vtkLineSource::New();
line->SetPoint1(0, 0, 0);
line->SetPoint2(0, 0, 0);
line->SetResolution(200);
vtkPolyDataMapper *lineMapper = vtkPolyDataMapper::New();
lineMapper->SetInput(line->GetOutput());
lineActor = vtkActor::New();
lineActor->SetMapper(lineMapper);
lineActor->GetProperty()->SetDiffuseColor(0,1,0);
```

vtkCellPicker is used to select a cell by shooting a ray into graphics window and intersecting with actor's defining geometry; specifically its cells. Beside returning coordinates, actor and mapper, it returns the id of the closest cell within the tolerance along the pick ray, and the dataset that was picked.

```
PickCommand* pickObserver = PickCommand::New();
picker = vtkCellPicker::New();
picker->AddObserver(vtkCommand::EndPickEvent, pickObserver);
```

vtkTextMapper is used to display what the operator has pick in the render window. VTK allows the operator to set the properties of the text according to their preference using the tprop where SetFontFamilytoArial is setting the font of the text to Arial. The tprop function also allows the text to be bigger or smaller according to the suitability and preference of the operator by using SetFontSize (x) and setting the value of x , where a smaller value means a smaller text size and a bigger value means a bigger text size. Like

before, the colour of the text can also be changed by setting the RGB value using the `SetColor` function.

```
textMapper = vtkTextMapper::New();
vtkTextProperty *tprop = textMapper->GetTextProperty();
    tprop->SetFontFamilyToArial();
    tprop->SetFontSize(20);
    tprop->BoldOn();
    tprop->SetColor(1, 0, 0);
textActor = vtkActor2D::New();
    textActor->VisibilityOff();
    textActor->SetMapper(textMapper);
```

Finally, `vtkInteractorStyleTrackballCamera` allows the usage of mouse and keyboard in controlling the camera.

```
vtkInteractorStyleTrackballCamera *style =
vtkInteractorStyleTrackballCamera::New();
vtkCallbackCommand * pickerCommand = vtkCallbackCommand::New();
    pickerCommand->SetClientData(style);
    pickerCommand->SetCallback(PickerInteractionCallback);
    style->AddObserver(vtkCommand::LeftButtonPressEvent,
pickerCommand);
    style->AddObserver(vtkCommand::MouseMoveEvent, pickerCommand);
    style->AddObserver(vtkCommand::LeftButtonReleaseEvent,
pickerCommand);
    style->AddObserver(vtkCommand::RightButtonPressEvent,
pickerCommand);

iren->SetInteractorStyle(style);
iren->SetPicker(picker);

sphereActor1->VisibilityOff();
sphereActor2->VisibilityOff();
lineActor->VisibilityOff();

vtkTIPWCallback *myCallback = vtkTIPWCallback::New();
myCallback->Plane = plane;
myCallback->Volume = volume;

vtkImplicitPlaneWidget *planeWidget = vtkImplicitPlaneWidget::New();
planeWidget->SetInteractor(iren);
planeWidget->SetPlaceFactor(1.25);
planeWidget->GetPlaneProperty()->SetOpacity ( 0.1 );
planeWidget->GetOutlineProperty()->SetColor(0,0,1);
planeWidget->SetOrigin(xx/2,yy/2,zz/2);
planeWidget->SetInput((vtkDataSet *)readerImageCast->GetOutput());
planeWidget->PlaceWidget();
planeWidget->On();
planeWidget->AddObserver(vtkCommand::InteractionEvent,myCallback);

vtkCamera *cam1 = ren1->GetActiveCamera();
    cam1->Zoom(1.5);
```

VTK also allows for the colour of the background of renderer window to be changed using the RGB value as well as the size of the window by using the function `SetSize`. The

SetKeyCode is ('i') is defining it to have a specific function which in this program is turning on/off the visibility of the plane.

```
ren1->AddVolume(volume);  
ren1->AddActor(sphereActor1);  
ren1->AddActor(sphereActor2);  
ren1->AddActor(lineActor);  
ren1->AddActor2D(textActor);  
ren1->AddActor( outlineactor );  
ren1->SetBackground(0.2, 0.3, 0.4);  
renWin->SetSize(1000, 800);  
renWin->Render();  
iren-> SetDesiredUpdateRate(99);  
iren->SetKeyCode('i');
```

3.4. 3D Slicer

Another software is used to create a 3D model of the ultrasound image. Data is uploaded in JPEG format into the 3D slicer software with a similar data spacing which is $x = 1mm, y = 1mm, z = 1mm$. The data spacing value is identical to the one used in VTK to allow for comparison of the 3D model to be done. Volume rendering is then conducted using VTK CPU Ray Casting, which is similar to the one used in VTK. Pre-set is chosen to be MR-Default, and some of the volume properties such as the ambient, diffuse, and specular are set to be 0.2, 0.8, and 0.4 similar to the properties used in VTK. The measurement of the femur length is done using 'Markups', where two points can be added and measured.

CHAPTER 4: RESULTS

4.1. 3D Reconstruction of Foetal Image

In a 2D ultrasound, the image of the foetal is flat, hence the features of the foetal could not be seen clearly. In a 3D reconstructed image, the features of the foetal such as their legs, hands, ears, and eyes can be seen clearly, such as the one shown in fig. 4.1 and 4.2. Two hundred ninety-five slices of ultrasound images are used in generating the 3D model.

4.1.1. Surface Rendering Technique

The image shown in fig. 4.1 is reconstructed using surface rendering technique. From the figure, it can be seen that the image appears less 3D.

