FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Interactive Reformation of Fetal Ultrasound Data to a T-Position

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Biomedical Engineering

eingereicht von

Dipl.-Ing. Eric Mörth, BSc
Matrikelnummer 01325306

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Associate Prof. Dipl.-Ing. Dr.techn. Ivan Viola
Mitwirkung: Univ.Ass. Renata Raidou, MSc PhD

Wien, 5. März 2019

Eric Mörth

Ivan Viola

FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Interactive Reformation of Fetal Ultrasound Data to a T-Position

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Biomedical Engineering

by

Dipl.-Ing. Eric Mörth, BSc
Registration Number 01325306

to the Faculty of Informatics

at the TU Wien

Advisor: Associate Prof. Dipl.-Ing. Dr.techn. Ivan Viola
Assistance: Univ.Ass. Renata Raidou, MSc PhD

Vienna, 5th March, 2019

Eric Mörth

Ivan Viola

Erklärung zur Verfassung der Arbeit

Dipl.-Ing. Eric Mörth, BSc
Pfeilgasse 43, 1080 Vienna

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 5. März 2019

Eric Mörth

Danksagung

Eine Masterarbeit ist ein Projekt, das viel Unterstützung, Verständnis und auf jeden Fall viel Zeit benötigt. Die Entscheidung, ein zweites Masterstudium abzuschließen und eine zweite Masterarbeit zu schreiben, ist eine große Herausforderung. Ich möchte mich daher vor allem bei meinen Eltern, meiner Familie und bei meiner Freundin bedanken. Die mentale Unterstützung und die Gespräche über die gescheiterten Versuche und neuen Ansätze waren für mich essentiell, um die Arbeit zu gestalten. Studium, Arbeit und Freizeit unter einen Hut zu bringen, braucht viel Planung und Flexibilität. Ohne euch hätte ich diesen Weg nicht beschreiten können und es gäbe diese Masterarbeit nicht.

Vor allem möchte ich mich bei Associate Prof. Dipl.-Ing. Dr.techn. Ivan Viola bedanken, der mir die Chance ermöglicht hat, an diesem spannenden Thema arbeiten zu dürfen. Seine Expertise im Bereich des 3D Ultraschalls und der medizinischen Abläufe im klinischen Umfeld war essentiell für diese Arbeit. Da Prof. Viola im Laufe meiner Diplomarbeit eine Associate Professor Stelle an der King Abdullah University in Kaust begonnen hat, wurde meine Diplomarbeit in Wien von Univ.Ass. Renata Raidou, MSc PhD betreut. Ich bin ihr für alle Gedanken, Anregungen und ihr Feedback dankbar. Sie hat zum Gelingen dieser Masterarbeit entscheidend beigetragen.

Ein weiterer Dank gilt Herrn Professor Philipp Thurner, der in seiner Position als Dekan des Studiums Biomedical Engineering es erst ermöglich hat, dass ich die Masterarbeit am Institut für Computergrafik verfassen konnte. Weiters möchte ich mich für die Möglichkeit der flexiblen Gestaltung des Studiums und für das große Angebot von sowohl Pflicht- als auch Wahllehrveranstaltungen bedanken. Ebenfalls bedanken möchte ich mich bei meinen Studienkolleginnen und Studienkollegen, vor allem bei Stephan Reichel, die das Studium zu einer positiven Erfahrung gemacht haben.

Acknowledgements

Setting up a master thesis is a project which can only be performed with lots of support, understanding and of course lots of time. Deciding to start a second master study in parallel as well as finishing it by writing a second master thesis was a big challenge. Therefore, I would like to thank first of all my parents and my family as well my girlfriend. The mental support and the discussions about the various attempts that did not work out and about new approaches were essential for the creation of this master thesis. To combine work, study and life needs a lot of planning and flexibility. Without having my girlfriend and family as support on my way this master thesis and study would not have been possible, so thank you.

A great thank belongs to Associate Prof. Dipl.-Ing. Dr.techn. Ivan Viola who gave me the opportunity to work on this challenging and interesting topic. His expertise and knowledge about 3D ultrasound as well as the workflow of such an investigation and the clinical basics in this field where very important for this work. During the thesis Prof. Viola started a position as associate professor in Kaust at the King Abdullah University and therefore I had the honor to have Univ.Ass. MSc Phd Renata Raidou as my supervisor in Vienna. I would like to thank her very much for her support, her thoughts and her feedback in our weekly meetings and during the writing of this thesis.

A big thank goes to professor Philipp Thurner who in his position as dean of the biomedical engineering study enabled me to perform my master thesis at the Computer Graphics group of the Technical University Vienna. I would also like to appreciate the various offer of obligatory and electoral lectures which gave me the opportunity to study in a very flexible way. Finally, I would like to thank my study colleagues particularly Stephan Reichl who made the studying funny and enjoyable.

Kurzfassung

Dreidimensionale Ultraschall Untersuchungen werden heutzutage häufig bei fetalen Untersuchungen eingesetzt. Diese Untersuchungsmethode liefert genaue Informationen sowohl über die Haut und die Oberfläche des Fötus als auch über die inneren Organe. Pränatale Untersuchungen werden häufig eingesetzt, um die Entwicklung des Fötus zu untersuchen. Diese Untersuchungen sind sehr wichtig, um die gesunde Entwicklung des Fötus zu kontrollieren. Die Analyse der Daten basiert auf einer zweidimensionalen Schicht-Ansicht, um die Abmessungen durchzuführen, die für die Berechnung des Volumens und des Gewichtes des Fötus notwendig sind. Die Ergebnisse sind jedoch auch immer abhängig von der Person, die die Messungen durchführt, da es oft auch wenig eindeutige Daten gibt. Die Messungen und Berechnungen sind jedoch für die Kontrolle der Entwicklung des Fötus und auch für die Geburtsvorbereitung wichtig.

Das Ultraschall Bildgebungsverfahren ist oft von Artefakten wie Punkten, Rauschen oder blockierter Sicht betroffen. Diese Störungen treten auf, weil das Verfahren akustischer Natur ist. 2D-Schichten als Basis für verschiedene Messungen sind nicht optimal, weil die Messungen von der Projektion abhängt und nur korrekt ist, wenn genau die richtige Ebene gewählt wurde, um diese durchzuführen. Eine Analyse der Daten in 3D wäre sinnvoller, da der Beobachter bzw. die Beobachterin einen besseren Überblick hätte und besser zwischen Artefakten und „echten“Daten unterscheiden könnte. Gewisse Messungen des Fötus wie zum Beispiel die Länge von Kopf bis Rumpf oder die Länge des femoralen Knochens können mit standardisierten Tabellen verglichen werden, um die Entwicklung des Fötus zu überprüfen.

Standardisierung ist in der Medizin weit verbreitet und hat den Zweck, die Vergleichbarkeit der Daten verschiedener Patienten zu ermöglichen. Deshalb wird hier eine Methode vorgestellt, die 3D Ultraschalldaten einer fötalen Untersuchung standardisiert. Den Fötus in eine standardisierte Pose zu bringen, ermöglicht die Durchführung von automatischen Messungen und ermöglicht auch die Einführung von neuen Messungen, die unter anderem das Volumen von bestimmten Körperregionen beinhalten könnte. Eine standardisierte Pose des Fötus würde es auch ermöglichen, Föten miteinander zu vergleichen oder die Daten eines Fötus im Verlauf der Schwangerschaft zu beobachten.

Die neue Methode besteht aus sechs Schritten: Laden der Daten, Vorverarbeitung, „Rigging“ des Modells, Gewichtung, die eigentliche Transformation oder auch „Vitruvian Baby“genannt und zum Schluss die Analyse. Es wurde versucht so viele Schritte des Workflows zu automatisieren, manche Schritte jedoch sind manuell. Vollkommen automatisiert ist die Gewichtung des Modells und die Transformation. Die Vorverarbeitung und das „Rigging“des Modells benötigt manuellen Input des Users. Beim „Rigging“wird ein abstrahiertes Skelett in das Modell eingesetzt. Dieses wird dazu verwendet, den Fötus zu beschreiben und dient als Grundlage für die Transformation.

Die Leistung unseres neuen Verfahrens wurde getestet, indem ein Phantom-Datensatz eines Mannes verwendet wurde. Das Phantom ist bereits in der T-Pose gegeben und gilt daher auch als Vergleichsbasis für das Ergebnis unserer Methode. Das Phantom wurde zuerst in sieben verschiedene Fötus Posen gebracht. Der Vergleich des Ergebnisses unserer Methode und der Ziel T-Pose ergab eine Übereinstimmung auf Voxel Ebene von 79.02% im Durchschnitt. Die Ähnlichkeit der Messung von Kopf bis Fuß und von den Fingern der linken Hand zu den Fingern der rechten Hand wurde ebenfalls gemessen. VOn Kopf bis Fuß ergab sich eine Genauigkeit von 94.05% im Durchschnitt und bei der Messung von Finger zu Finger von 91.08% im Durchschnitt. Der manuelle Schritt des „Riggings“ dauerte in Durchschnitt sieben Minuten. Die Methode wurde auch mit einem Fötus-Modell und einem Phantom-Datensatz einer dreidimensionalen Ultraschall Untersuchung getestet und die Ergebnisse sind vielversprechend.

Abstract

Three dimensional ultrasound images are commonly used in prenatal screening. The acquisition delivers detailed information about the skin as well as the inner organs of the fetus. Prenatal screenings in terms of growth analysis are very important to support a healthy development of the fetus. The analysis of this data involves viewing of two dimensional (2D) slices in order to take measurements or calculate the volume and weight of the fetus. These steps involve manual investigation and are dependent on the skills of the person who performs them. These measurements and calculations are very important to analyze the development of the fetus and for the birth preparation.

Ultrasound imaging is affected by artifacts like speckles, noise and also of structures obstructing the regions of interest. These artifacts occur because the imaging technique is using sound waves and their echo to create images. 2D slices as used as basis for the measurement of the fetus therefore might not be the best solution. Analyzing the data in a three dimensional (3D) way would enable the viewer to have a better overview and to better distinguish between artifacts and the real data of the fetus. The growth of a fetus can be analysed by comparing standardized measurements like the crown foot length, the femur length or the derived head circumference as well as the abdominal circumference.

Standardization is well known in many fields of medicine and is used to enable comparability between investigations of the same patient or between patients. Therefore we introduce a standardized way of analyzing 3D ultrasound images of fetuses. Bringing the fetus in a standardized position would enable automatized measurements by the machine and there could also be new measurements applied like the volume of specific body parts. A standardized pose would also provide possibilities to compare the results of different measurements of one fetus as well as the measurements of different fetuses.

The novel method consists of six steps, namely the loading of the data, the preprocessing, the rigging of the model, the weighting of the data, the actual transformation called the "Vitruvian Baby" and at the end the analysis of the result. We tried to automatize the workflow as far as possible resulting in some manual tasks and some automatic ones. The loading of the data works with standard medical image formats and the preprocessing involves some interaction in order to get rid of the ultrasound induced

artifacts. Transforming data into a specific position is a complex task which might involve a manual processing steps. In the method presented in this work one step of the transformation namely the rigging of the model, where a skeleton is placed in the data, is performed manually. The weighting as well as the transformation although are performed completely automatically resulting in a T-pose representation of the data.

We analysed the performance of our novel approach in several ways. We first use a phantom model which has been used as a reference already presented in a T-pose. After using seven different fetus poses of the model as input the result was an average of 79,02% voxel overlapping between the output of the method and the goal T-pose. When having a look at the similarity of the finger to finger span and the head to toe measurement we considered a value of 91,08% and 94,05% in average. The time needed for the most complex manual task was in average seven minutes. After using a phantom model of a man, we also assessed the performance of the method using a computer model of a fetus and a phantom model of a 3D ultrasound investigation. The results also look very promising.

Contents

CHAPTER 1

Introduction and motivation

Medical fetus diagnostics involves in most cases ultrasound imaging and visualization [VBS$^+$13]. The low costs and the fact that there is no radiation involved leads to a growing interest in this medical recording process and the involved visualization techniques [HZ17]. 3D ultrasound has become a major field of research since the early 1980s [CSC10]. Nowadays also four dimensional (4D) ultrasound imaging is possible where the 3D images are recorded in real time and represented as volumes [YCSV07].

1.1 Ultrasound imaging and characteristics

Ultrasound imaging uses the principle of producing short ultrasonic bursts namely in the region of 1 Mega Hz to 40 MHz and sends them into the specimen which should be examined [KvdMF$^+$17]. The echo is afterwards received. The timing of the echo encodes where the tissue is located and the strength of the signal received encodes information about the interface between tissues, from with the acoustic wave is reflected [KvdMF$^+$17]. One might already see one limitation of the technique namely the tissue which should be examined can be blocked by some other structures, the ultrasound imaging will not be able to record it. The ultrasound imaging uses a so called transducer to produce and perceive the ultrasound signal. The signal is created by piezoelectric crystals [KvdMF$^+$17]. If those are aligned in a grid in the transducer 3D images are produced. If the signal is also recorded over time it is called 4D ultrasound imaging [KvdMF$^+$17, YCSV07].

Yagel et al. [YCSV07] state that 3D and 4D ultrasound scanning devices are available in many centers where fetal diagnostics is performed. The diagnostics of the fetus growth is one of the major fields where 2D, 3D and 4D ultrasound techniques are used [MTDL05]. But ultrasound images are not always the easiest ones to analyze due to e.g. artefacts and low resolution [NPH$^+$03, SBS$^+$17]. According to Nelson et al. [NPH$^+$03] artefacts in ultrasound have to be classified by their origin. They can origin from the B-mode or from the color/power Doppler effect [NPH+03]. One also has to differentiate between artefacts in 2D and their possible representation in 3D and artefacts that only occur in 3D. Artefacts which are originated from the B-mode are related to resolution, propagation of the wave in the tissue and by attenuation [NPH+03]. The resolution in ultrasound imaging is determined by the pulse length, the pulse width and the focusing of the wave. Those parameters are limited. Resolution artefacts cause distortions of the perceived shape and size of objects [NPH+03]. Speckles in the recorded image are caused by interference of the tissue.

Doppler caused artefacts are mainly important when analyzing blood flow [NPH+03]. If there is for example a very large flow, the Doppler sonography result may show vessels growing into each other [NPH+03]. More important in terms of this thesis are the 3D specific ultrasound artefacts. Those are mainly originated of movement of the specimen [NPH+03]. Other reasons for artefacts in the output image are for example caused by rendering where the threshold is e.g. too low or also by shading, which highlights problems that are not present [NPH+03]. A comparison of images with too low threshold and perfect threshold is presented by Figure 1.1. Another issue might also be caused by editing where the operator eliminates important structures by accident [NPH+03].

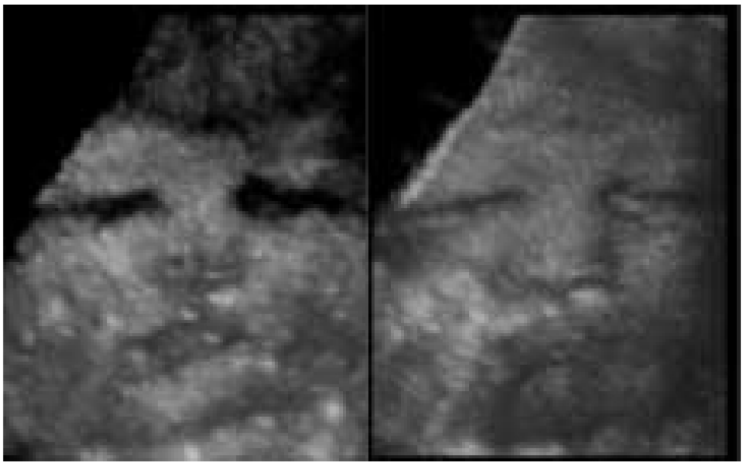


Figure 1.1: The eyes of the fetus appear black because of a two high threshold. The operator has set the value to exclude the data of the eyes as if they would be unimportant. The right image is the result of a perfect threshold [NPH+03].

1.2 Analysis of the data

After discussing ultrasound and the various amount of artefacts we would like to focus on the analysis of the data. In a clinical environment the analysis of 2D ultrasound images is common. Loughna et al. [LCEC09] write about the fetal size measurements and the dating of the pregnancy and introduced recommended charts that shall be used in clinical obstetric practice. The overall growth is an important measurement for dating the pregnancy and for controlling the development of the fetus. Important variables are for example the crown-rump length (CRL) and the head circumference (HC) [LCEC09]. There are also other measurements used to predict the fetal growth using circumferences like the head circumference or the abdominal circumference. Different approaches may be used depending on multiple measurements [LCEC09]. According to Schild et al. [SFH00] another interesting measurement is also the femur length because in combination with the HC it may be used to calculate the predicted weight of the fetus.

Measuring the growth of the fetal organs in addition to the overall development is a very important factor for the healthy development of the fetus [MTDL05]. Ultrasound is widely used to measure e.g. the growth of the fetal lungs [MTDL05]. Having a look at the general growth development of the fetus it is very important for the early detection of abnormalities in development or other complications during the pregnancy [WD10].

Standardization and normalization is often used in medicine to create comparability [MMNG15, NUX02, LDB95, KQ03]. Nyul et al. [NUX02] for example write about standardization in magnetic resonance imaging (MRI) where it is important to normalize the value range because it differs from patient to patient. Miao et al. [MMNG15] introduced a way to standardize the circle of Willis, which is a circulatory anastomosis that supplies blood to the brain of humans. We think that standardization is also needed in ultrasound based prenatal diagnosis. Comparability of inner and inter specimen images is one of the goals of the novel approach we introduce with this thesis.

1.3 Problem definition and research question

Analyzing volumetric data in a 2D slice view is likely to be inaccurate, because the plane where the measurement is taken has to be perfectly aligned with e.g. the femur bone. if the plane is not aligned perfectly the user will only see a projection of the bone and the measurement is likely to be too short. Circumference measurements like described by Loughna et al. [LCEC09] are also error prone if they are only performed by having a look at the length and the width of the structure and estimating how the circumference might look like. It might be that there are some irregular structures e.g. at the head or at the abdomen which might not be considered. Abnormalities are very important when examining fetal growth [LCEC09].

Another challenge gynecologists have to face are the numerous possibilities of fetal poses [FH89]. The pose of the fetus during the ultrasound investigation can change a lot, especially between different women and also during the different stages of pregnancy. A sketch of the possibilities is shown in Figure 1.2. The images and also the volumetric data may be saved but cannot be compared directly. As Loughna et al. [LCEC09] state, only standardized quantitative measurement like lengths and circumferences may be compared to standardized tables.

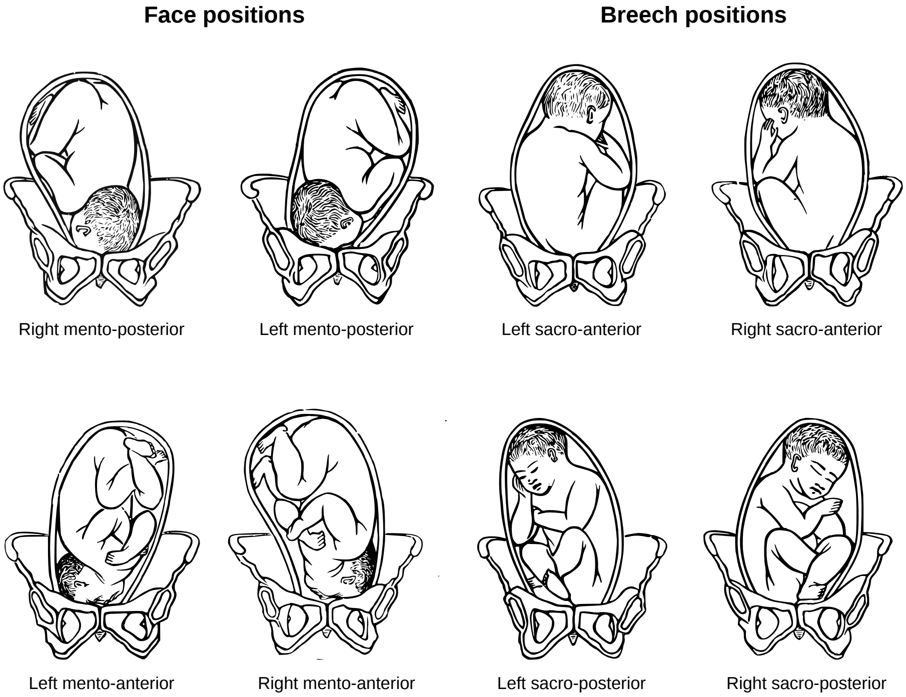


Figure 1.2: Possible fetal poses sketched by Haultain and Ferguson [FH89].

Standardization is for example introduced by Miao et al. in terms of the circle of Willis [MMNG15] and also in terms of the cardiovascular system by using a bulls eye plot [ADCB$^+$15, OKG$^+$06] and also in terms of value standardization in MRI imaging [NUX02]. Therefore we propose standardization in terms of prenatal 3D ultrasound investigations. We want to investigate if transforming the fetus in 3D to a standardized position can be helpful to analyze the volumetric data of one fetus over time or between different fetuses.

When thinking of a transformation of the result of a 3D ultrasound investigation a number of questions may arise. The overall question is if we can provide a transformation of the fetus in 3D to a standardized position, to analyze the fetal development.

1.4 Overview of the aimed solution

Having a look at the clinical procedure when performing fetus analysis we identified options where our solution might be an improvement. The first step is to acquire the data of the fetus. This step may be one of the most important ones, because it determines the completeness and the quality of the fetal data. Therefore this step should already be guided and in a way that the clinical personal is able to perceive the data which is already acquired and where some additional data may be useful. In order to be able to capture and visualize a fetus during the ultrasound session, the compounding feature introduced by Viola et al. [VBS$^+$13] may be of a great interest.

After the recording of the data the processing of it is the next step. The idea is that the fetus is processed in a way that it can be presented in a T-position. This position enables different kinds of measurements and enables a clear view of the limbs as well as the head. Transforming of given data into a specific position requires somehow knowledge of the given data in order to perform the task [SS04]. This knowledge might be included in the dataset given or may be the result of some interaction with the user. In order to somehow anatomically correctly model a fetus into a new formation the skeleton or an abstract version of the skeleton may be used. At some point one might consider the interaction with the operator of the program useful and fruitful and this might be one of those. The process of interactively reforming the fetus data into a given position requires first the integration of a so called armature into the data. The armature describes the joints and bones of the skeleton in an abstract way. Some points may be moved into the data automatically but it might be a better solution to ask the operator of the system to define some key points and let the machine describe the rest of the data.

The transformation of the fetus has to be done in a way that the result is a fetus with his or her arms in a horizontal line and the legs shall be straight and in parallel. One may imagine the famous image of Leonardo da Vinci named the Vitruvian Man where a man is drawn in a perfect T-pose. Figure 1.3 shows the famous picture. The problem at this step is to find out how the different parts have to be transformed in order to bring them into that special shape.

After having the armature embedded in the data the weighting has to be done. Weighting means that the program has to calculate the affinity of voxels to segments of the armature. This step can be done completely automatically. After the volume data has been linked to the armature the transformation can be done. The transformation analyzes the position of the armature and moves each part of the limbs of the fetus against others in respect to the joints. The angles which have to be used when applying the rotations in 3D space may be calculated and applied automatically. The result should be a representation of the acquired fetus data showing the whole fetus at a glance in a T-position.

The end results may also include the standardized measurements automatically taken by the machine. The introduction of this system may also lead to the development of new measurements which consider the whole span from finger to finger or the size of the fetus from head to toe. The representation of a fetus in a standardized way also enables the data to be comparable in means of time or inter specimen differences.

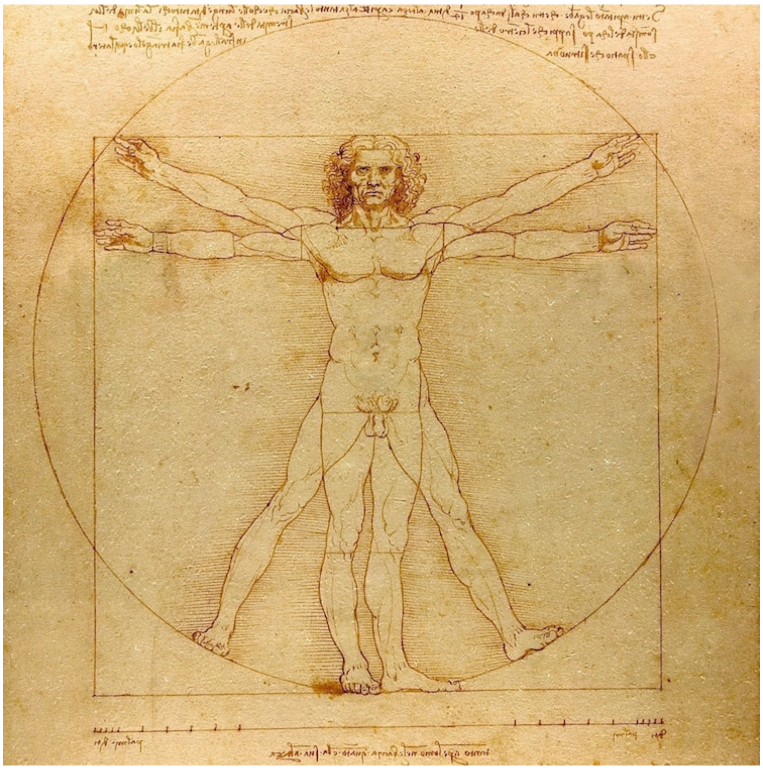


Figure 1.3: The vitruvian man by Leonardo da Vinci [Vin90].

The workflow of the method introduced in this thesis is visualized in Figure 1.4. Methods describing how to load medical data are discussed in Section 4.2. The preprocessing steps are presented in Section 4.3 and options of how to perform the rigging of the model are shown in Section 4.4. The weighting is discussed in Section 4.5. The methods used in the Vitruvian Baby step are presented in Section 4.6.2 and the implementation details about this step in the workflow are discussed in Section 5.3. The last step namely the analysis of the data is shown in Section 4.7.

1.5 Requirements

In this section we would like to discuss the requirements of the method. We would like to introduce domain and technical requirements of the system.

1.5.1 Domain requirements

How many and which user interaction will be needed in order to carry out the task? Do the users need special domain knowledge or a special training in order to work with the system? How much time will the processing and especially the user interaction need? Will the results be comparable? How will the analysis be carried out, will there be an automatic

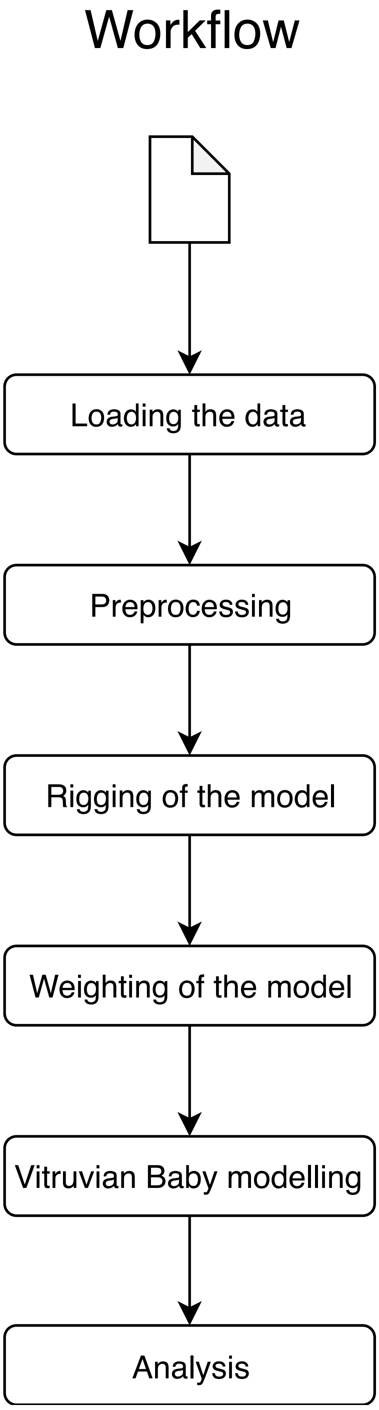


Figure 1.4: The workflow of the method introduced in this thesis.

report given by the system which can be compared with standardized measurements? How good is the performance of the transformation, will there be deformations included and how much percent deviation is possible in the provided measurements? Can the system be set up in a clinical environment and will it help the gynecologist to perform the analysis more efficiently? Are there any artefacts of the method which may lead to a dangerous misinterpretation of the results? We are sure that there are many more questions which may come to the mind of clinical personnel but we will focus on those presented in this paragraph.

1.5.2 Technical requirements

Is it technically possible to perform a transformation of volumetric data into a given pose? Is it possible to do it fully automatically? How will the solution perform and how much computer resources will it need? Which parts of the problem are already solved and which have to be newly developed? Where are the limits of volume resolution? How much time will it take to perform the transformation?

1.5.3 Data format

Loading the volumetric data of the fetal ultrasound analysis into the program shall be rather easy and the system should support well-known file formats used in medicine like the file formats used by the Insight Segmentation and Registration Toolkit (ITK) [LCIB13]. Another approach would be to use the file format called Digital Imaging and Communications in Medicine (DICOM) [MA19] which is commonly used in clinical application or generally when working with medical data.

1.5.4 Preprocessing

Preprocessing steps are deployed in order to get rid of the ultrasound specific artifacts discussed in section 1.1. The processing may include filtering to get rid of artefacts induced by for example the mother fluid or interactions with the tissue which the sound wave has to cross in order to perceive the fetus. In ultrasound images the data which represent the fetus normally has a specific range of values, therefore a threshold may be applied. If the further processing steps also assume or require that the data only includes the fetus and nothing else a largest connected component analysis should also be performed. This step searches, like the name says the largest connected component in a 3D volumetric dataset which should be the fetus. The rest of the image may be declared as artefacts or not relevant and can be neglected. When thinking of the analysis of twins first a region of interest might be selected to split the data before starting the method in order to work with the images of each fetus separately.

1.5.5 User interfaces

The solution shall provide an interface where the user is able to interact with the system. The system shall provide an "easy to use" graphical desktop where the interaction is carried out. A pipeline that guides throught the different steps of the transformation would be appreciated. In general the aim is to minimize the input needed from the end-user and shift the focus on automatic processing of the data.

One step where interaction is very likely to be used is the rigging of the data. Finding a skeleton or incorporating a predefined armature into the data may be out of scope of this thesis and therefore user interaction is thinkable. The integration of the armature into the data should not take too much time and has to be very easy to carry out. The armature or skeleton is a preset where all the joints and the bones are already defined and only have to be put at the right spot inside of the volume. This step can be done by using a simple drag and drop interaction in a 3D view of the model. The user therefore has to be familiar with the navigation in an volume explorer.

1.5.6 Transformation and deformation

The main step of the processing steps introduced in this thesis namely the T-transformation should be performed completely automatically and the results should also be exportable for further visualization and analysis. Transforming the data should not require any further input or interaction with the user. The data and the armature should be the only input for the module handling the transforming process. The end result although should again be inspectable and there should be some possibilities to automatically measure the fetus. Automatically displaying the main measurements like the span from fingertip to fingertip and from head to toe would also be possible.

Transformation of volumetric data often goes along with deformation caused by it. We think that the method described in this thesis is also likely to cause deformations to the prenatal investigation data. The requirement should be that the deformation causes only minimal distortion to the measurement of head to toe and finger to finger span distance. The measurement of the head circumference as well as the abdominal circumference shall also be correct after the transformation.

1.6 Contribution of the work

After analyzing the related work presented in Chapter 3 and requirement analysis and considerations in Chapter 4 we present our solution in Chapter 5. Our contribution is a new and novel approach to transform 3D ultrasound data of a fetus to a standardized T-pose. We designed and implemented the workflow to do so, and provide an interface to explore the result as well as the intermediate results of the method. The presented method is designed to work in a semi-automatic way and guides the user to perform the steps where manual interaction is needed. We investigated the degree of automation to increase the reproducibility and the performance of the approach.

CHAPTER 2

Clinical background

In this section the clinical background of the data and the processing steps are discussed. The source of the data and the different forms of data acquisition are especially important in order to understand why different processing steps and enhancements have to be performed. Linking the visualization and the processing of the medical data to their origin and thinking about possible applications is also important for understanding the overall concept.

2.1 Related data and data acquisition

First we would like to discuss the imaging modality ultrasound. The basics of ultrasound as an imaging source can be found in recent textbooks on medical visualization [PB14].Those have to be within a frequency range of more than 20kHz as well as between 1 and 15 MHz. The main part of the ultrasound equipment is the transducer. The transducer produces the sound waves as well as records the reflected ones. Kruizinga et al. [KvdMF^{+}17] mention in their paper about compressive 3D ultrasound imaging using a single sensor that those transducers are often equipped with a huge amount of tiny sensors which are able to create ultrasonic bursts. Those are reflected by the tissue and the time delay between recording and sending them into the tissue encodes the location of the tissue reflection. The density is encoded in the strength of the signal recorded. Preim and Botha [PB14] state that there are some problems using this image acquisition technique, namely that the results are affected by noise and speckle. Tomographic images would lead to a higher contrast due to the higher intensity of the waves used [PB14]. The operator should have a very good understanding of the anatomy that is examined in order to prevent misinterpretation of the data in case of occlusion.

Volumetric ultrasound scanner use an array of sensors on a transducer. According to Ivancevich et al. [IDTS06] the sensor is able to perceive a 65° times 65° pyramidal volume. Figure 2.1 represents how the volume looks like. The data is aligned at this grid and has to be processed in order to align it. The pyramidal grid is one reason, that makes it complex to stitch the volumetric data given by the ultrasound imaging together in order to generate a compound volume.

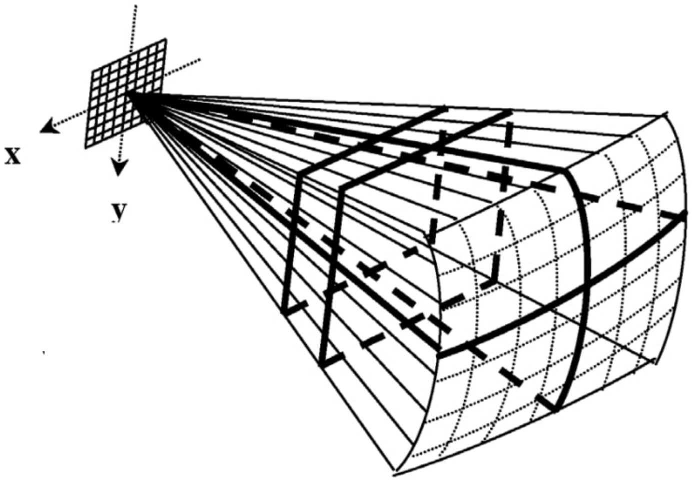


Figure 2.1: Imaging pyramid with 65° times 65° producing a volumetric scan result.

According to Huang and Zeng [HZ17] real-time 3D ultrasound imaging is a very well accepted technique in medicine, because it provides interactive feedback for the clinicians. Viola et al. [VBS+13] state that there are different issues when working with ultrasound data like noise or other visual artefacts. It is very important to have a precise navigation and visual feedback in order to acquire meaningful data with a high level of details [VBS+13]. One problem in 3D ultrasound is that the perceivable volume might not be large enough to record the whole range of interest. This problem occurs in many cases for example when examining inner organs like the liver but it is also a very prominent issue when working with fetal data [VBS+13, LDB95].

The data delivered by 3D ultrasound can be seen as a cloud of data points aligned at a 3D grid. In case of a fetal examination they can represent various things like mother fluid, artefacts from the tissues which the signal of the ultrasound has to travel through or also organic tissue which is floating in the mother fluid as well as of course the fetus. The data points have to be characterized and it has to be clear which of them really belong to the fetus, obviously the object of interest. Having a look at the anatomy of the woman the data acquisition may be somehow tricky. The belly is very well exposed and it leads to beautiful pictures from one side but having a look at the other side of the fetus may be impossible. The ultrasound waves do not travel very well through hard tissue like bones and therefore the examination of the fetus through the back of a woman is not helpful. A schematic explanation of how a fetal ultrasound investigation is carried out is depicted in Figure 2.2.

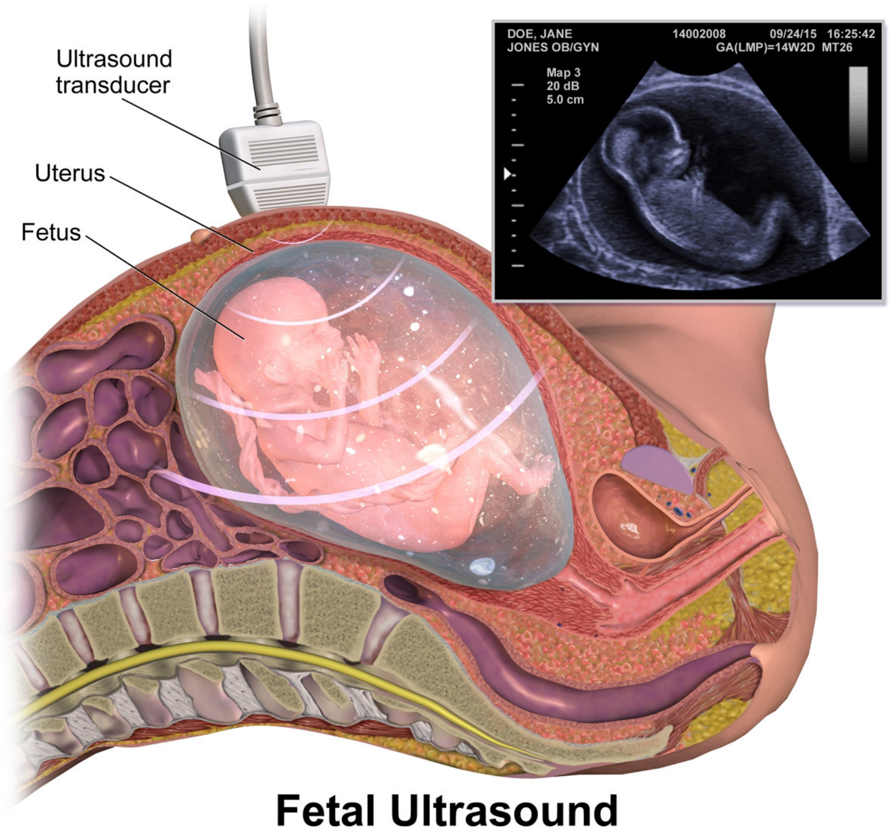


Figure 2.2: Schematic description of how a fetal ultrasound investigation is carried out including anatomical details of the pregnant woman [Bla15].

2.2 Analysis of the data

2D, 3D and also 4D ultrasound investigations are techniques which are without a doubt very important for the analysis of a fetus. Gathering insight of the fetal surface as well as the organs and the cardiovascular system are essential for supporting a healthy development.