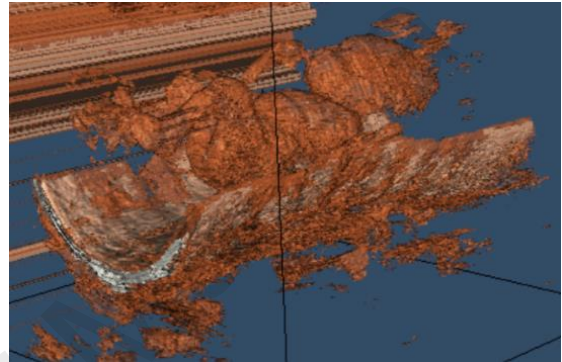


Figure 4.1: 3D Reconstructed Image using Surface Rendering

4.1.2. Volume Rendering Technique

These two images shown below is done by increasing the impact of shading, hence making the foetal image looks more 3D. In this thesis, the quality of the 2D ultrasound image is limited to 320 x 240 pixels. The model has not undergone any image processing to enhance the image qualities such as image contrasting. Hence, the quality of the reconstructed 3D image is affected.

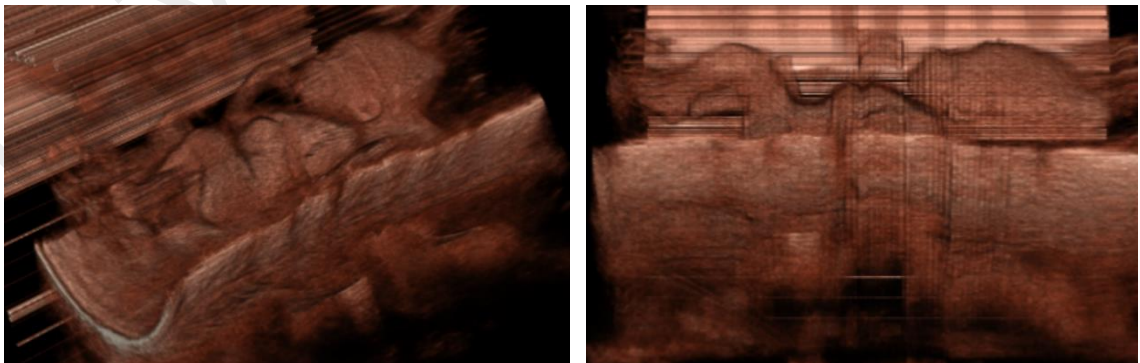


Figure 4.2: 3D Reconstruction of the 2D Ultrasound Image using Volume Rendering

4.1.3. Multiplanar Rendering Technique

Fig. 4.3 is showing the three orthogonal planes; sagittal, coronal, and axial plane created in VTK. Sagittal Plane in fig. 4.2 is shown as black colour, white for the coronal plane, and lastly, red for the axial plane.

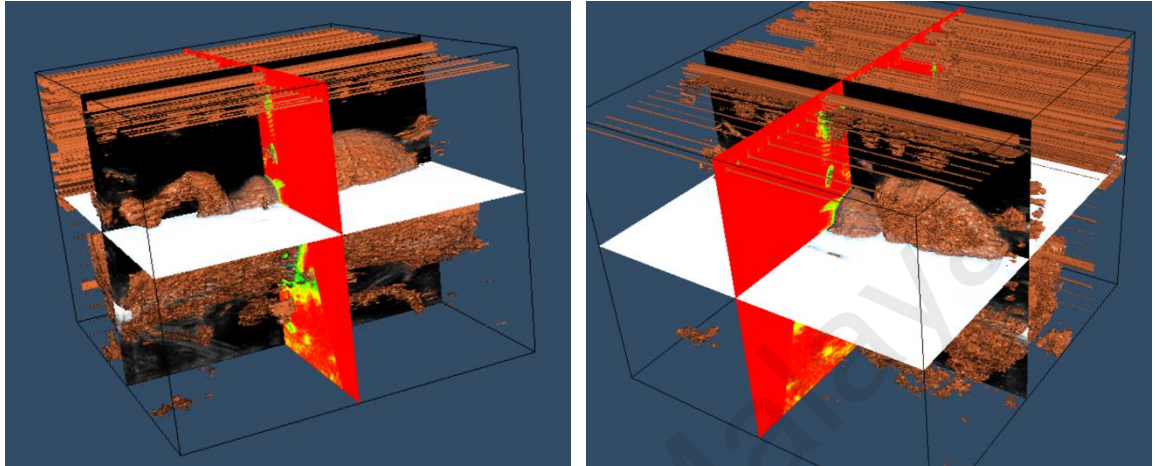


Figure 4.3: Multiplanar in Visualisation Toolkit; Sagittal Plane (Black); Axial Plane (Red); Coronal Plane (White)

4.2. Measuring the Femur Length

In VTK, the program allows for the cross-sectional view of the 3D image to be viewed as well as measuring the length between two points to be obtained. Fig. 4.4 shows a red plane that can be moved 360° by using the white arrow in order to view the cross-section of the image.

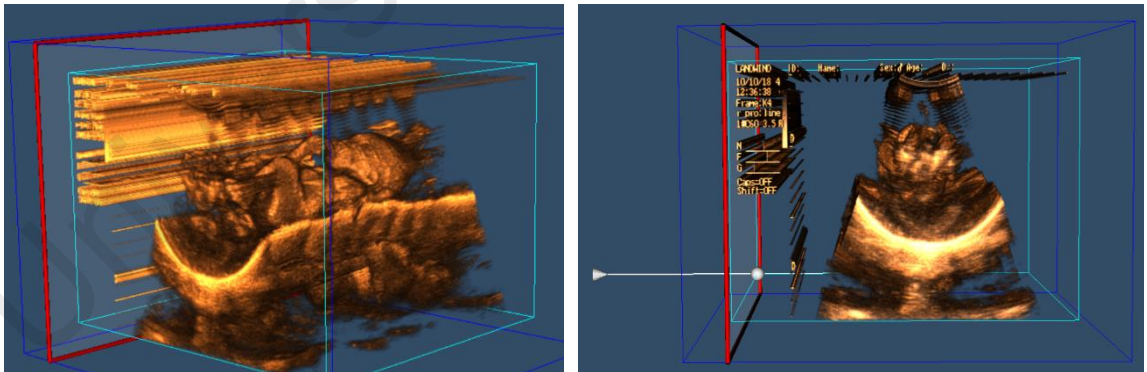


Figure 4.4: 3D Reconstruction of 2D Ultrasound Image

Fig. 4.5 shows some of the position of where the planes can be positioned. The movements of the plane allow the operator to view the foetal according to their inclinations.

As shown in fig. 4.5, from the left, the first image allows the operator to view the side view of the foetal; the second image allows the operator to view the foetal at an angle, whereas the third image allows the operator to view the foetal from the top.

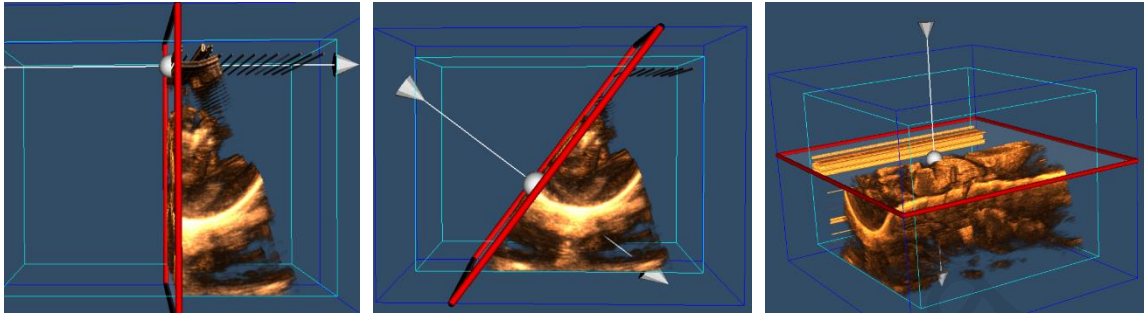


Figure 4.5: Different Plane Positions

Next, fig. 4.6 shows some of the cross-section views when the plane is moved across the foetal. Different cross-section views allow for the physicians to observe any abnormalities on the foetal that could not usually be seen in a 2D ultrasound.

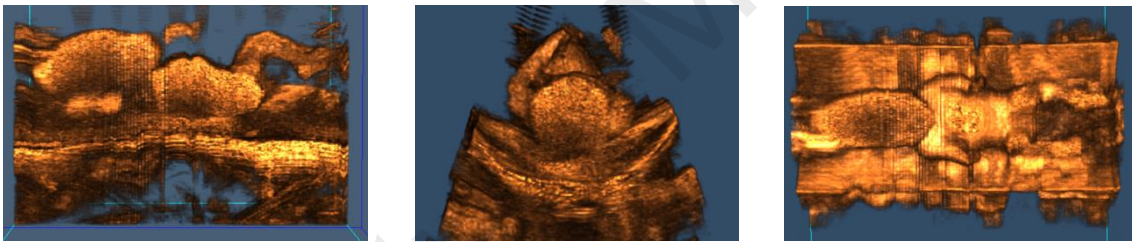


Figure 4.6: Cross-sectional View of the 3D Rendered Image

VTK allows for the measuring of length between two points. In a 2D ultrasound, the clinician usually measures it by viewing the foetal at an angle and manually measure it

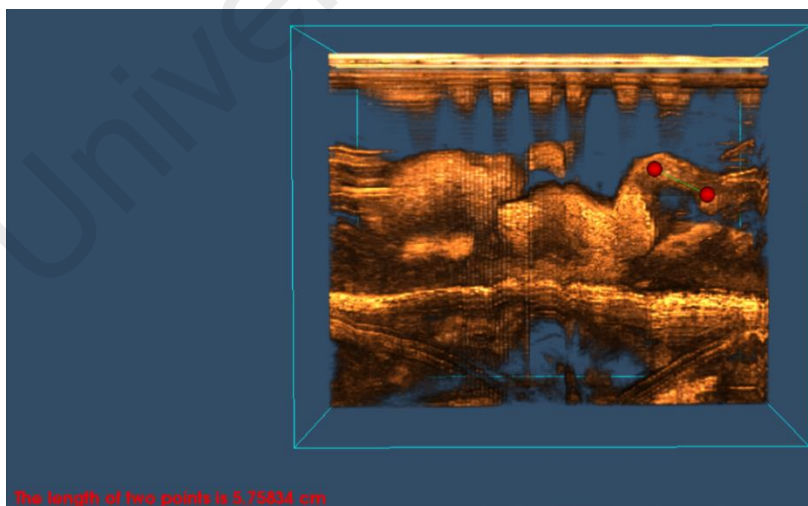


Figure 4.7: Measurement of Femur Length

from the machine screen. Fig. 4.7 shows the measurement taken in VTK in order to determine the length of the femur. As can be seen from fig. 4.7, the length of the femur can

be determined automatically by placing the first and the second sphere in the right position. VTK will then measure and display the distance in the rendering window.

4.3. Comparison between Visualisation Toolkit (VTK) and 3D Slicer

Both VTK and 3D slicer software allows for 2D images to be reconstructed using volume rendering techniques as well as read the input data in either DICOM or JPEG format. The 3D reconstruction shown in fig. 4.8(a) is reconstructed using VTK, whereas fig. 4.8(b) is reconstructed using 3D Slicer.