2.2.1 Organ analysis

The development of the inner organs is a very important aspect in the early development of a human being. Many problems during the growth phase of the essential organs like the heart or the lungs probably origin from influences of the outside. Therefore in case of growth issues expectant mothers could be advised on how to avoid bad influences.

Lung growth

Moeglin et al. [MTDL05] state that various ultrasound techniques may be used to identify fetal lung growth. They write about using 2D and 3D ultrasound based approaches as well as MRI based techniques. The MRI based measurements are able to encounter more detailed analysis because they able to overcome the limitations of ultrasound based imaging but the downside is that the costs are higher, the patient compliance is lower and it is highly affected by the fetal movement [MTDL05, BCPC^{+}15].

Fetal lung volume might be measured using the Virtual Organ Computer-aided Analysis (VOCAL) software by General Electric Medical Systems KretzTechnik which is able to produce a 3D model of the lung. A representation of the graphical output of the software can be seen in Figure 2.3 [MTDL05].

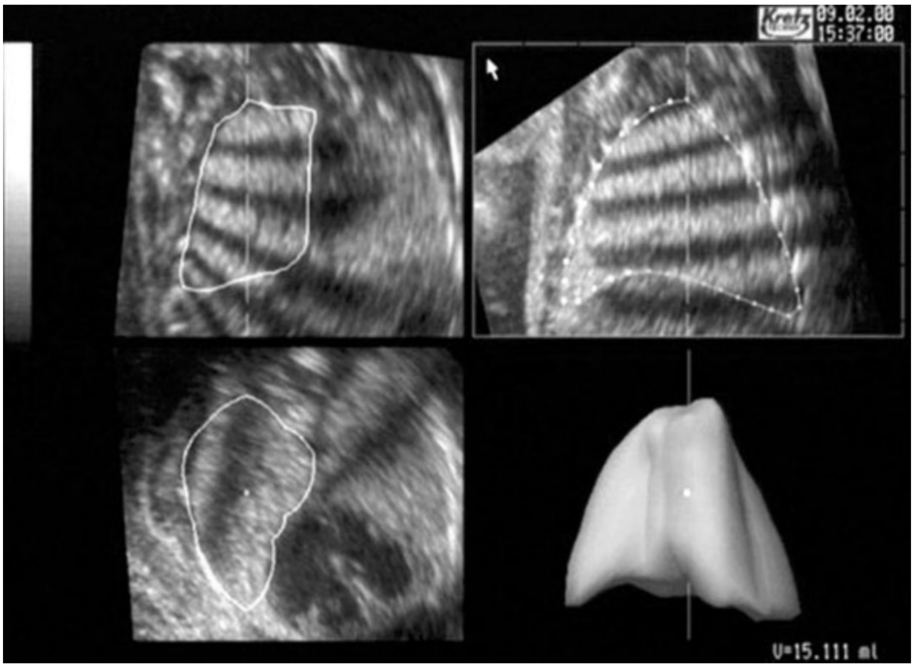


Figure 2.3: Output of the VOCAL software using an rotation technique to estimate the fetal lung volumetry. The lung volume is estimated by rotating it around the vertical axis. In the lower right corner a produced 3D model of the lung can be seen [MTDL05].

Cardiovascular system

The fetal cardiovascular system is also subject of analysis [YCSV07, Bas11, DZL13]. It suffers from congenital pathology most often [DZL13]. 3D ultrasound and modern visualization techniques can for example be used in order to examine and visualize the inter ventricular septum [YCSV07]. Figure 2.4 represents the ultrasound image and the surface rendering of the septum.

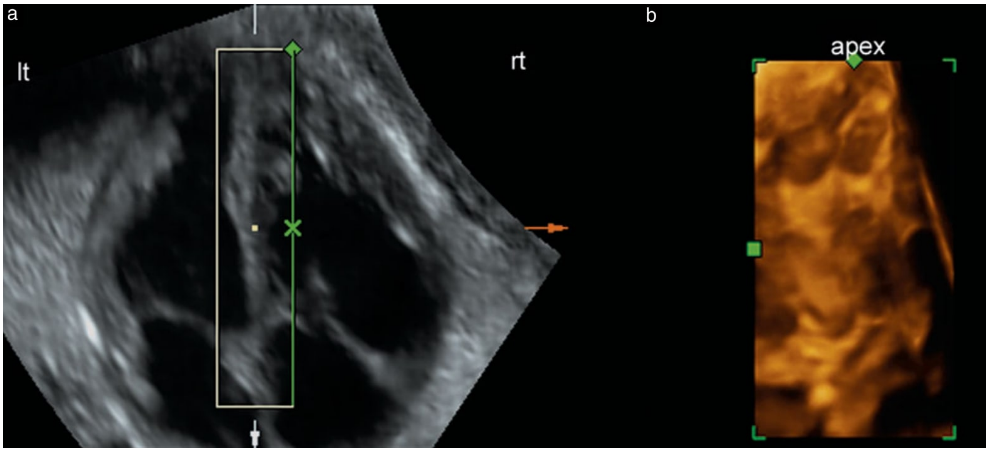


Figure 2.4: Left side of the image shows the ultrasound image of the heart and the interventricular septum is inside of the green box. The surface of the septum is visualized on the right side which might help doctors to examine for abnormalities [YCSV07].

Another important measurement at the cardiovascular level is to measure the cardiac output volume using Doppler evaluation [Bas11]. Baschat shows in his paper that this technique may be used to examine the flow through different vessels in a fetus in order to find out if there is any occlusion or abnormality [Bas11]. The author states that it is an essential tool for perinatal specialists for the generation of clinical diagnoses. Figure 2.5 shows an example of how the output of such a Doppler evaluation may look like.

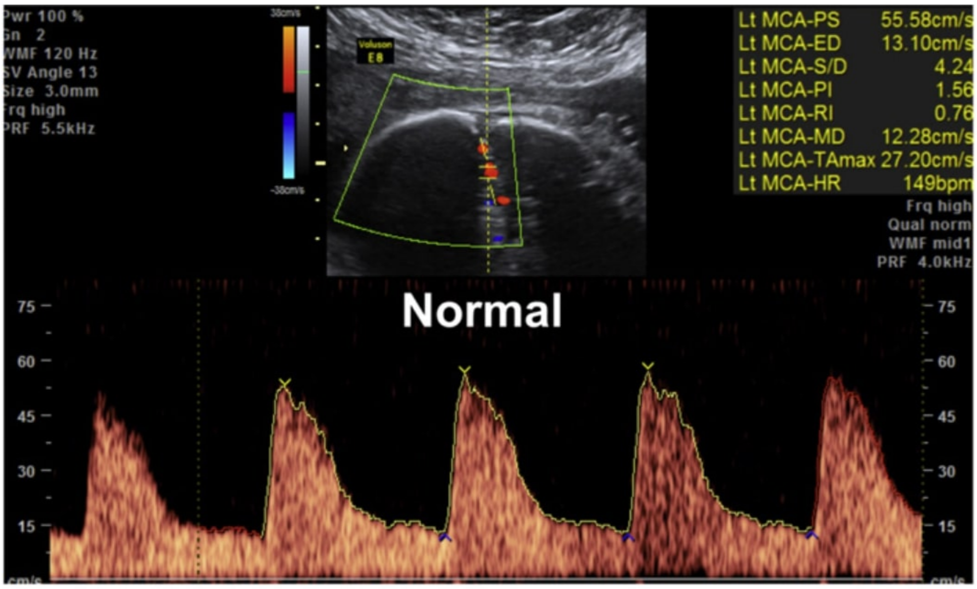


Figure 2.5: Doppler signal of the middle cerebral artery of a fetus [Bas11]. https://www.overleaf.com/project/5c7edf5a9c804c4085911acb

2.2.2 Facial development

Another very important aspect which can be examined using 3D ultrasound is the development of the face of the fetus. The information about facial deformations is not only important for obstetricians or paediatricians the surface visualization of the given data is also important when telling the parents [LDB95]. It can be somehow tough to explain them with words that their child is going to have deformations in the face but showing them makes it a lot easier to perceive. An example for such a facial surface visualization is depicted in Figure 2.6.

2.2.3 Fetal growth analysis

Fetal size measurement more general growth analysis is a well known topic and many different scientists have written about it [HDHP84, SFH00, LCEC09, WD10]. The size and the weight of the fetus is of a special interest for the clinicians in the birth preparation. Knowing what to expect can be very important during the birth. In earlier times the growth analysis has only taken the 2D images into account and the measurements where for example the following: CRL or HC. Those are very well defined and Loughna et al. [LCEC09] perfectly illustrate how to measure them which may be seen in Figure 2.7 and Figure 2.8. The head circumference is measured using the outer to outer biparietal diameter and the occipital-frontal diameter [LCEC09].

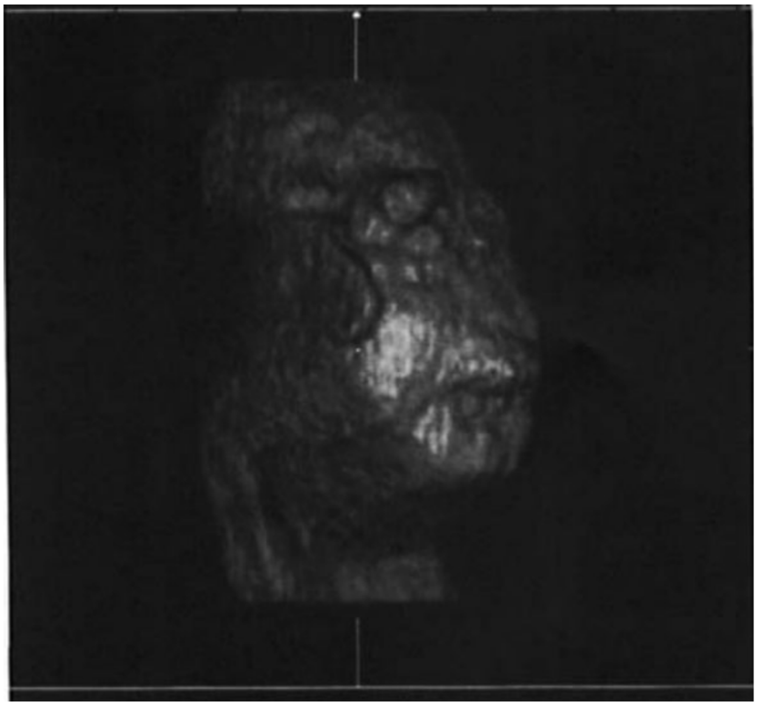


Figure 2.6: Ultrasound surface visualization of a male fetus in the 30th week of pregnancy. The face does not show any regular facial structure [LDB95].

Loughna et al. [LCEC09] state that the biparietal diameter is especially important to be analysed separately from the head circumference, because it indicates irregular shapes of the head. A dolichocephalic head which is longer and therefore less broad then a regular head shape would have a head circumference that is in the normal range but the biparietal diameter would be larger than normal.

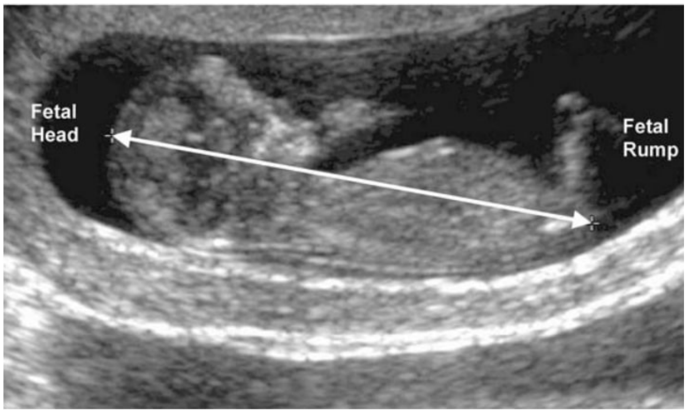


Figure 2.7: Crown-rump measurement of a fetus [LCEC09].

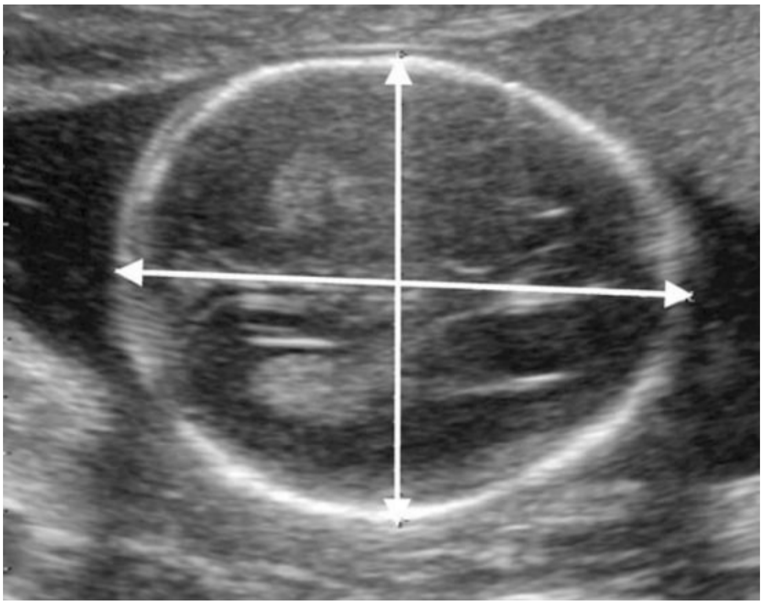


Figure 2.8: Head circumference measurement [LCEC09].

Another important measurement is the femur length. Hadlock et al. [HDHP84] state that the head circumference in combination with the femur length are a strong predictor of the fetal age. The femur length should be measured in an image where the ultrasound beam is as close to 90° as possible. The view should not be obscured and the bone should be very well visible as in Figure 2.10. The abdominal circumference is also an important aspect of the fetal growth. Loughna describes the measurement by taking the transverse abdominal diameter (TAD) and the posterior abdominal diameter (APAD) [LCEC09]. Those measurements are combined to form the abdominal circumference. The measurements are visualized in Figure 2.9. Knowledge about the age is essential in the fetal growth analysis because only with this information the growth is comparable to standardized tables like stated in [LCEC09].

3D ultrasound also plays a role in fetal growth analysis. Schild et al. [SFH00] found out that using the 3D data produced by the 3D ultrasound, weight estimation can be enhanced. They also use the influence of soft tissue in order to find out more about the constitution of the fetus and the effects of this on the weight [SFH00].

The growth analysis is done over the whole pregnancy of the woman starting with the 13th week and ending with the 42nd [LCEC09]. In this period of time the growth is calculated using the formulas presented in the paper of Loughna et al. [LCEC09] and than compared with standardized tables. The problem which may arise here is that those measurements take some time and have to be performed for every investigation. A

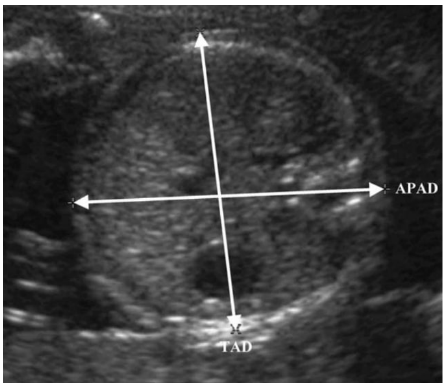


Figure 2.9: Measurement of the abdominal circumference [LCEC09].

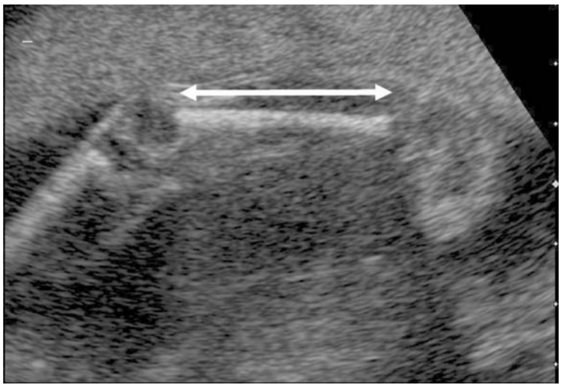


Figure 2.10: Measurement of the femur length [LCEC09].

standardized position or pose of the fetus may be helpful to not only visualize the growth of the fetus in the different stages of the pregnancy but also analytically calculate it.

CHAPTER 3

Related work

In this section prior related work will be presented and summarized. We ordered the discussion by the order of the workflow.

First standardization in visualization as well as normalization is discussed [BUB10, NUX02, MMNG15]. When working with ultrasound, data filtering is an important step [SHW$^+$12, SBS$^+$17]. Creating a compound volume of the ultrasound data is also very useful [VBS$^+$13, MHP$^+$14]. Analyzing the fetus data can be performed in many different ways including skeletonization [JKT13, SS04, PSS$^+$03]. A more detailed approach is to use an inverse-kinematics (IK) skeleton of the data which already includes the joint information [GS99, CSYB05].

Rigging is also discussed in this chapter by having a look at different related works [BP07, SFW$^+$14]. Another option to make volumetric data formable is by using joint mapping [BTST12]. One option is also to perform the rigging manually by using Bender [FOA$^+$14]. Skinning the data is the next step after having positioned the skeleton in the data [MTLT88, BTST12, BP07, FOA$^+$14]. Reforming of the volumetric data may be performed using flattening-based techniques [KMM$^+$18, KCE$^+$12, VLP05, MPG$^+$17, RLT$^+$15, KST$^+$14]. Another option to perform volume manipulation tasks is to use VolEdit [SS04, DS05]. Volume manipulation is also used in terms of surgery preparation [NM10].

3.1 Standardization in visualization

According to Bagci et al. [BUB10] standardization plays a major role when trying to compare medical imaging data between patients. Standardization describes techniques that can be used in order to cope with inter-subject variations. In case of magnetic resonance imaging, standardization has to be performed in order to make the image intensity comparable because it differs from patient to patient and from acquisition to acquisition [NUX02, BUB10]. The intensities do not have a fixed meaning and therefore only a standardized version of the data is usable [NUX02]. Standardization often goes along with normalization. Using a value range from 0 to 1 is commonly used when trying to standardize medical image data, especially in case of magnetic resonance imaging [BUB10].

Another example where standardization is used to provide comparability is the circle of Willis which is an arterial structure in the brain's blood supply [MMNG15]. Miao et al. [MMNG15] introduced the CoWRadar which uses an abstract visualization of the data by using specially defined concentric pie charts. The usage of an abstract version of data provides more insight and comparability, according to Miao et al. [MMNG15]. Another way to abstract and standardize representation of e.g. vessel structures is to use skeletonization [LDB95]. The skeleton of 3D datasets is the minimal representative data included. Having this information makes the data usable to be compared between cases.

Having a look at different standardization and normalization techniques is important for this thesis because the transformation to a T-pose is a standardization technique. The medical personnel should be used to work with such techniques and there should be a sense behind doing so.

3.2 Ultrasound data handling

From a technical perspective ultrasound is an acoustical investigation and therefore is susceptible of various noise induced artifacts [SHW$^+$12]. The volumetric area which is perceivable during an ultrasound investigation is limited [VBS$^+$13]. It is for example unable to scan the whole liver or a whole fetus, after a specific development period at a glance [VBS$^+$13]. Therefore techniques to generate compound volumes on the fly are important [VBS$^+$13, MHP$^+$14].