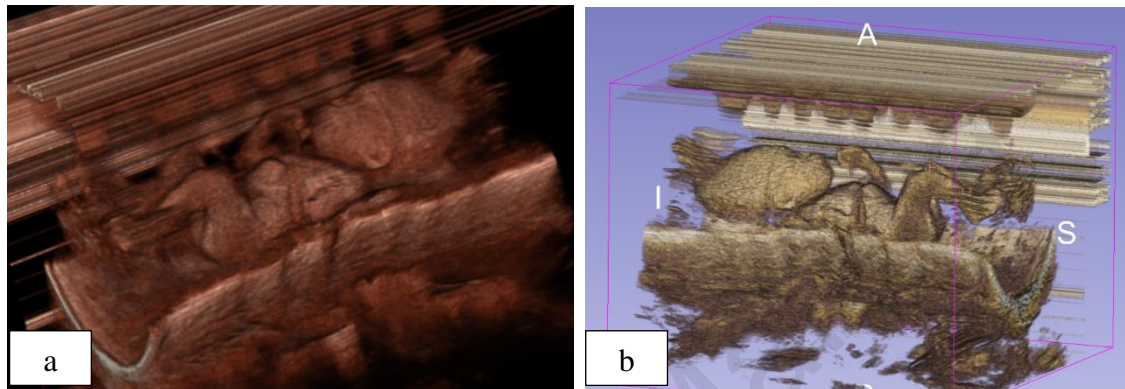


Figure 4.8: Volume Rendering done using a) Visualisation Toolkit b) 3D Slicer

One of the features that do not exist in VTK is the features of viewing different plane in a single window as can be seen in fig. 4.9 below. Instead of using a single movable plane like in VTK, 3D slicer allows the operator to view the cross-section view of the 3D image by moving the slider. By referring to fig. 4.9, the red coloured tab represents the axial plane, the yellow coloured tab is the sagittal plane, and the green coloured tab represents the coronal plane.

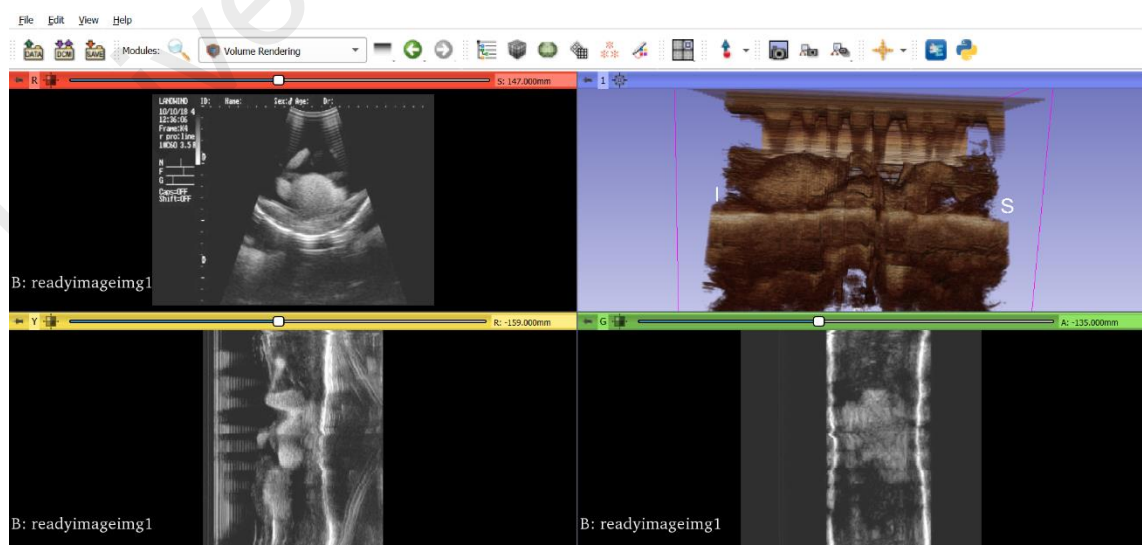


Figure 4.9: Viewing different plane in a single window using 3D Slicer

In terms of colours, VTK allows the operator to set the tone of their 3D model by assigning the RGB values. In 3D Slicer, the display of the 3D model has a pre-set where

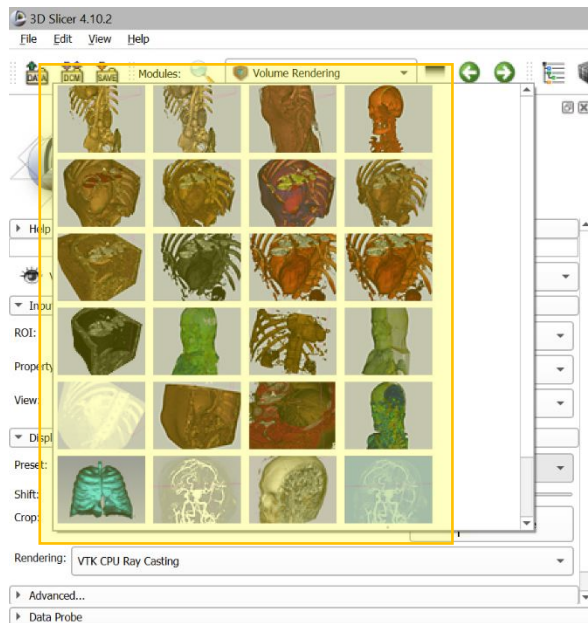


Figure 4.10: Pre-set for 3D Model Display

the operator can choose how they want it to be displayed as can be seen in fig. 4.10, and also an advanced option where the operator can change the volume properties according to the opacity, as well as filters like the one used in VTK such as the ambient, diffuse, specular, and power as can be seen in fig 4.11.

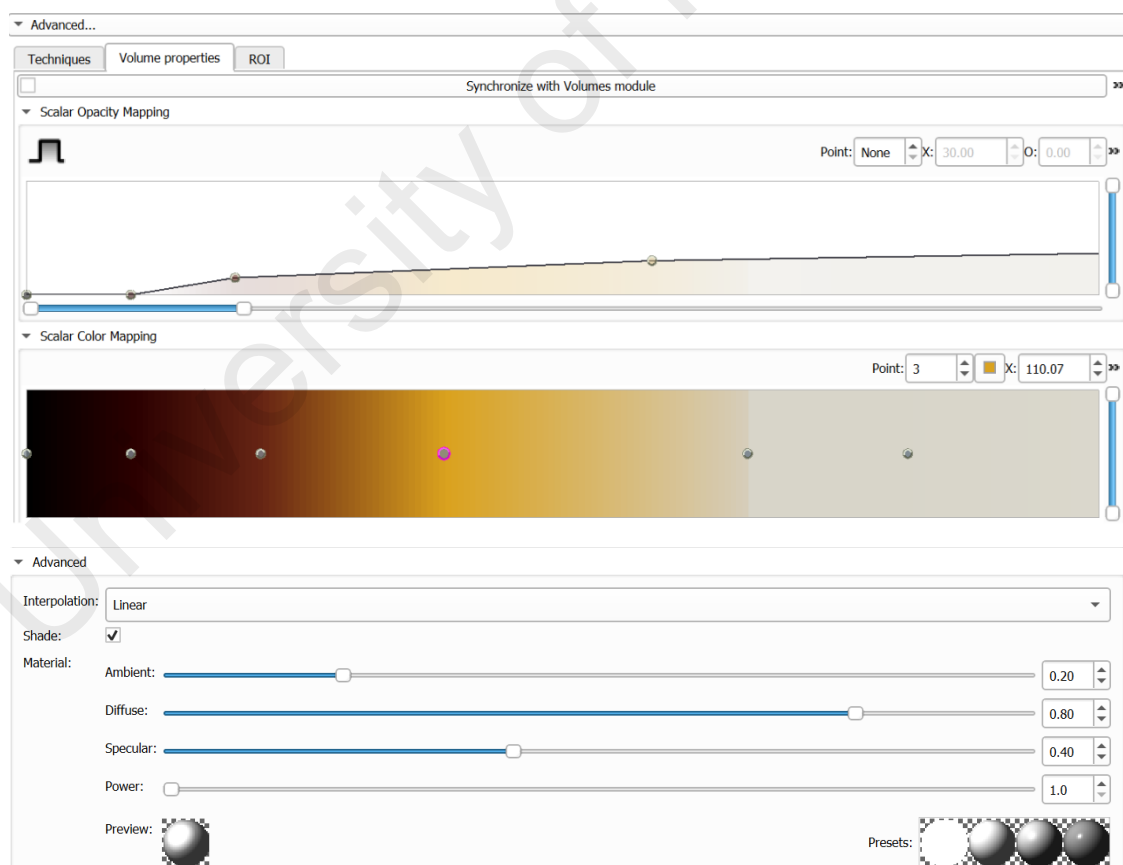


Figure 4.11: Advanced Settings in 3D Slicer for the Volume Properties

Both VTK and 3D Slicer allows for the measurement of between two points. As can be seen in fig. 4.12, the summed linear distance: 29.6 mm is the femur length.

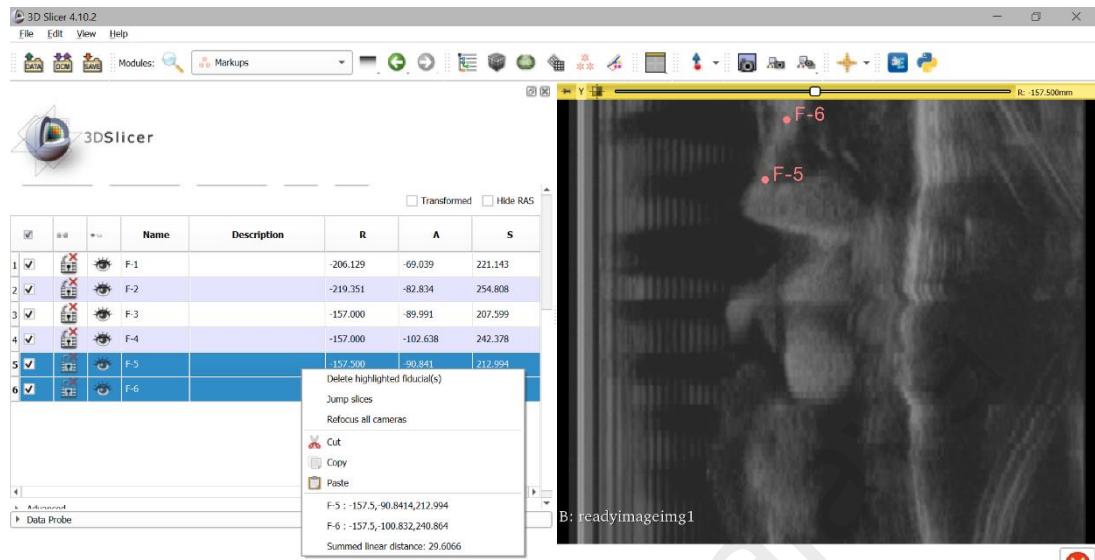


Figure 4.12: Measuring Femur Length in 3D Slicer

Some of the modules or techniques that can be used in 3D Slicer can be seen in fig. 4.13 below.