3.2.1 Filtering 3D ultrasound

Especially when using 3D ultrasound investigations in a medical context some kind of filtering is essential in order to perceive anatomical structures visually and mentally. Therefore Solteszova et al. [SHW$^+$12] introduced a technique called Lowest-Variance Streamlines for filtering 3D ultrasound. The filtering identifies borders of structures by calculating the lowest variance direction. According to Solteszova et al. [SHW$^+$12] their procedure is similar to a streamline integration in a vector field. The approach is interesting because it is especially useful for medical ultrasound applications in order to reduce artifacts like noisiness and speckles which may lead to a miss interpretation of the underlying data. The output of the filtering method can be seen in Figure 3.1.

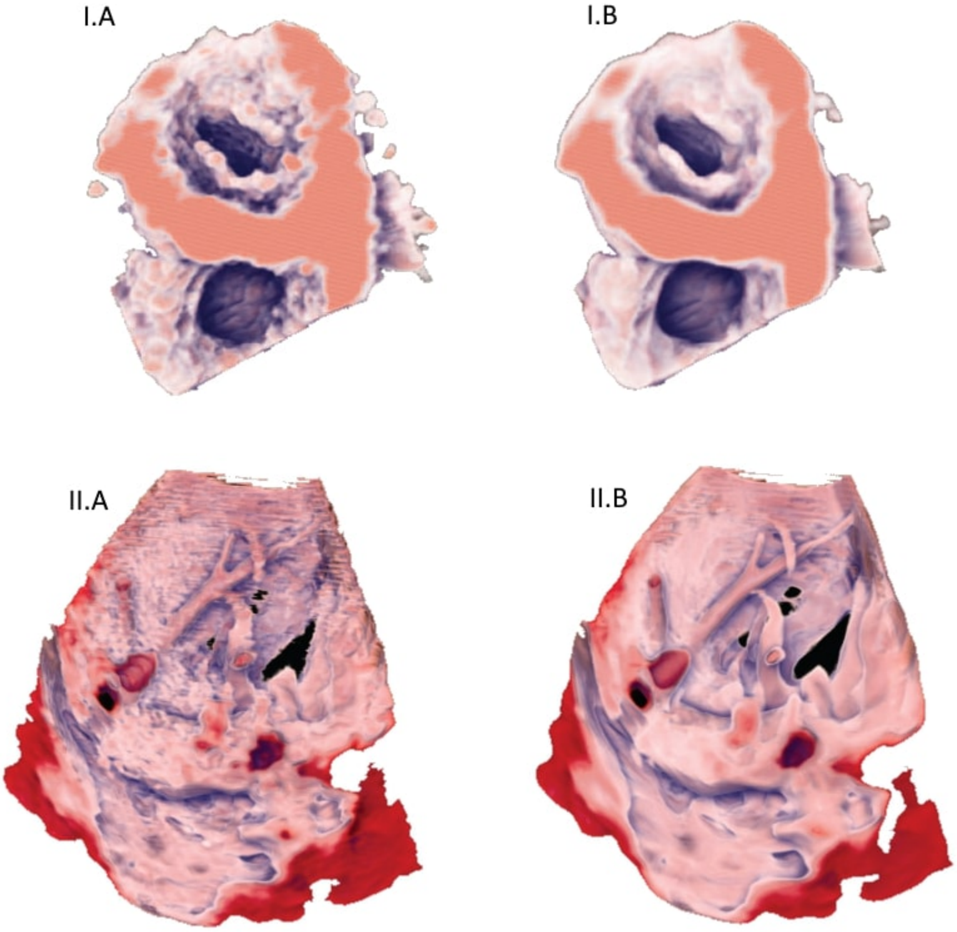


Figure 3.1: I.A and I.B depict a cardiac ultrasound before and after filtering and II.A and II.B show an ultrasound of the liver before and after filtering [SHW$^+$12].

Solteszova et al. also introduced another approach for filtering especially fetal ultrasound called the output-sensitive filtering of streaming volume data [SBS$^+$17]. The approach is especially designed to be used in 4D ultrasound investigations where it is important that the filtering is very fast and accurate. They limit the filtering to regions of the volume which have a potential to have an effect on the image e.g. occluded regions do not have to be filtered because they cannot be seen anyways. They state that their approach is an important improvement for streaming volume data as it is the case in a 4D ultrasound investigation. The output of the filtering applied to a fetal ultrasound investigation of Anna is depicted in Figure 3.2.

Filtering of 3D ultrasound data is important to get rid of artifacts that might affect

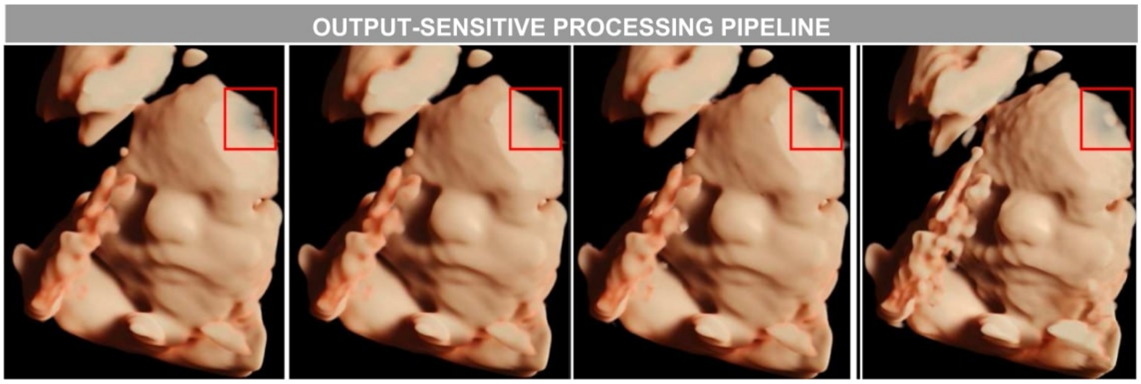


Figure 3.2: The output of the output-sensitive processing pipeline with different percentage amounts of voxels filtered. From 38% on the left side to 0% on the right side [SBS+17].

the methods described in this thesis. Filtering can be performed directly during the investigation of the data or after recording them. Without filtering the rigging and transformation can not be carried out.

3.2.2 Compound volume generation

Viola et al. [VBS+13] state that a compound view of volumetric data created by ultrasound investigation can be useful for clinical investigation. It enables the clinicians to e.g. slice through a whole organ after is has been registered at a glance. Especially in fetal ultrasound investigation after a specific time, when the fetus is too large to be perceived as a whole, registering the data as well as producing a compound-volume is very useful and essential for the introduced procedure. The possibility to swipe with the ultrasound investigation probe over the body and seeing the result on the screen in real-time producing a compound volume is essential for a better understanding of the examined specimen. The creation of a compound volume presented by Viola et al. [VBS+13] is depicted in Figure 3.3.

Being able to automatically register and process 4D ultrasound data without externally tracking the ultrasound probe can be done with a model-based approach [MHP+14]. Mueller et al. [MHP+14] state that their approach of generating a compound volume of the liver uses multi-modal registration considering the 3D voxel neighbourhood. They have used various transformations and optimization strategies in order to register the different ultrasound investigations during one session. They also had to cope with anatomical distortions of the ultrasound investigation which they stated had not a significant impact if the continuous motion of the investigation sweep had no rapid changes in direction.

Depending on the pregnancy week at some point of time the fetus will be bigger than the perceivable volume of the 3D investigation probe. When wanting to unfold the fetus the

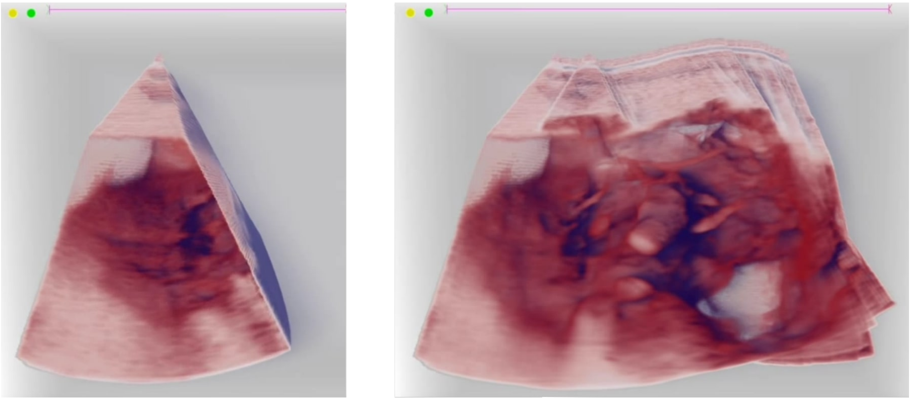


Figure 3.3: On the fly creation of a compound volume by Viola et al. [VBS+13] The right part of the image in comparison to the left part shows the progress of the compound volume generation. The images have been taken from the video of.

whole fetus information has to be available in one volume. Therefore compound volume generation is important for the method described in this thesis.

3.3 Understanding the data

Having the ultrasound data of the fetal investigation filtered and registered in order to generate the whole volume the next steps will be about figuring out the position of the fetus. There are different approaches to find more detailed content-wise information of volumetric data. One step would be to skeletonize the data and try to find out if the joints and bones of the fetus can be found automatically [SS04, JK17, JKT13]. Another approach is to rig the data by including a predefined armature or skeleton which is then either automatically [BP07] or manually [FOA+14] included in the data.

3.3.1 Skeletonization

Skeletonization of data is an approach where it is tried to reduce the given object to the most important features [SS04]. This description may then be used for further processing and can be linked back to the data it has been abstracted from [JKT13]. When writing about 3D skeletonization one has to distinguish between surface and curve skeletons. According to Jalba et al. [JKT13] the surface skeleton of a 3D shape is the collection which contains the "loci of maximally-inscribed balls in a shape" [PSS+03]. In respect to the surface skeleton the curve skeleton may be described as one dimensional curves which are positioned at the center of the 3D shape. The second skeleton is especially interesting when trying to transform fetus data. Jalba et al. [JKT13] presented a technique which is based on the graphics processing unit (GPU) for calculating curve skeleton points by firstly evaluating the surface skeleton and then detecting the curve skeleton points by their algorithm. Their algorithm works fast, is reliable and delivers a high-quality curve skeleton and automatically delivers the mapping to the underlying data. The results of their approach are depicted in Figure 3.4.

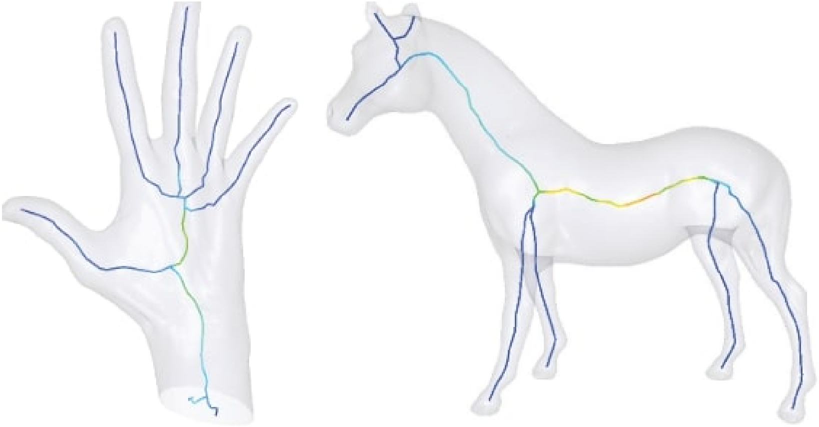


Figure 3.4: Result of the curve skeleton produced by the algorithm introduced by Jalba et al. [JKT13] for a human hand on the left and a horse on the right.

Another approach to produce a skeletal representation of voxel data is presented by Jin and Kim [JK17]. Their algorithm is based on a thinning approach and can be implemented on the GPU. Jin and Kim identified general patterns of 3D neighbourhood relation which can be used in order to decide if a voxel is part of the skeleton or not. The algorithm can be executed in parallel for each voxel in the data. They also introduced a skeleton correction algorithm which takes care of connecting skeletal parts which may have been disconnected by the thinning algorithm. The authors evaluated their approach using 100 models of the SHREC 2015 benchmark testing set and state that their proposed algorithm is robust and provides an overall good performance. The results when performing on a human model can be seen in Figure 3.5.

A third approach of gathering the skeleton of a 3D model is presented by Singh and Silver [SS04]. They used two approaches one is based on the calculation of the curve skeleton introduced by Gagvani [GS99] and the other one is based on a potential field algorithm shown by Cornea et al. [CSYB05]. Singh and Silver state that they use both approaches in order to get the IK skeleton of the data which does not only consist of the simple skeleton but also has information about bones and joints [SS04]. This information is especially important when thinking of interactively transforming data to another position. The positioning of the joints and the bones on the skeleton can be performed manually or semi-automatically. The semi-automatic approach uses the potential field algorithm and the manually one the thinning algorithm. The authors state that the semi-automatically way may be more user-friendly on the other hand the algorithm used in the manual processing is faster [SS04]. One major benefit of the algorithm introduced by the authors is that they already automatically calculate the correspondence of each voxel to the specific bone. This is also a huge benefit when thinking of transforming data to a new position. Figure 3.6 represents the data and the corresponding IK skeleton.

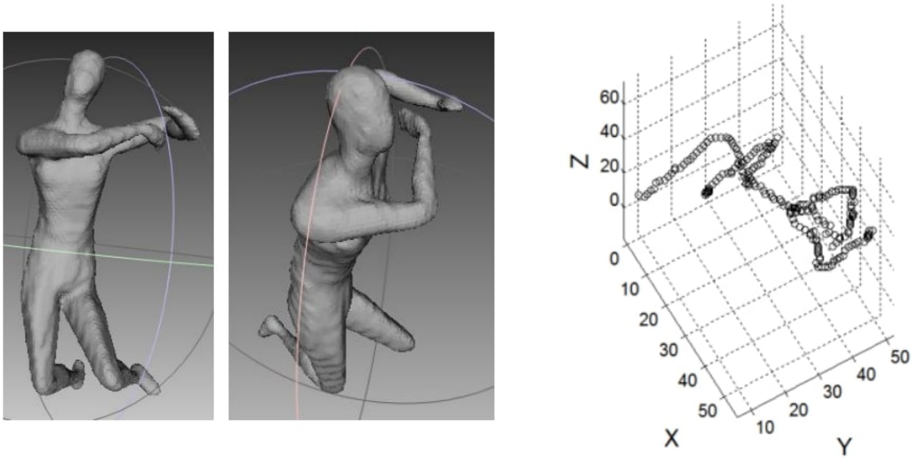


Figure 3.5: Human model and the skeletal representation using the partial parallel 3D thinning based skeletonizaion algorithm introduced by Jin and Kim [JK17].

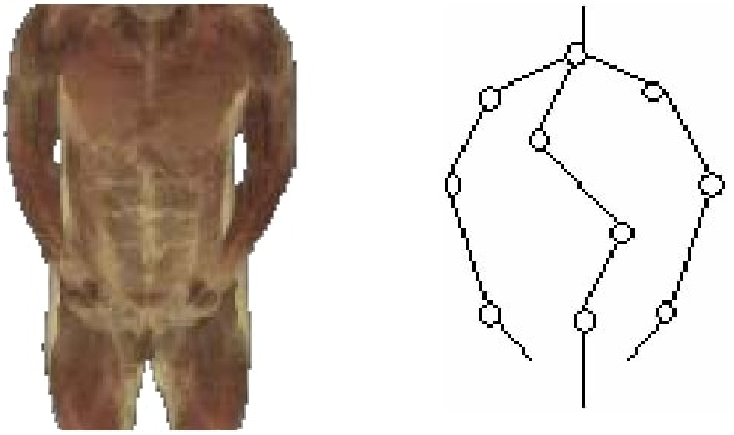


Figure 3.6: The given data and the corresponding IK skeleton are shown. The skeleton already features the joints where parts of the volume can be moved against [SS04].

Skeletonization delivers information about the anatomy of the fetus in the data. Identifying joints and separating the volumetric data according to segments in the skeleton is very important when trying to transform the model. Skeletonization can be a very good option to describe the parts of the fetus that have to be moved in order to transform it into the T-pose.

3.3.2 Rigging approaches

Rigging defines the process to determine the skeletal structure and the position of it in a 3D model in order to be able to move parts of the model and knowing how the surface or the volume has to be deformed [BP07]. This task can either be performed in a manual way needing an animator or similar professional to define the skeleton and the position or in a semi-automatic or automatic way. The complexity to automatically rig a 3D model should not be neglected.

Automatic rigging and animation of 3D characters

The first approach for automatically rigging a 3D model represented by its mesh is introduced by Baran and Popovic [BP07]. The procedure creates a distance field from the given mesh and it identifies a skeleton by approximating the medial surface. The extracted skeleton is then fitted to template skeleton to refine the output. For their approach they are using discrete penalty functions in order to match the extracted skeleton to a predefined instruction how a skeleton should look like. For example the penalty function checks where the feet are and that they should be on the ground and not at the top area of the mesh. With those penalty functions it can clearly be seen that the method loses generality. The general approach of rigging in the Pinocchio prototype is shown in Figure 3.7.

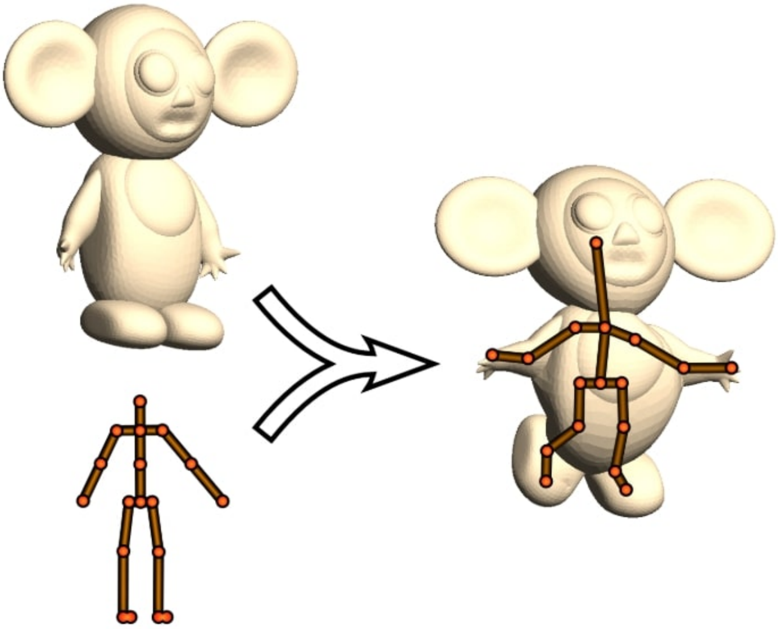


Figure 3.7: General overview of the pinocchio rigging prototype placing an predefined skeleton into a 3D mesh model by extracting the curve skeleton and than matching it with a predefined prototype skeleton [BP07].

Shapiro et al. [SFW$^+$14] introduce another approach which is quite similar to the strategy of Baran and Popovic [BP07] but uses voxels instead of the mesh representation of the 3D model. They have tried to make 3D scans of humans available for using them in computer animations. Therefore they scanned the humans in four different poses and created a 3D model out of the gathered data. The difficulty of the rigging was that their meshes were not perfect and had holes and other artifacts. Therefore they decided to create a voxelized version of the mesh by using depth buffer carving. The data they use are produced by the consumer product Kinect [SFW$^+$14]. After voxelizing the data they used the same technique as Baran and Popovic [BP07] to find the skeleton in the mesh. The approach therefore has the same downside that it does not allow the model to be in another formation before the rigging is done. It has to be in a predefined initial position. The output of the procedure is depicted in Figure 3.8.