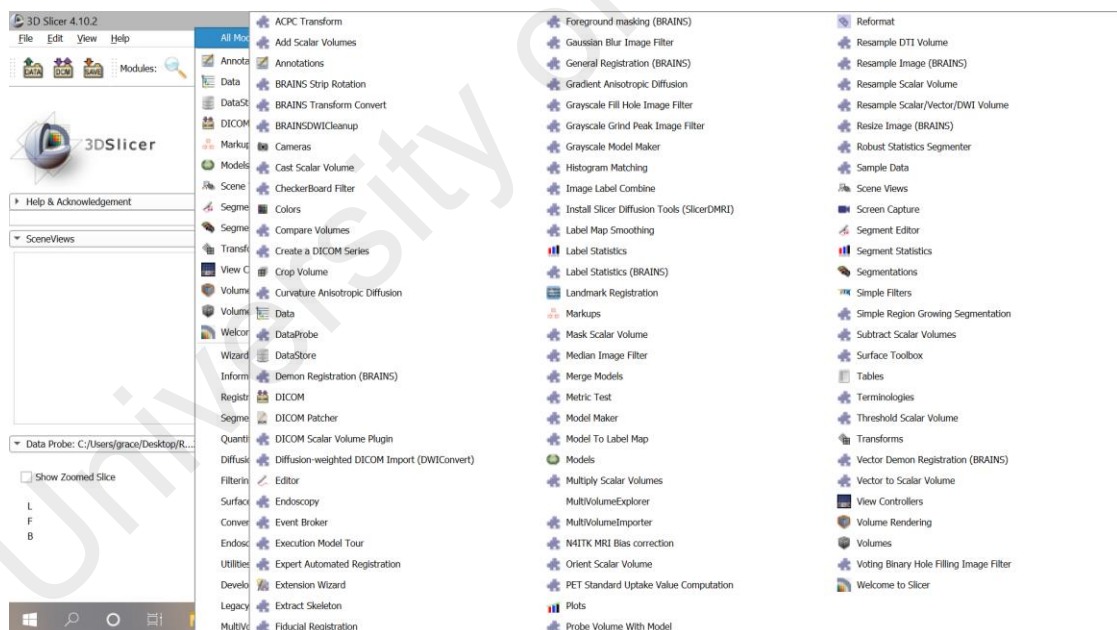


Figure 4.13: Available Modules in 3D Slicer

CHAPTER 5: DISCUSSION

In this thesis, three different techniques are used to reconstruct the 3D image of the foetal. All three methods have successfully reconstructed the 3D model as can be seen in fig. 4.1, 4.2, 4.3, and 4.4. A significant difference can be seen between fig. 4.1 and 4.2 as they are reconstructed using different techniques, i.e. surface rendering for fig. 4.1, and volume rendering technique for fig. 4.2. As can be seen in fig. 4.1, the quality of the 3D image is lower than the one shown in fig. 4.2. One of the main reasons is because when a surface rendering technique is used, it tends to generate a lower quality of an image in a short amount of time. However, in the volume rendering technique, it takes a much longer time as it retains all the volume data during the rendering process, hence a high-quality 3D image is obtained.

Next, based on the results shown in fig. 4.1 and 4.2, it can be seen clearly that the foetal phantom is reconstructed successfully. However, due to the presents of noise in the images, the 3D reconstructed image is not as smooth as what has also been experienced by other researchers (Hafizah, Kok, & Supriyanto, 2010). In order to overcome this problem, image processing can be conducted on the images before using it in VTK, primarily when surface rendering technique is used. For example, as had been done by Hafizah et al. (2010), the process of choosing an ROI can be done. Next, median filtering can be applied to the image where the process of smoothing the model is done. The method of median filtering, when applied to these 2D images, helps in reducing noise without compromising the sharpness of the image. Finally, the 2D images will also go through the operation of image contrasting where the intensity of the image will be increased, hence making the image look sharper.

Another factor that needs to be considered when reconstructing a 3D image from 2D ultrasound image is the number of slices of the medical images affects the whole quality of the rendered image (Hafizah et al., 2010).

Hence, by having more layers of images used in reconstructing the 3D image, a higher quality of the 3D image can be achieved. Since there are only 295 layers of ultrasound image available for this foetal, a comparison between the quality of using a different number of layers could not be conducted. The ability to render 2D ultrasound images into a 3D image allows for the clinicians to diagnose issues such as cleft lip; hence very useful for the examination of foetal anatomy (Yang et al., 2010). Also, since 3D ultrasound is less operator-dependent, the accuracy in diagnosing any foetal anomalies are improved.

Next, in a 2D ultrasound, the clinicians will mentally imagine the orthogonal planes, i.e. the sagittal, coronal, and axial plane (“ULTRASOUND EVALUATION OF THE FETAL FACE AND NECK | Radiology Key,” n.d.). However, by using VTK, the plane can be easily viewed as can be seen in fig. 4.3, and 4.5. Different combination of planes is needed as it allows the clinician to assess facial anomalies, and this usually requires the operator to be skilful. One of the advantages of using VTK for 3D reconstruction of the ultrasound images is it includes the visualisation of scanning planes that are physically impossible or very difficult to obtain in a 2D ultrasound scan such as viewing the foetal in an odd angle. Hence, it is much easier for new ob-gyn to assess any facial anomalies of the foetal. VTK also offers the demonstration of the surface of the foetal’s face and even a panoramic view of the entire foetal face as can be seen in fig. 4.6.

In expert hands, 2D ultrasound is extremely precise in the identification and categorisation of craniofacial malformations, however, when an inexperienced ob-gyn uses 2D ultrasound, the quality of the image as well as the time is taken for them to do the test may be affected. As what had been researched by Yang et al. (2010), beginners have problems in obtaining the required planes before measuring due to lack of familiarity with the anatomy of the foetal. With the help of 3D reconstruction in VTK, the foetal can be seen clearly, hence helping the new ob-gyn to easily measure the femur length, head

circumference, biparietal diameter, etc. as they can be familiar with the techniques of obtaining a standard plane for foetal biometric measurement (Yang et al., 2010).

Next, in VTK, a plane that can be moved at a different angle and positions can quickly help the clinicians to view the cross-section of the foetal's head, stomach, etc. as can be seen in fig. 4.6. Not only does this help in viewing any abnormalities in the foetal, but it would also help in viewing and checking whether there is a presence of tumour in the foetal. The availability of various display modes in VTK allows for the demonstration of both the normal and abnormal foetal anatomy in a controlled plane and rendered images from different angles. This not only allows for the foetal defects to be viewed from the most optimal perspective, but it would also help the physicians to help parents understand the severity of existing malformations quickly, or conversely ensuring them of the absence of abnormalities in the foetal.

The 3D reconstructed image of the foetal also allows for the clinicians or the ob-gyn to easily measure the height, depth, and width of the foetal as VTK allows for the measurements of length between two points (Yang et al., 2010). As can be seen in fig. 4.7, the femur length can easily be measured without having to imagine the position of the femur mentally. In fig. 4.7, the femur length is 5.8 cm, whereas in 3D Slicer is 2.97 cm. The difference between the two measurements obtained in VTK and 3D Slicer is due to an estimation that has been made in the pixel size of the image for VTK. The ability of VTK to measure the length of the distance between two points allows for a much more accurate and faster measurement to be conducted.

Both VTK and 3D Slicer is open-source software that is free and can easily reconstruct and visualise various medical images. Fig. 4.8 shows the 3D model constructed using volume rendering techniques. From fig. 4.8(a), the 3D rendered image was done using VTK appears to be looking more 3D due to the easy adjustments that can be done just by changing the values assigned for the shading in the algorithm. However, the 3D image

reconstructed using 3D Slicer has a much lighter view and the noise are more apparent than the 3D model reconstructed using VTK. Both VTK and 3D Slicer allows for the volume properties such as the colours, gradient opacity, the ambient, diffuse, and specular to be changed by just assigning a value. However, even when the properties are the same, both displayed a very different outcome. As can be seen in fig. 4.8, the 3D model from VTK appears to be better than the model constructed using 3D Slicer even when the volume properties are the same.

Next, both VTK and 3D Slicer allows for the measurement of length between two points. In VTK, the accuracy of the measurement depends on whether the assigned pixel size value is correct. However, in 3D Slicer, the value does not need to be assigned as the software automatically measures it as can be seen in fig. 4.12. In VTK, a DICOM file format is needed to know the pixel size of the image, but since the ultrasound image is in the JPEG format, the pixel size of the image is unknown. Due to this, 3D Slicer allows for a much easier measurement in comparison with VTK software as it can automatically compute the length or distance between two points.

One of the main differences between VTK and 3D Slicer is the techniques that can be used to render the 3D image. VTK can use surface rendering technique, however, 3D Slicer do not have that option as can be seen in fig. 4.13. It only allows for volume rendering of the ultrasound image.

Both VTK and 3D slicer does the job. However, the 3D image reconstructed by using VTK looks better than the image created in 3D Slicer. Both software are interactive, but it is better to use VTK when one is looking for the flexibility to change some of the properties such as adding layers, changing the colours and opacity of specific segments to differentiate between skin, bone, and tissues. For 3D Slicer, although it does allow for the user to change the opacity and colour, it does not allow for the addition of new layers, hence differentiating or changing the colours between skin, tissues, and bones are

challenging. In short, using VTK and 3D Slicer can reconstruct the 3D image accurately, but it should be noted that using VTK needs the user to know visualisation techniques and some programming expertise. On the other hand, 3D Slicer does not need for the operator to have experience or knowledge in programming. However, the user does need to understand some of the visualisation techniques that are provided in the software. Although VTK offers a very flexible software where users can create layers, change the colours of each layer, 3D Slicer is also a good software for people who would like just to get the job done without putting much effort into it.

In short, both VTK and 3D Slicer are excellent software for 3D Reconstruction. However, the reconstructed 3D image in VTK has a clearer and detail view of the foetal. Since it offers the users to change the colours and opacity of each layer, the user can easily differentiate between tissues, skins, and bones. The cross-sectional images created using the multiplanar technique also offers a much more accessible view of the foetal as the physician can easily adjust the angle of the plane in order to find the most optimal picture of the foetal for assessment purpose.

CHAPTER 6: CONCLUSION

In conclusion, visualisation toolkit (VTK) allows for 3D reconstruction of foetal ultrasound image by using surface rendering, volume rendering, and multiplanar rendering technique. The measurement of foetal biometry can efficiently be conducted using VTK as it allows for the cross-section views of the foetal to be observed. Not only does it help in easy measuring of the foetal, but it also allows beginners to be familiarised with the foetal anatomy as well as the planes. Both VTK and 3D Slicer did a great job in reconstructing the 3D model of the foetal ultrasound image. However, VTK allows a much more interactive and easier way of reconstructing the 3D image. This is because of the flexibility of the user to assign any values to the volume properties in order to improve the shading of the image, as well as turning on and off some features such as bone visibility. In conclusion, VTK software is a program that can do the 3D reconstruction of the foetal ultrasound images and allows for the measurement of length. It is a free open source software and is very helpful for doctors as it can help reduce the risk of misdiagnosis and the time taken to interpret as a less human cognitive ability is needed.

In the future, a comparison between VTK and 3D Slicer can be made by using a few different sets of ultrasound images that have a different number of layers. The 2D ultrasound image should be obtained using a machine that has a tracking sensor in order to maintain a constant scan rate as different scan rate and angle causes a low-quality image to be achieved. Next, image processing needs to be done before doing 3D reconstruction to help reduce noise, hence increasing the quality of the resultant 3D reconstructed image.