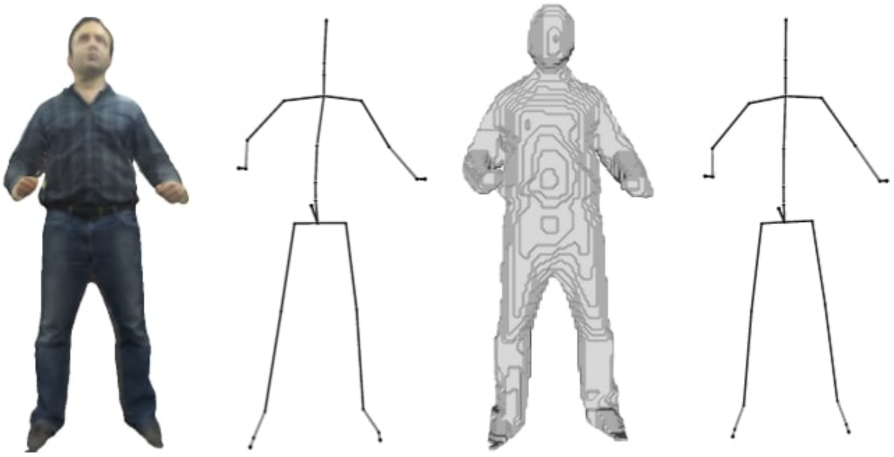


Figure 3.8: On the left the original mesh of the human 3D scan and the corresponding skeleton. On the right the voxel representation of the mesh using the depth buffers and the skeleton of it. It can be seen that the skeletal representation looks the same for both representations of the 3D model [SFW$^+$14].

Joint mapping

Bharaj et al. [BTST12] introduce a procedure which is able to automatically rig a character by using a joint mapping technique. Their approach is especially suitable for multi component characters. They are using point clouds in order to sample each identified component of the input model. These clouds are represented as a graph which is simplified and transformed into a tree by using clustering. Their attempt needs as a second input an animation skeleton which is predefined. The authors use joint mapping in order to find the relation between their tree and the given animation skeleton. Normally the mapping is defined by a user. Their approach is closely related to the approach stated in this thesis because they also would like to afterwards find a way to transform the data into a standardized T-pose [BTST12]. The transformation of the data can be done using an animation software because their export format is an industrial standard. Two examples of their rigging process can be seen in Figure 3.9.

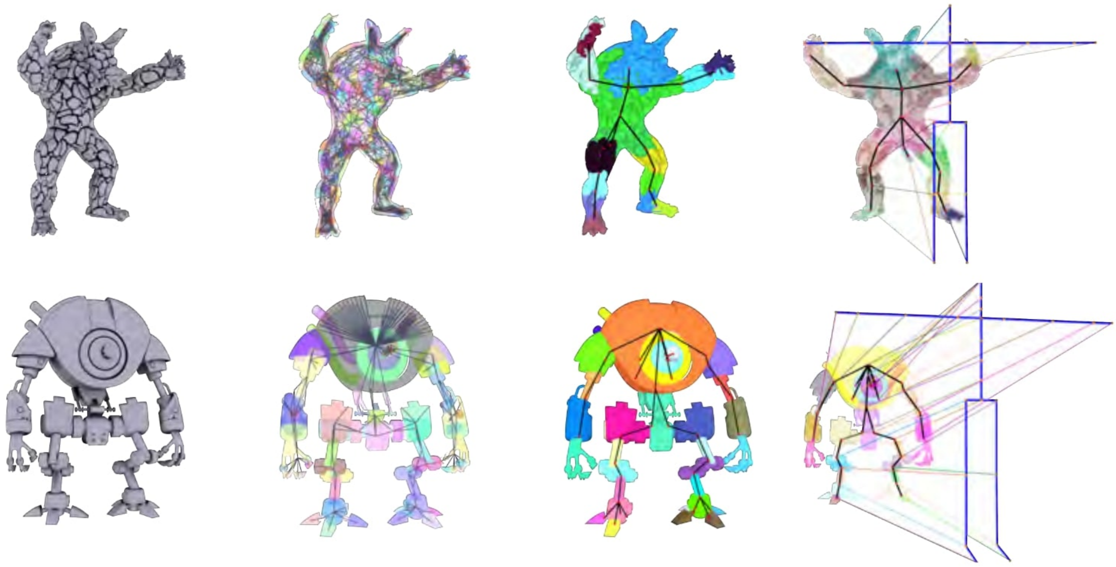
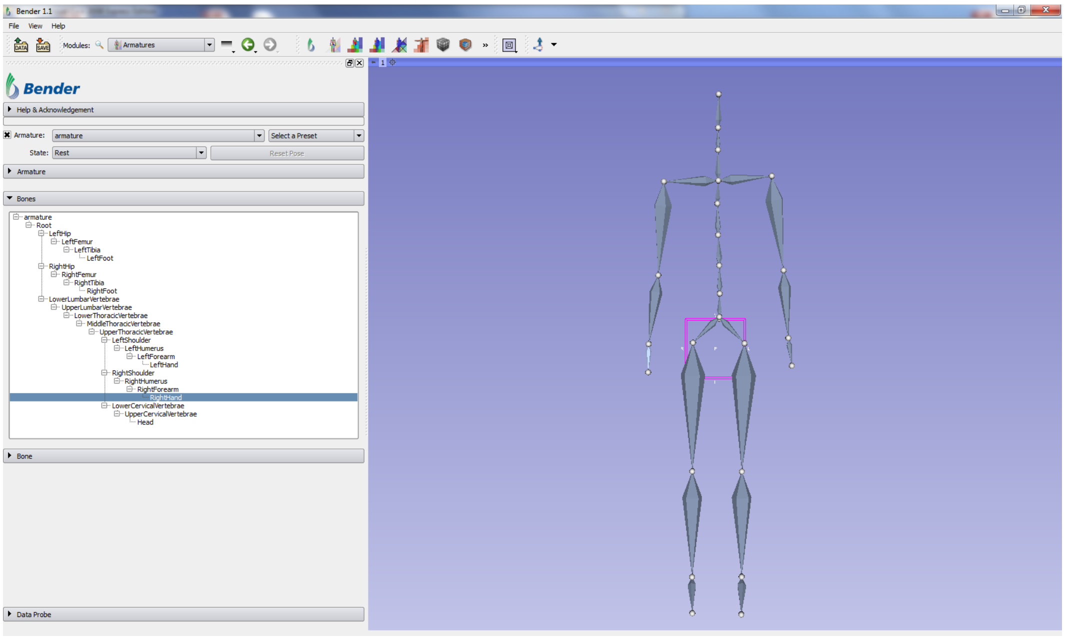


Figure 3.9: Two multi component models rigged by the approach of Bharaj et al. [BTST12].

Automatic rigging is a very promising approach and can be done in multiple ways. Some of those approaches are working with mesh data which is not available in case of 3D ultrasound. Other work based on voxels, but nearly all assume that the data is in a predefined shape or pose. Therefore those approaches are not directly applicable to the data given in this thesis.

Manual rigging

Manual rigging is the process to define the position of a predefined skeleton or armature in a mesh or volume. The user has to first define the skeleton and than to drag the joints and the bones into the model in order to define the rigging of it. Finet et al. [FOA$^+$14] introduced a software called Bender which is open source software used for efficient model posing and morphing. One of their modules is called Armatures and enables the user to rig a model by defining or loading an armature in the *.vtk format and afterwards also saving the result. The armature can be represented in different ways using lines, cylinders or octohedrons. The armature parts are defined from a head to a tail. The head is normally connected to another armature part unless it is the first one. The tail may not be connected e.g. in case of the feet or the hands. The armature created has a hierarchy which can be defined individually. Each part of the armature can have a name. An example armature and the software are shown in Figure 3.10. Manual rigging is an approach which needs the user to interact with the program for some time and it can also be exhausting. Therefore the number of segments and joints should not be too high and the interaction mode shall be appealing.



☒ Armature: armature

State: Rest

```
armature
  Root
    LeftHip
      LeftFemur
        LeftTibia
          LeftFoot
    RightHip
      RightFemur
        RightTibia
          RightFoot
    LowerLumbarVertebrae
      UpperLumbarVertebrae
        LowerThoracicVertebrae
          MiddleThoracicVertebrae
            UpperThoracicVertebrae
              LeftShoulder
                LeftHumerus
                  LeftForearm
                    LeftHand
              RightShoulder
                RightHumerus
                  RightForearm
                    RightHand
              LowerCervicalVertebrae
                UpperCervicalVertebrae
                  Head
```

Figure 3.10: Screenshot of the armature module of Bender. The shown armature is only en example and can be individually customized and generated. The screen is also used to drag the armature into the model if a model is loaded [And14].

3.4 Skinning or weighting of the armature

After the armature or the skeleton is defined and the position inside of the mesh or volume has been found the data has to be weighted. That means that either the mesh or the voxels have to be linked to parts of the armature in order to know which parts have to be moved or transformed to create an animation or simply a movement. Some rigging approaches which originate from skinning the data already have this information available because during the skinning process the program simply has to save which voxels have to be removed in order to create parts of the skeleton. Afterwards these voxels are those who belong to the specific parts of the armature. One approach that is well known and has been introduced by Magnenat-Thalmann et al. [MTLT88] is called linear blend skinning (LBS). The principle behind the algorithm is that between two adjacent points of the armature there can be a so called heat distribution plot. This distribution can be used in order to define which mesh or voxel parts are deformed or affected by a movement of the given segment. Figure 3.11 visualizes how the LBS works.

In case of the topological approach introduced by [BTST12] the skinning is already available because it is well known how the data have been abstracted in order to generate

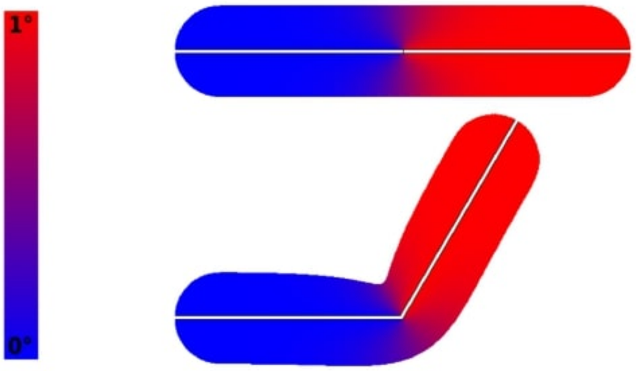


Figure 3.11: The left side of the image represents the heat distribution between two points which can also be seen at the top. The lower parts shows how the e.g. voxels will be affected if the right part of the joint moves about 45 degrees [BP07].

the point cloud, the graph and at the end the tree. The skinning of their approach also conforms the LBS approach. The Pinocchio prototype uses the LBS method to calculate for each vertex of the mesh the weight how it is affected by a bone transform. It is stated that it is highly affected by the distance to the bone and the interaction between two bones is calculated by using the LBS method [BP07].

3.4.1 Geodesic voxel binding

Dionne and Lasa introduced the geodesic voxel binding as a voxel weighting approach in order to be able to transform voxel based characters using an armature [DdL13]. The authors state that their approach is particularly useful when having data that is not represented by a watertight mesh. They use the geodesic distance between each voxel and the nearest bone from the skeleton in order to determine which voxel is affected by which bone. The benefit is that the mesh does not have to perfect and the complexity of the calculation is low and therefore fast [DdL13]. The process described by the authors is shown in Figure 3.12.

Bender VolumeSkinning

Bender has a volume skinning module which takes the volume as an input and a predefined armature [FOA$^+$14]. It then calculates the weights an delivers as output a new volume where the voxels are identified by the IDs of the armature parts they belong to. The mapping between armature parts and the voxels belonging to it are represented using different colors. One example output of the bender software is shown in Figure 3.13.

The skinning or weighting of the data can be performed by applying different methods. Some take more context information into account, others assume evenly distributed data

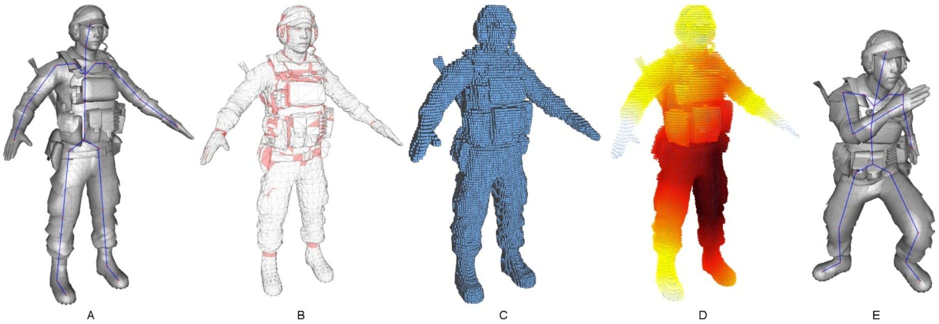
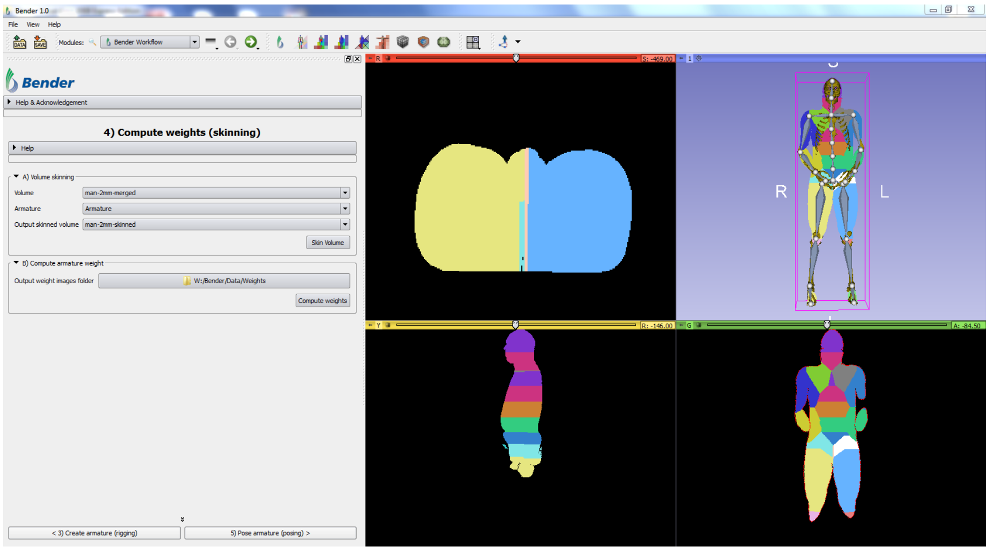


Figure 3.12: This image represents the whole processing pipeline from the mesh at the beginning over the voxelized model in the middle, the geodesic representation at position D and the transformed mesh at the right hand side. Position B represents that there are some imperfections in the mesh which make it not watertight and therefore unsuitable for different skinning approaches [DdL13].


4) Compute weights (skinning)

A) Volume skinning

Volume: man-2mm-merged

Armature: Armature

Output skinned volume: man-2mm-skinned

B) Compute armature weight

Output weight images folder: W:/Bender/Data/Weights

Figure 3.13: The Bender VolumeSkinning module with an example output of the armature and the skinned volume depicted in different colors representing which parts of the model belong to which armature part [And14].

between two segments of the skeleton. In case of this thesis the weighting may has to be performed in respect to the segments in order to follow some anatomical rules.

3.5 Reformation of volumetric data

The reformation of volumetric data can be performed in different ways. Some possibilities are shown in this sections.

3.5.1 Flattening-based medical visualization techniques

In the survey of Kreiser et al. [KMM+18] about flattening-based medical visualization techniques different approaches for bone based reformations are given. Trying to pose medical data into a new formation especially when thinking of a fetus the biomechanics of the body is very important [KCE+12]. In their paper they are analyzing different techniques to visualize the kinematics of the human spine.

The human spine is often flattened in order to analyze it in a standardized way [VLP05, MPG+17]. The approach of Vrtovec et al. [VLP05] is based on magnetic resonance images of the spine and tries to work with minimal user interaction. The transformation is based on mapping the image coordinate system to the spine-based coordinate system. They tell that their proposed method reduces complexity and it also improves the presentation of the features. The input and output data as well as the mapping between the coordinate systems is depicted in Figure 3.14.

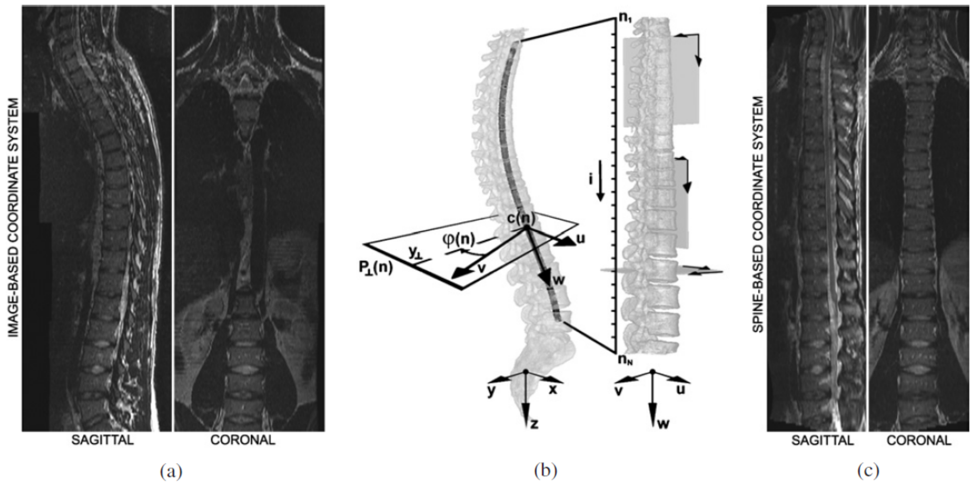


Figure 3.14: The left part of the image shows the image-based coordinate system. The middle part represents the mapping of the two coordinate systems and the right part is the result of the spine data in the spine-based coordinate system [VLP05].

Unfolding the ribs has been discussed by Ringl et al. [RLT+15]. The unfolding process results in a 2D representation of the whole rib cage which may be used to identify fractures. The method is dependent in some cases on user interaction in order to correct the segmentation of the ribs. The authors state that looking at each CT image is tedious and needs lots of time [RLT+15]. The results showed that the analysis of the unfolded rib cage needed less time and the sensitivity has even been increased. Figure 3.15 shows the ribs unfolded.

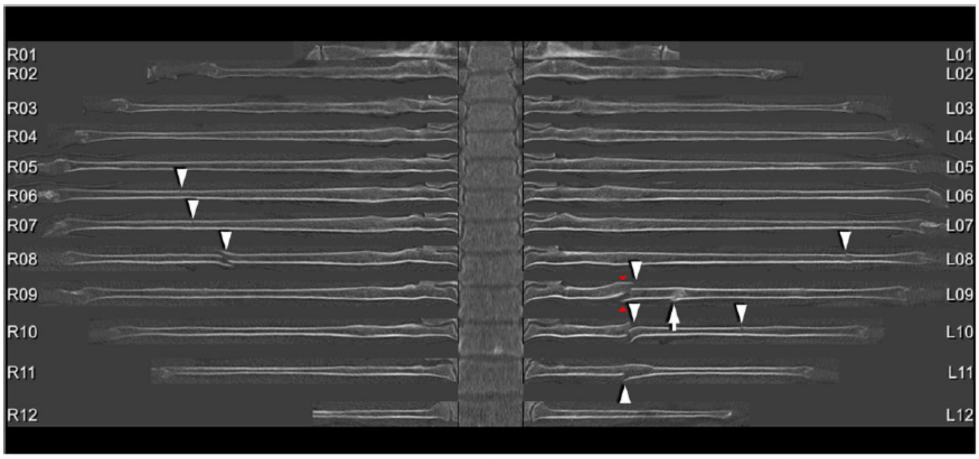


Figure 3.15: Unfolded and flattened image of the ribs with fractures marked. The image also depicts artifacts which are caused by the unfolding of curved data [RLT+15].