REFERENCES

- 3D volume rendering using vtk toolkit. | Quppler. (n.d.). Retrieved August 2, 2020, from <https://quppler.com/visualization-toolkit-vtk-volume-rendering-pipeline/>
- Abramowicz, J. S., Kossoff, G., Maršál, K., & Ter Haar, G. (2000). International Society of Ultrasound in Obstetrics and Gynecology (ISUOG). *Ultrasound in Obstetrics and Gynecology*, 16(6), 594–596. <https://doi.org/10.1046/j.1469-0705.2000.00296.x>
- Arakeri, M. P., & Reddy, G. R. M. (2013). An Effective and Efficient Approach to 3D Reconstruction and Quantification of Brain Tumor on Magnetic Resonance Images. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 6(3), 111–128. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.360.4229>
- Bandyopadhyay, O., Biswas, A., & Bhattacharya, B. B. (2016, January 1). Long-bone fracture detection in digital X-ray images based on digital-geometric techniques. *Computer Methods and Programs in Biomedicine*. Elsevier Ireland Ltd. <https://doi.org/10.1016/j.cmpb.2015.09.013>
- Brinkmann, D. H., & Kline, R. W. (1998). Automated seed localization from CT datasets of the prostate. *Medical Physics*, 25(9), 1667–1672. <https://doi.org/10.1118/1.598346>
- Çimen, S., Gooya, A., Grass, M., & Frangi, A. F. (2016, August 1). Reconstruction of coronary arteries from X-ray angiography: A review. *Medical Image Analysis*. Elsevier B.V. <https://doi.org/10.1016/j.media.2016.02.007>
- Convert Pixel (X) to Millimeter. (n.d.). Retrieved September 9, 2020, from <https://www.unitconverters.net/typography/pixel-x-to-millimeter.htm>
- Fenster, A., & Downey, D. B. (2000). Three-Dimensional Ultrasound Imaging. *Annual Review of Biomedical Engineering*, 2(1), 457–475. <https://doi.org/10.1146/annurev.bioeng.2.1.457>
- Fenster, A., & Downey, D. B. (2003, August 30). Three-dimensional ultrasound imaging and its use in quantifying organ and pathology volumes. *Analytical and Bioanalytical Chemistry*. Springer Verlag. <https://doi.org/10.1007/s00216-003-2169-6>
- Gebhard, R. E., Eubanks, T. N., & Meeks, R. (2015, October 1). Three-dimensional ultrasound imaging. *Current Opinion in Anaesthesiology*. Lippincott Williams and Wilkins. <https://doi.org/10.1097/ACO.0000000000000228>
- Gunay, M., Shim, M.-B., & Shimada, K. (2007). Cost- and time-effective three-

- dimensional bone-shape reconstruction from X-ray images. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 3(4), 323–335. <https://doi.org/10.1002/rcs.162>
- Hafizah, M., Kok, T. C., & Supriyanto, E. (2010). Development of 3D image reconstruction based on untracked 2D fetal phantom ultrasound images using VTK.
- Hsu, P. W., Prager, R. W., Gee, A. H., & Treece, G. M. (2009). Freehand 3D ultrasound calibration: A review. In *Advanced Imaging in Biology and Medicine: Technology, Software Environments, Applications* (pp. 47–84). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-68993-5_3
- Lai, K. W., Hum, Y. C., & Supriyanto, E. (2011). Surface rendering of three dimensional ultrasound images using VTK. *Journal of Scientific and Industrial Research*, 70(6), 421–426.
- Merz, E., & Pashaj, S. (2017). Advantages of 3D ultrasound in the assessment of fetal abnormalities. *Journal of Perinatal Medicine*, 45(6), 643–650. <https://doi.org/10.1515/jpm-2016-0379>
- Merz, E., & Welter, C. (2005). 2D and 3D ultrasound in the evaluation of normal and abnormal fetal anatomy in the second and third trimesters in a level III center. *Ultraschall in Der Medizin*, 26(1), 9–16. <https://doi.org/10.1055/s-2004-813947>
- Meyer-Baese, A., & Schmid, V. (2014). Introduction. In *Pattern Recognition and Signal Analysis in Medical Imaging* (pp. 1–20). Elsevier. <https://doi.org/10.1016/b978-0-12-409545-8.00001-7>
- Mohamed, F., & Vei Siang, C. (2019). A Survey on 3D Ultrasound Reconstruction Techniques. In *Artificial Intelligence - Applications in Medicine and Biology*. IntechOpen. <https://doi.org/10.5772/intechopen.81628>
- Moran, C. M., & Thomson, A. J. W. (2020). Preclinical Ultrasound Imaging—A Review of Techniques and Imaging Applications. *Frontiers in Physics*, 8(May). <https://doi.org/10.3389/fphy.2020.00124>
- Mozaffari, M. H., & Lee, W. S. (2017, October 1). Freehand 3-D Ultrasound Imaging: A Systematic Review. *Ultrasound in Medicine and Biology*. Elsevier USA. <https://doi.org/10.1016/j.ultrasmedbio.2017.06.009>
- Shahidi, R., & Kaufman, A. (1996). *Hardware Technologies Needed for Volumetric Rendering of Medical Data-sets by the Year 2000 John Flynn Surface Rendering versus Volume Rendering in Medical Imaging: Techniques and Applications*.

Ultrasound - 2D | imagingandultrasound. (n.d.). Retrieved July 2, 2020, from <https://www.imagingandultrasound.com/copy-of-ultrasound-3d>

ULTRASOUND EVALUATION OF THE FETAL FACE AND NECK | Radiology Key. (n.d.). Retrieved July 30, 2020, from <https://radiologykey.com/ultrasound-evaluation-of-the-fetal-face-and-neck/>

Ultrasound scan—fetal growth scan - Patient Information Brochures - Mater Group. (n.d.). Retrieved July 5, 2020, from <http://brochures.mater.org.au/brochures/mater-mothers-hospital/ultrasound-scan-fetal-growth-scan>

What is voxel? - Definition from WhatIs.com. (n.d.). Retrieved August 1, 2020, from <https://whatis.techtarget.com/definition/voxel>

Wheeler, G., Deng, S., Toussaint, N., Pushparajah, K., Schnabel, J. A., Simpson, J. M., & Gomez, A. (2018). Virtual interaction and visualisation of 3D medical imaging data with VTK and unity. *Healthcare Technology Letters*, 5(5), 148–153. <https://doi.org/10.1049/htl.2018.5064>

Yang, F., Leung, K. Y., Lee, Y. P., Chan, H. Y., & Tang, M. H. Y. (2010). Fetal biometry by an inexperienced operator using two- and three-dimensional ultrasound. *Ultrasound in Obstetrics and Gynecology*, 35(5), 566–571. <https://doi.org/10.1002/uog.7600>

Zhang, X., Zhang, K., Pan, Q., & Chang, J. (2019). Three-dimensional reconstruction of medical images based on 3D slicer. *Journal of Complexity in Health Sciences*, 2(1), 1–12. <https://doi.org/10.21595/chs.2019.20724>