According to Kretschmer et al. [KST+14] anatomy driven reformations may be used as a tool for medical image reformation. Their tool shows promising results for trauma related bone reading tasks [KST+14]. They are using the anatomical information of different structures like the feet or the rib cage in order to transform the medical imaging data to a normalized form. The result is better perceivable and can be easily used for further analysis by specialists [KST+14]. Having a look at unfolding of medical data and the analysis of it is especially interesting for this thesis because it proves that the idea is worth investigating and promising.

3.5.2 VolEdit

Singh and Silver have introduced a method to manipulate volumetric data by using the IK skeleton [SS04]. After having identified the skeleton in the data and the joints of the skeleton they can use their software to manipulate the volume. One approach is to transform the skeleton using rotations around a joint. They are able to move each segment of the skeleton individually in respect to the according joint. In their approach the authors are using a program called VolEdit to modify the volumetric data [SS04]. A simple interaction can be to apply a transfer function to different parts of a model and to combine them. Figure 3.16 represents a lobster model which has been transformed using two different transfer functions in order to move the arms of the animal away from the body.

Using the same software VolEdit the authors also showed another example where they tried to unbend a rat which has been pictured in a small housing. This technique can be used to bring pictures into a standardized form like used in this thesis [DS05]. Figure 3.17 shows the rat example before and after the unbending. Another interesting approach of Singh and Silver is to do a so called juxtaposition of different data sets. Using their

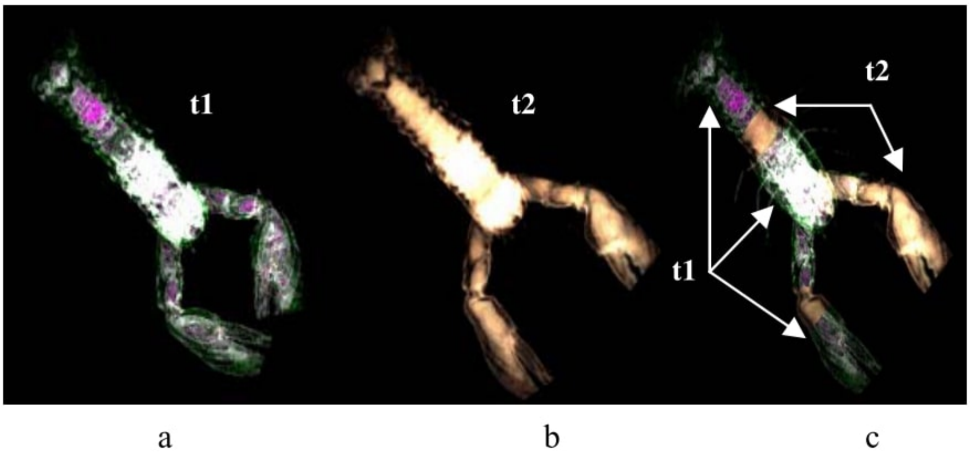


Figure 3.16: The lobster model transformed with two different transfer function which are applied to different regions of the model in order to move the arms away from the body [SS04].

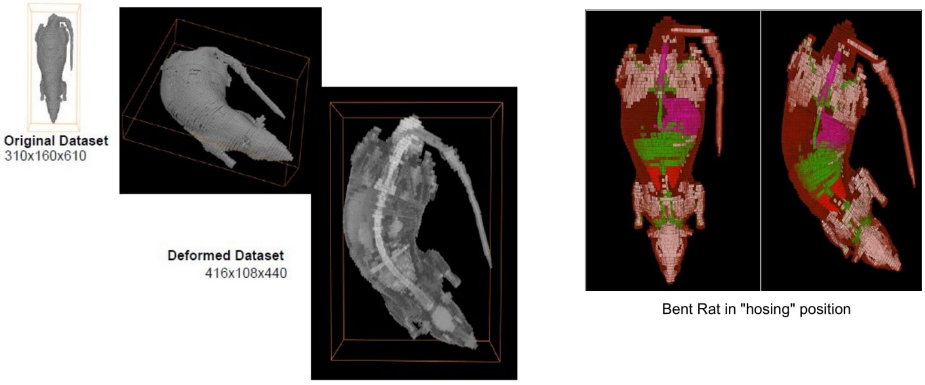


Figure 3.17: Unbending of a rat which is capt in a small housing [DS05].

program they are able to exchange the arm of a man by the arm of a lobster and the result looks quite promising. The application of the juxtaposition is depicted in Figure 3.18.

VolEdit is a good approach when the data can be described by an IK skeleton. If it is able to find it in the data the tool may be used to transform also fetus data.

3.5.3 Volume manipulation for surgical processes

Nakao and Minato represent in their paper a novel technique which can be used by surgeons to prepare for their surgical process or to share their knowledge in an interactive and visually appealing way [NM10]. In their application the user is able to manipulate patient specific volumetric data according to a real surgical intervention. The authors used computed tomography (CT) and MRI images as a source for their method and

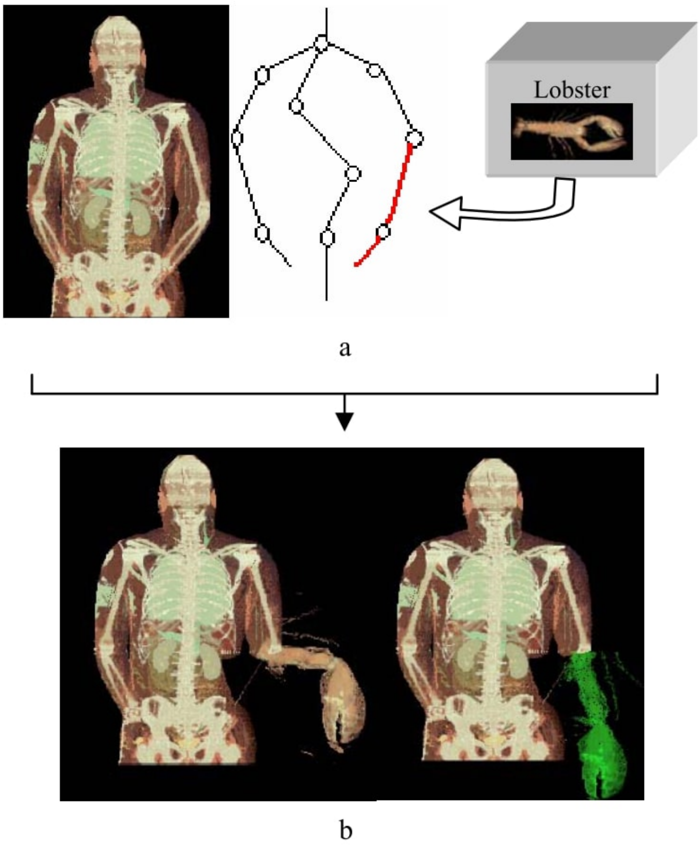


Figure 3.18: Juxtaposition of the arm of a man and the arm of a lobster [SS04].

enable the user to perform physically correct modelled surgical procedures. The used a vertices based grid and finite element analysis in order to create a visualization that behaves physically correct [NM10]. An example for such a surgical intervention and the volumetric rendering can be seen in Figure 3.19.

This approach looks very promising and describes different techniques to interact with volumetric data and having in mind how the tissue would react in reality. Physical limitations and simulation of real tissue behaviour would also be interesting when transforming the fetal data.

3.6 Analysis and outcomes

Overall we can say that some parts of our workflow have already been done and some have to be designed by us. We think that using MeVisLab as a tool to perform the preprocessing steps is appropriate. Bender [FOA^{+}14] provides us already with a workflow that guides the user through the rigging and the weighting and is compatible with various

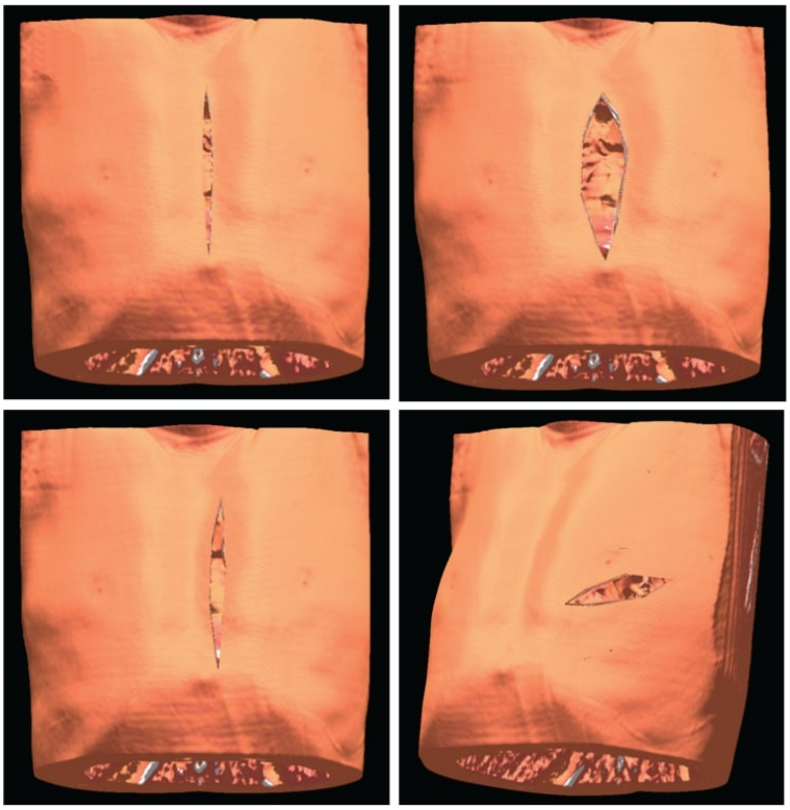


Figure 3.19: Visualization of a volume cutting considering the tension that arises from the surrounding tissue resulting in a cleft with a differing size [NM10].

input and output data formats. The transformation is the part of our work where the existing approaches are not applicable or would have to be reworked. Therefore we decided to implement this step from scratch. We used 3D Slicer [PHK04] as a platform because it is completely compatible with Bender [FOA^{+}14] which is also 3D Slicer [PHK04] based. As a future step we also think about exporting the relevant Bender [FOA^{+}14] features and import them in our application. On the other hand our approach will be available as an own extension for 3D Slicer [PHK04] and could also be integrated in Bender [FOA^{+}14]. In terms of analyzing the result we think that 3D Slicer [PHK04] offers various options to perform this step.

CHAPTER 4

The Vitruvian Baby

In this section the methods used to accomplish the aims of the application are presented. The sections are ordered after the workflow the user has to perform. We discuss methods which are used in the resulting application and alternative approaches which may have not found their way into the final solution will be presented. It will also be described why and how we have decided which approaches fit best for the solution.

4.1 Application overview

In the first section the aim is to provide the reader an overview of the presented methods. Figure 4.1 shows the workflow of the application. The first step is to load of the data. This step can be very complex because it highly depends on the format the data is provided. In case of ultrasound data originating from an clinical investigation or already registered data representing the whole fetus, the format might be given in DICOM. Another possibility would be to provide the data using a *.raw format which only includes the data itself and no surrounding information which e.g. DICOM includes. The *.raw format might come along with a *.mhd file which is also called the MetaHeader and represents meta information about the data included in the *.raw file. The data might also be provided in various other formats and therefore there might be some restrictions. Thinking about loading data the usage of the tools MeVisLab [MeV] or 3D Slicer [PHK04, FBKC$^+$12] is endorsed. Both software solutions are publicly available, open source and for free.

The preprocessing of the input data is performed using MeVisLab. The preprocessing pipeline used in this thesis includes a threshold processing to be able to distinguish between background data and the data belonging to the fetus. The next step is to calculate the largest connected component in the data and to get rid of all other information in the dataset. The output of this step is a volume only holding information of the fetus. Preprocessing can also be performed using 3D Slicer [PHK04]. The computational

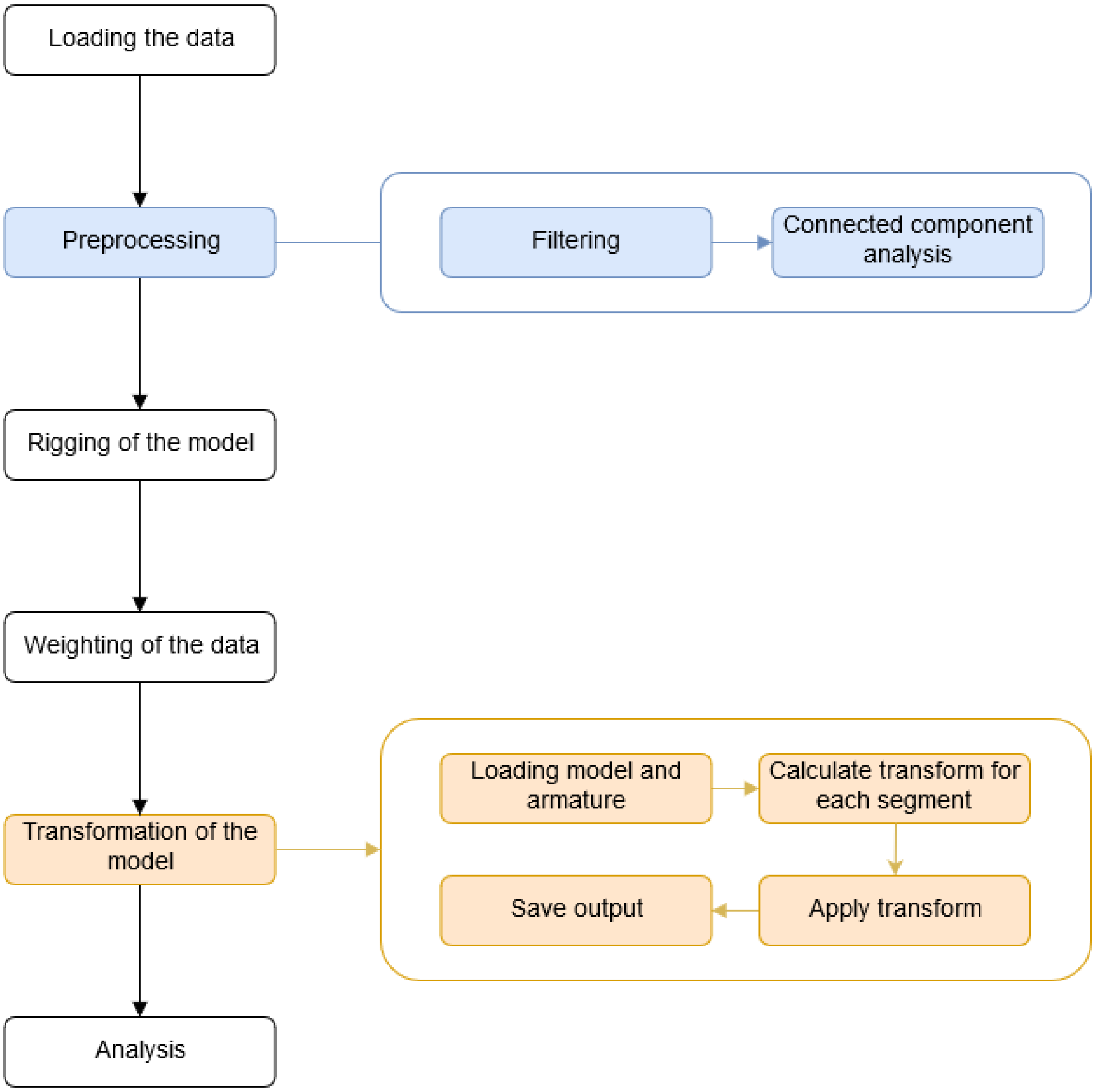

Loading the data
Preprocessing
Filtering
Connected component analysis
Rigging of the model
Weighting of the data
Transformation of the model
Loading model and armature
Calculate transform for each segment
Apply transform
Save output
Analysis

Figure 4.1: The application overview represents the pipeline that has to be performed in order to result in a T-pose fetal representation.

performance needed for this depends highly on the type of filtering applied as well as the size of the volume.

Rigging is performed using Bender which is based on 3D Slicer and the paper about it has been written by Finet et al. [FOA$^+$14]. Bender is also an open source program which includes various modules for model posing and morphing. One model is called armature and can be used to create and place an armature in a model. This step can be performed in lots of different ways like described in section 4.4. Trying to solve the rigging of a complex model in an automatic way is technologically challenging and would probably involve some machine learning aspects. We chose after considering different technologies to perform this step in a manual way involving user interaction.

The weighting or the skinning of the data is also performed using Bender [FOA$^+$14]. This processing step is used to calculate the contribution of different segments of the armature to the input data namely the volumetric data of the fetus. The output of this processing step is a volume which includes the mapping for each voxel to one of the armature segments. In Bender the module $VolumeSkinning$ shall be used which needs as input the armature and the volume. The result will be just the described one. Having a look at different approaches of skinning volumes it can clearly be seen that the computational power needed is linked to the size of the volume. The bigger the dataset the more power needed to calculate the link between armature and each voxel in the dataset.

The vitruvian baby extension introduced in this thesis is used to actually transform the data given into the T-pose. After considering different approaches of related works we chose to implement this part of the method on our own. The technological challenge is also dependent on the size of the volume which is given as input. The mathematical basis is not the most complex one, therefore the processing of the volumetric data is more computationally challenging. The transformation is performed automatically. As an import the armature and the volumetric data is required. The volume has to be weighted and each segment of the armature has to be mapped to some voxels. The transformation calculates the needed transformations of each segment in respect to the adjacent ones, resulting in a fetus model in a T-pose. As represented in the overview Figure 4.1 with a dashed line the transformation of the model might result in a feedback loop back to the rigging. The user might want to reset some joint positions in order to get a more realistic and representative result in the end. The transformation will be described in greater detail in the Section 4.6. The output of the transformation may uses different file formats but it is recommended to choose a well known form in order enable the analysis to be performed in different consumer products.

The last step of the pipeline is the analysis and has the aim to gather insight in the now transformed data. A part of the analysis step could be performed automatically e.g. some measurements like the finger to finger span or the head to toe length. The user gets new insights in this module and is be able to interactively view the now unfolded data. Taking measurements like specific circumferences or lengths like the femur length for example should be possible. This module can be very user specific because it depends on the preferences of the user and which modules they would like to use. One example of an GPU base approach is the GVDB library by NVIDIA which is particularly written for voxel visualization [Hoe16].

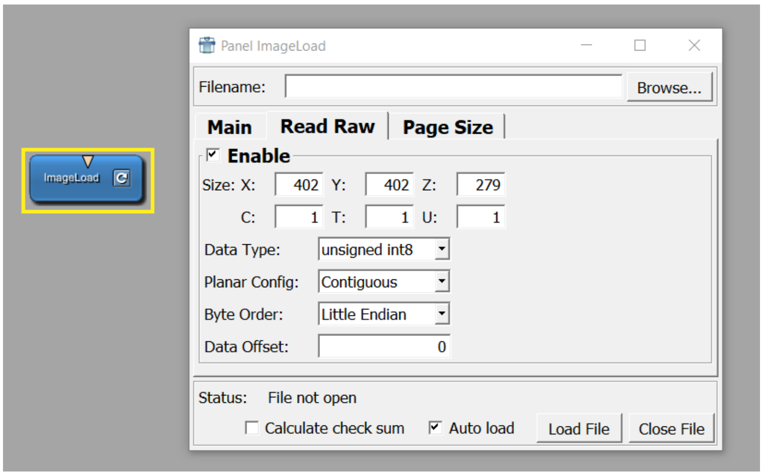
4.2 Loading the data

Loading the medical or phantom data is a step which occurs not only once during the processing of the data. Some processing steps of the pipeline may not be implemented using the same system and therefore data exchange namely loading and generating output is essential. Loading of the data somehow differs between the modalities used. In this section the loading steps which are used in this thesis are described.

4.2.1 MeVisLab

The first loading step is performed in MeVisLab [MeV]. MeVisLab is a tool which consists of a large amount of modules, the user can choose from. Certainly there are many different modules to load data. In this thesis, two can be used namely the *ImageLoad* or the *LoadAny* module. The *ImageLoad* module and the corresponding properties can be seen in Figure 4.2. The *ImageLoad* module is used to examine *.raw data but the user has to define the spatial resolution and the data type which is included in the *.raw file.

If the data is present in a *.mhd file format one might just use the *LoadAny* module. This module has the benefit that the user does not have to apply any settings but the file type has to be supported to being loaded. A *.raw file might not be loadable because the module does not have any contextual information included.



Filename:

☑ Enable

Size: X: 402 Y: 402 Z: 279

C: 1 T: 1 U: 1

Data Type: unsigned int8

Planar Config: Contiguous

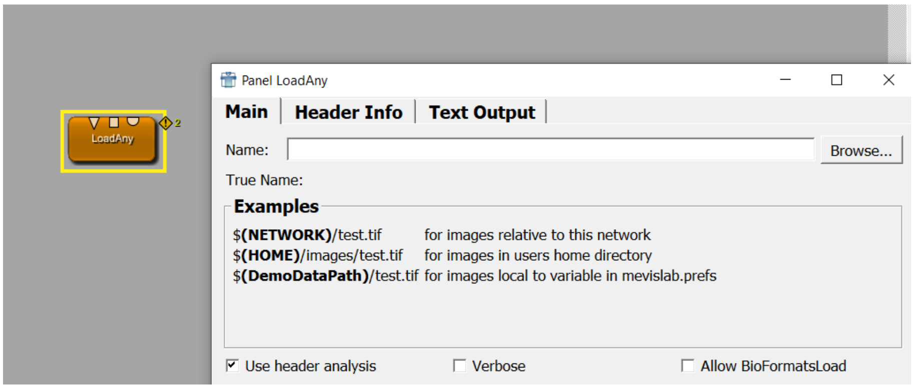
Byte Order: Little Endian

Data Offset: 0

Status: File not open

□ Calculate check sum ☑ Auto load

Figure 4.2: The screenshot shows the *ImageLoad* module and the corresponding settings where the user has to define the resolution of the *.raw file and the data type.



Name:

True Name:

Examples

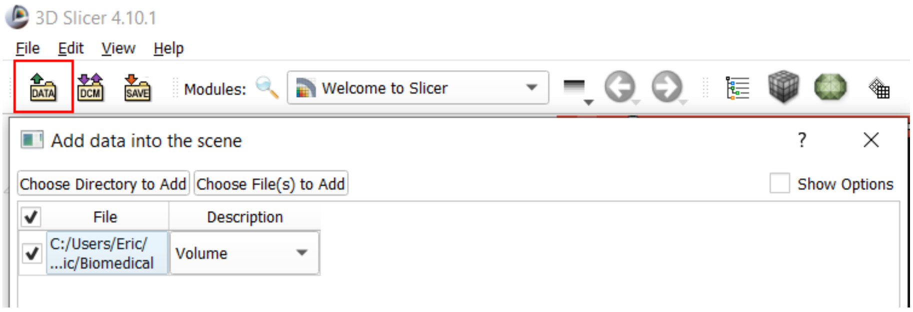
$(NETWORK)/test.tif for images relative to this network
$(HOME)/images/test.tif for images in users home directory
$(DemoDataPath)/test.tif for images local to variable in mevislab.prefs

☑ Use header analysis □ Verbose □ Allow BioFormatsLoad

Figure 4.3: The *LoadAny* module of MeVisLab which can be used to load e.g. *.mhd files along with the corresponding *.raw file without having to set any properties.

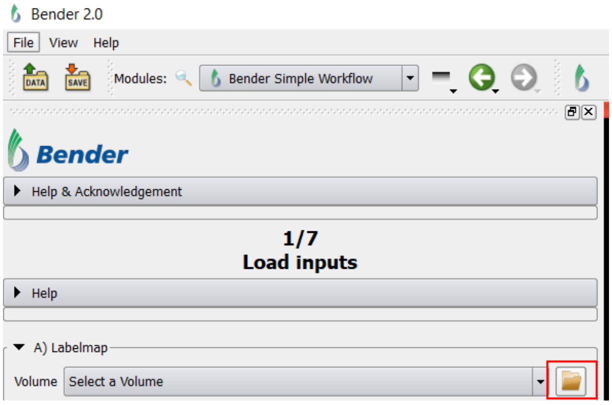
4.2.2 3D Slicer and Bender

3D Slicer [PHK04, FBKC+12] as well as Bender [FOA+14] which is a 3D Slicer based application also provide an option to load the medical data and this option is used during this processing steps. Loading data is quite simple using those programs. Loading data in 3D Slicer is performed by clicking on the button highlighted in the Figure 4.4 and in Bender by clicking the button in Figure 4.5. Both loading steps are simply performed by selecting the file to load, unfortunately both applications do not support loading files in *.raw format directly, it has to be provided in combination with the *.mhd file. The data is automatically available for visualization and further processing in the software.



☑	File	Description
☑	C:/Users/Eric/ ...ic/Biomedical	Volume

Figure 4.4: Loading of data in to the 3D Slicer program.



Volume: Select a Volume

Figure 4.5: Loading data for being processed in Bender.

4.3 Preprocessing

The preprocessing step depends as stated before highly on the data which has to be processed. In case of a fetal ultrasound investigation the data might be affected by artifacts such as speckles, shadows or information which does not belong to the fetus like the mother fluid. Cortes et al. [CKM+16] introduced in their paper a dataset of a phantom ultrasound investigation with artifacts included which may also arise in that way during a 3D fetal ultrasound investigation. In order to depict the preprocessing steps introduced in this section the publicly available dataset of the authors will be used. A realistic phantom dataset presented by Cortes et al. [CKM+16], before the preprocessing can be seen in Figure 4.6.

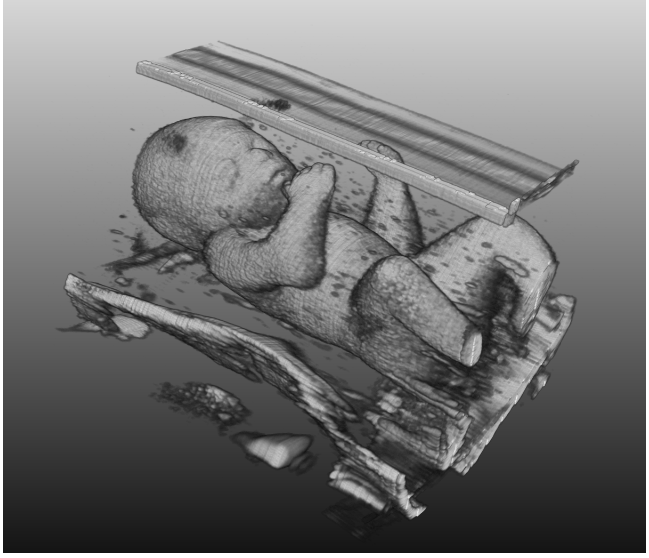


Figure 4.6: The fetus phantom model data including artifacts which may also arise during a 3D fetal ultrasound investigation [CKM+16].

4.3.1 Filtering

The first step is to apply a threshold. This operation is used to get rid of some of the artifacts in the data and to have a clear view of the fetus. The module used in MeVisLab is called *Threshold* and in the Figure 4.7 and the settings are that the data shall be greater than 60 is used. A greater value would lead to more data being included a smaller one would maybe cause to exclude data belonging to the fetus.

4.3.2 Connected component analysis

The last step of the preprocessing pipeline is the so called largest connected component analysis. This step is used in order to get rid of all elements in the volumetric dataset, which are not connected to the fetus. It is defined that the fetus in the gathered ultrasound investigation data consists of one large component without any detached parts. The second thought is that in a fetus centered ultrasound investigation the fetus should of course be the largest component in the dataset. In order to calculate the largest connected component and filter the others three components are used namely *ComputeConnectedComponent*, *FilterConnectedComponents* and *ConnectedComponentsToImage*. The first module takes the thresholded image data as an input and calculates the connected components. The second one is used to filter the components and search for the largest one by setting the *selectionMode* to *Largest*. The last stated module finally converts the component back to image data which then can be visualized again. The result of this processing step can be seen in the following Figure 4.8.

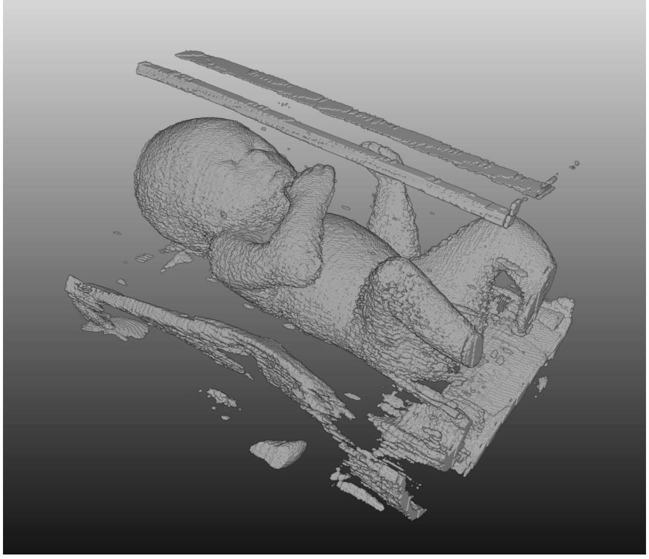


Figure 4.7: The fetus phantom model after applying a threshold greater than 60.

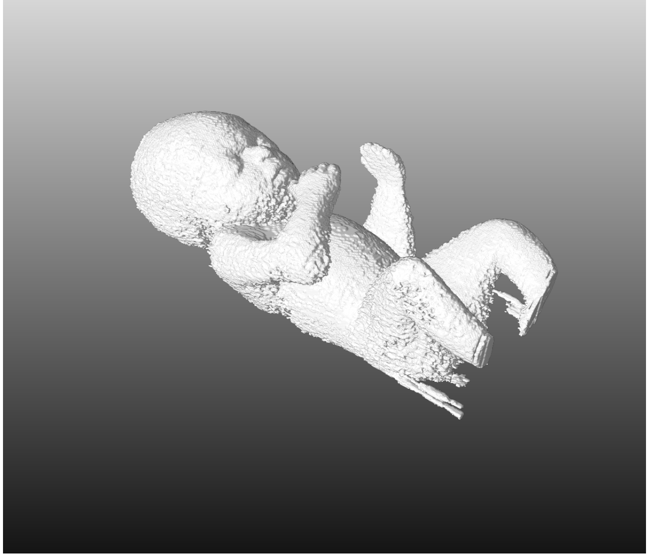


Figure 4.8: The filtered output of the connected component analysis by selecting the largest component.

4.4 Rigging of the model

Rigging of 3D models can be done in many different ways, during this thesis the approach implemented by Bender [FOA$^+$14] has been used. The first step was to produce a template skeleton which is rather simple and does not provide too many segments that have to be included in the structure. When thinking of simplification a good tradeoff between losing details and having too less details has to be found. The Skeleton used in terms of this work is composed out of 20 segments and of 18 joints. Figure 4.9 shows a schematical representation of the produced armature. The representation of this armature in Bender can be seen in Figure 4.10.

4.4.1 Bender

The rigging in Bender is done by simply dragging the skeleton piece by piece into the model. The 3D navigation has to be used in order to analyse if each part is correctly placed along each axis. The difficulty behind this approach is that in some cases the parts seem to be perfectly placed but when having a look in another representation the segments are not correct. The user has to check each segment along each axis and drag the segments until the place is correct along each axis. This procedure might seem tedious and complex but unfortunately there have not been any known approaches to rig characters automatically that are already in a transformed position which differs completely from the known T-pose. Figure 4.11 represents the input of the rigging process and the result.

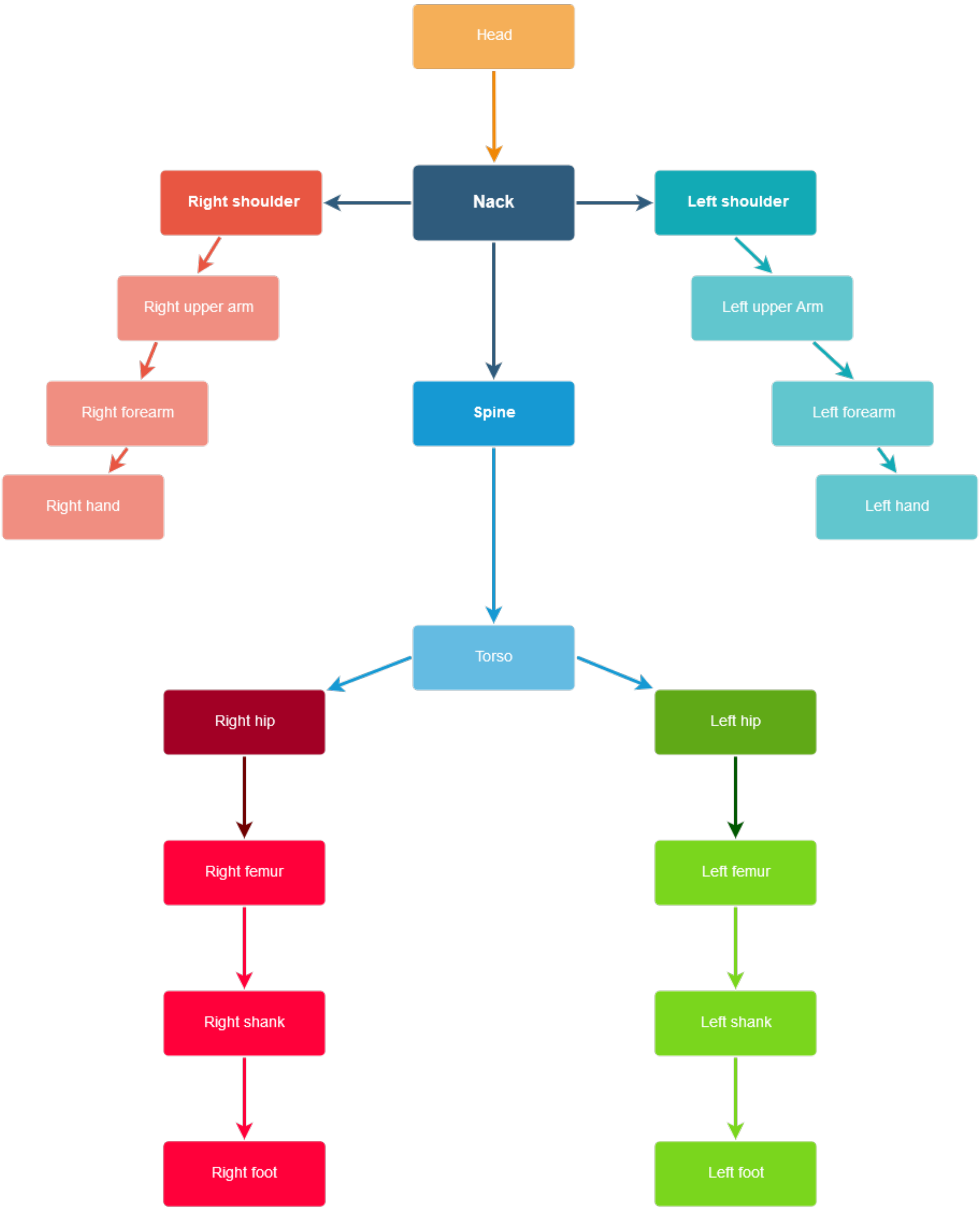


Figure 4.9: Schematical representation of the armature used in bender consisting of 20 segments and 18 joints.

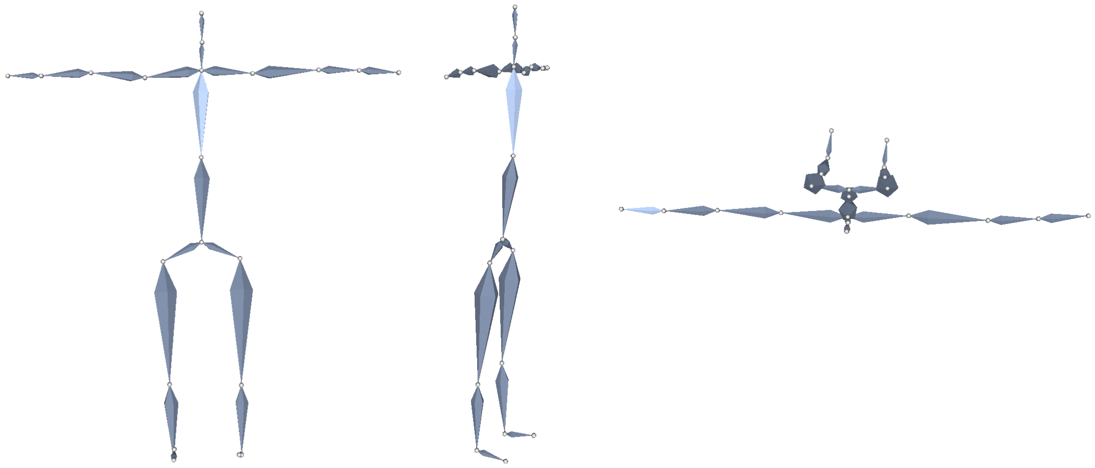


Figure 4.10: Representation of the template armature in a T-position in Bender using a octohedronal representation

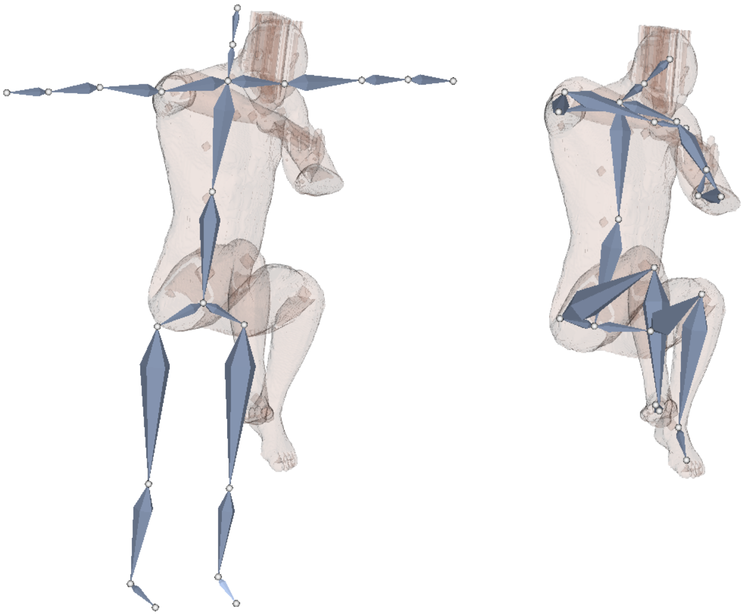


Figure 4.11: On the left side of the picture a phantom model of a man and the armature representation in Bender can be seen. The right image shows the result namely the armature perfectly in place.

4.4.2 GPU based voxel visualization

Another approach of rigging the data would be to implement a way to identify and work with keypoints of the model and automatically incorporate the armature. Therefore one approach might be to use a GPU based approach to visualize and analyze the data. NVIDIA introduced in 2016 a GPU based voxel database called GVDB which can be used as an efficient data structure for voxel visualization [Hoe16]. Their approach is open source and looked promising as a starting point for the tasks in this master thesis. First the visualization aspect has been analysed and it worked out well. The example section of NVIDIA provided an out of the box loading algorithm which can be used in terms of *.raw data and the loading worked perfectly.

In this work a simple and new approach has been introduced to select data based on a abstract representation of the voxel data. GVDB provides different abstraction levels which have been used in order to select parts of the voxelized model interactively using the three axis. The user had the option to move with the mouse and select a starting and an ending point along each axis to summarize six selections. The result would be a very simple selection process resulting in the selection of a specific part of the model which then would be highlighted. The selection process and the highlighted section of the data is depicted in Figure 4.12. The approach was unfortunately not used in the end because there was no option to apply transformations to only parts of the model which would have been needed in order to transform the data to a T-pose.

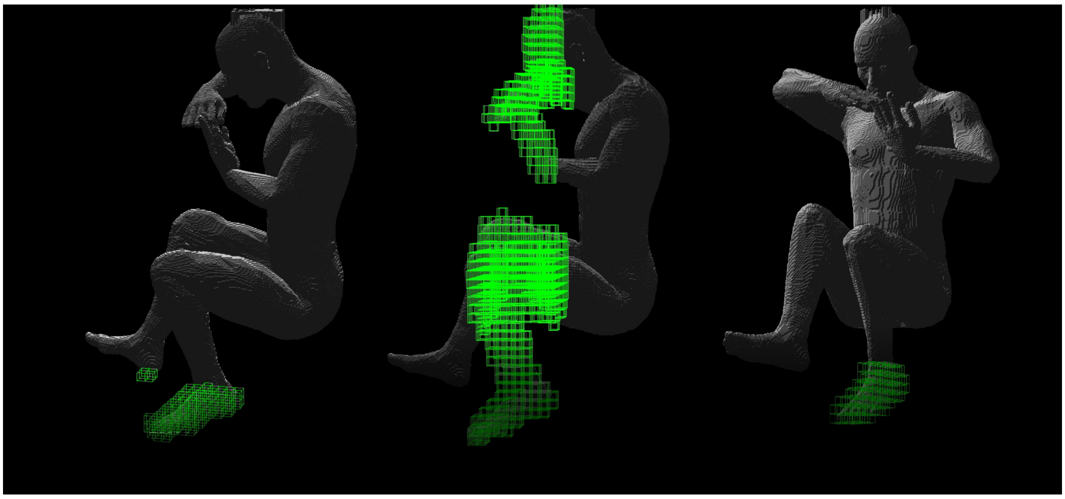


Figure 4.12: The left and the middle image represent the selection process of the one keypoint in the data namely the left foot of the phantom. The right image shows the selected foot represented in green. The green boxes are a representation of the data on a higher level in the GVDB database.

During the work on this master thesis different approaches have been tried to figure out how to automatically perform the rigging of the volumetric data. Unfortunately all approaches that have been tried did not deliver sufficient results. Although those attempts failed, the results of them are stated in this section in order to explain and show what went wrong.

4.4.3 Automatic rigging and animation of 3D characters [BP07]

The Pinocchio prototype introduced by Baran and Popovic is open source and available to download online [BP07]. As stated in Chapter 3 their approach needs a watertight mesh as an input and will automatically calculate the armature which is somehow based on a predefined skeleton. In the paper about the automatic rigging and automation of 3D characters the authors already mention that the discrete penalty functions used, are based on some assumptions which might lead to a unsatisfactory result when using their approach. Another downside of their algorithm is that it only works with mesh data and the data of fetal ultrasound investigations are normally given in voxel format. The transformation from the voxel data to a watertight mesh would lead to a loss of information and might not even be possible without human interaction. However, Shapiro et al. [SFW$^+$14] introduced an algorithm based on the Pinocchio prototype which works with a voxel model as input. Unfortunately their implementation is not publicly available.

The results when using the Pinocchio approach are depicted in Figure 4.13. The reason why this approach has not been progressed is that it seems that it will only work if the input data is somehow given in a standardized position.

4.4.4 Skeletonization

Another approach for the automatically rigging was to find a skeletonization representation of the data which may afterwards be mapped using joint mapping like introduced by Bharaj et al. [BTST12]. This approach would need a curve skeleton as an input which perfectly represents the volume and is not somehow to complex. There are different approaches how to skeletonize a mesh but when working with voxels the selection is restricted. Altought two approaches have been tried. One is based on a Matlab toolbox called Volume Skeleton Toolbox [Viz]. This toolbox provides different approaches for calculating the curve skeleton of volumetric data by thinning, distance field or potential field. When trying the approach with the distance matrix the result consisted of a huge amount of points. The potential field approach did not terminate and delivered a runtime exception and the thinning method somehow represented a skeleton but it was way too detailed. The result can be seen in Figure 4.14.

It has also been tried to generate the Skeleton in MeVisLab using the *DtfSkeletonization* module. The settings used to produce the skeleton shown in Figure 4.14 are the following: Object min value has been set to 1 which is the minimum of the object mask. The

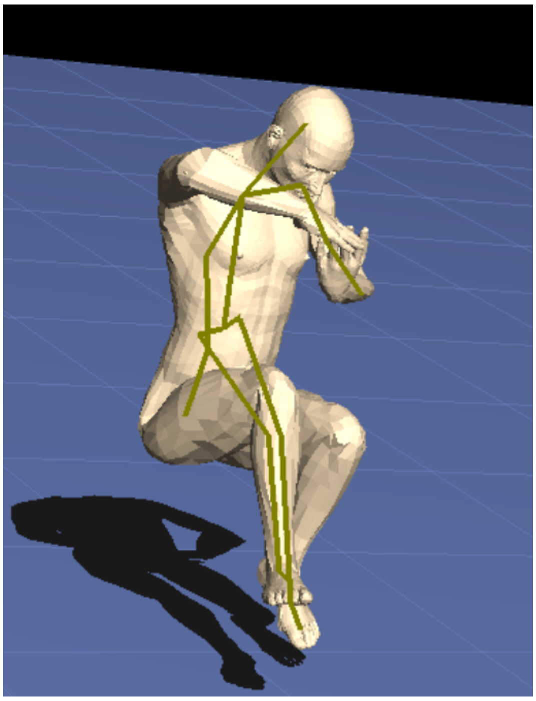
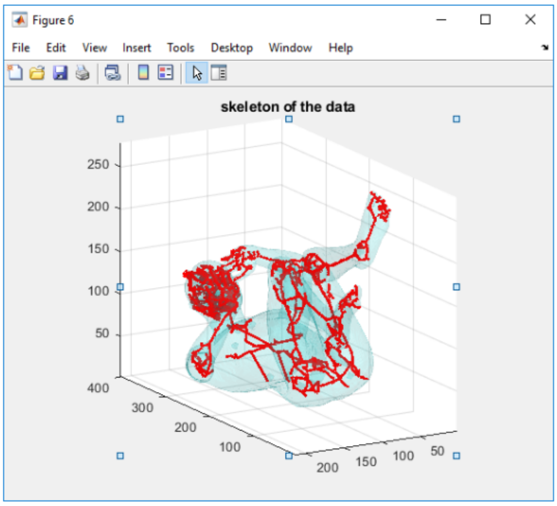


Figure 4.13: Result of Pinocchio prototype applied to the man phantom model in its mesh representation.

cavity volume has been set to 15 which represents the number of enclosed voxels to close cavities. The last setting is the maximal distance which has been set to 1000 meaning that the maximum erosion distance is 1000mm. It seams that there are some points of interest like the knee or other joints but it is also way too detailed and did not improve significantly when trying to use different settings in the module.

4.5 Weighting of the data

The weighting of the data is done by calculating the effect of each segment of the armature on the voxels surrounding the segments. For this processing step the Bender module *VolumeSkinning* is used. The module needs the original volume and the transformed armature as input and delivers an volume as output where the value of the voxels is set to the ID of the armature segments they belong to. The usage of this module is not intended to work with the data provided by the phantoms used in this thesis therefore unfortunately when using this module the option "Ignore Errors" has to be used. Normally the module would check if the armature is perfectly included in the volume and would not deliver an output if there are errors but this check does not work with the data provided here. Therefore the user has to be quite careful when using this module for the weighting process.



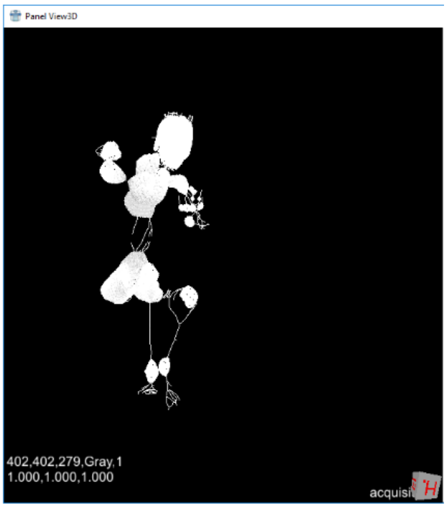


Figure 4.14: The left image shows the skeletonization result when using the thinning option in the Matlab Volume Skeleton Toolbox and the image on the right hand side represents the skeleton as a result when using the *DtfSkeletonization* module in MeVisLab.

There are also some alternatives which may be used in order to calculate the weighting e.g. the geodesic distance based approach by Dionne and de Lasa [DdL13]. Their approach is also voxel based and would maybe deliver better results than the one used in Bender where unfortunately the authors do not state which approach they use in order to calculate the weighting. In some cases like in the fetus phantom where the arms are quite stuck to the body the segments of the arm are in a conflict with the shoulders and the torso. In that case some artifacts might arise because the segments of the arms will have an influence on the voxels which should belong to the torso or to the shoulders. If such problems arise Bender also has a solution, namely the *Editor* module where the user is able to override the weighting result. But one has to mention that this process can be very complex and time consuming because the editor only works in 2D and therefore each slice has to be adapted.

4.6 Transformation of the data

The transformation of the data based on a model is the key feature introduced in this thesis. The transformation is done in 3D Slicer [PHK04, FBKC$^+$12] in an own extension with several modules that carry out the different operations needed in order to transform the data. The extension is portable and can be used for other purposes as well.

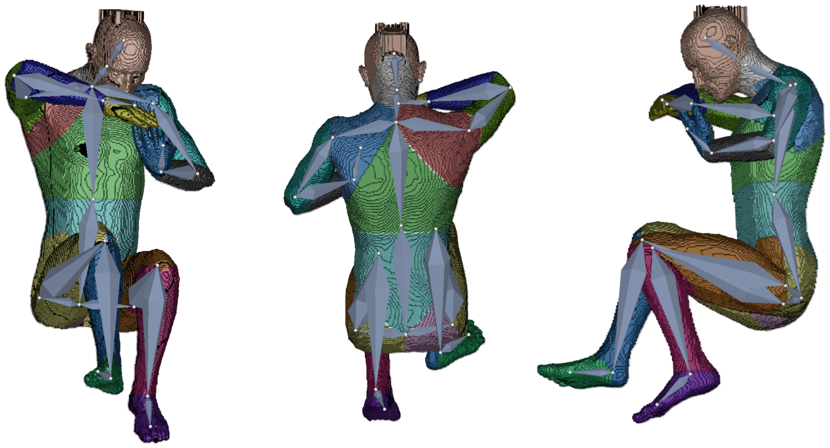


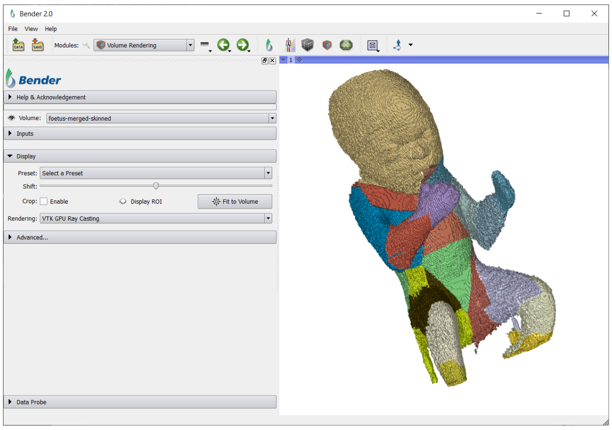
Figure 4.15: Result of the weighting module in Bender. The colors represent the mapping to the according segments of the armature inside of the volume.

4.6.1 Loading model and armature

The first step of the processing in this module is to load the data as well as the armature. The armature is saved as *.vtk which is an own file format of the visualization toolkit (VTK). It represents the armature as a graph which consists of several segments. Each segment is defined by a unique identifier and the segments are given by coordinates which state where the head and the tail of the segment are located in the coordinate system in 3D. The segments are always connected by their head except the first segment. The volumetric data is given in another file format namely *.mha which is a so called Meta Image. This file format is specified by the ITK and it contains the volumetric data. The input to this module is more specifically the output of the prior used Bender application which already includes the affiliation of the voxels to the segments of the armature by setting their value to the identifier of the segments of the armature. The data might be visualized using the $VolumeRendering$ module of the 3D slicer. Using it also reveals the mapping between the voxels and the armature. A visualization of an example output may be seen in Figure 4.16.

4.6.2 Calculate the transformation for each segment

The basic idea of the approach is to transform each segment of the armature in respect to the segment attached or in some special cases like the transformation of the legs in respect to another segment which is not connected. The transformation that is needed is calculated by thinking of rotating the segment in a way, that it faces into the same direction as the segment before. In order to visualize this transformation process the left arm of a phantom is used. In this example the forearm of the left hand shall be



Volume: foetus-merged-skinned

Preset: Select a Preset

Shift:

Crop: □ Enable ○ Display ROI

Rendering: VTK GPU Ray Casting

Figure 4.16: Output of the $VolumeRendering$ module of 3D Slicer when visualizing the weighted fetus phantom model.

transformed in a way that it faces in the same direction as the left upper arm. The left hand is just visualized in order to make the arm situation better perceivable. Of course the left hand also has to be transformed in order to align with the left forearm. The visualization of the initial pose is depicted in Figure 4.17. The legs are a somehow special case, because they should point in the same direction as the spine and not as the abstraction of the pelvis does. The artifacts at the armpit originate from an weighting issue between the upper arm and the torso. Considering the fact that the overall goal of this thesis is to transform the fetus to a T-pose to easily assess the measurements head to toe and finger to finger span, those artifacts have not been in focus of this thesis.

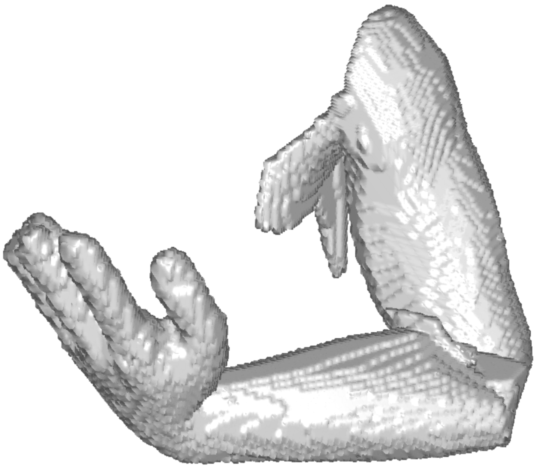


Figure 4.17: Visualization of the initial pose of the left arm before transforming it.

In order to make the problem definition perceivable from a mathematical view the forearm is shifted to the head of the left upper arm. This represents a positioning of the coordinate system to the head of the left upper arm and shifting the left forearm to the origin of the new coordinate system. This does not have to be done in that way during the implementation it is only a way to visualize the methodical approach presented in this work. You can see the shifting in the Figure 4.18.

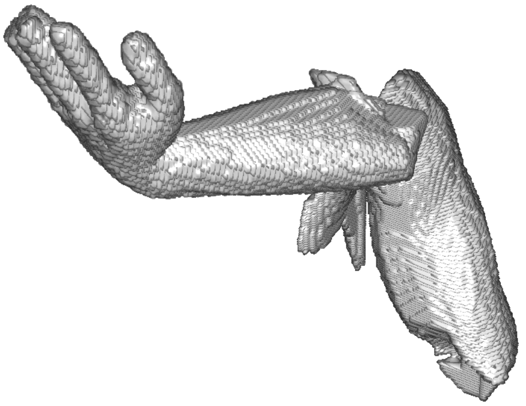


Figure 4.18: Image of the left forearm and hand shifted to the origin of the left upper arm in order to visualize the coordinate system shift.

Mathematical basis

For better understanding the problem is visualized as two vectors using Matlab. Figure 4.19 represents the output of the vector visualization.

The result of the transformation for most of the segments should be that they face directly in the same direction. To do so two calculations have to be done. First an axis of rotation has to be found and second the angle between the two vectors. The angle of rotation is calculated by the Equation 4.1 where u and v are the two vectors and a is the axis of rotation.

$$a = u \times v / ||u \times v|| \tag{4.1}$$

The angle between the vectors is calculated by Equation 4.2

$$\alpha = arccos(u \cdot v) \tag{4.2}$$

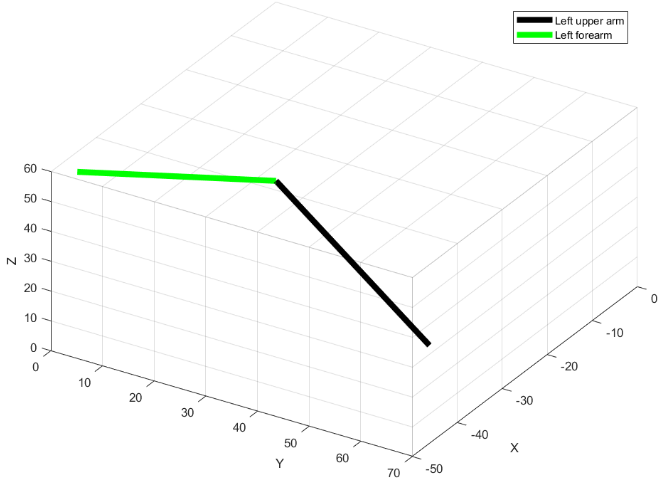


Figure 4.19: Vector representation of the left arm in Matlab, black being the left upper arm and green the left forearm.

Having this information given one can easily calculate the rotation matrix about this arbitrary axis using the information provided my McDonald in his paper [McD06]. Before representing the rotation matrix two further definitions have to be made, namely $c = cos(\alpha)$ and $s = sin(\alpha)$. The rotation matrix is presented in Equation 4.3.

$$M_r = \begin{bmatrix} a_x^2(1-c)+c & a_xa_y(1-c)-a_zs & a_xa_z(1-c)+a_ys \\ a_xa_y(1-c)+a_zs & a_y^2(1-c)+c & a_ya_z(1-c)-a_xs \\ a_xa_z(1-c)-a_y & a_ya_z(1-c)+a_xs & a_z^2(1-c)+c \end{bmatrix} \tag{4.3}$$

McDonald states in his paper that it is important to have a as a unit vector because the matrix would not represent a rotation if it is not [McD06]. The fact is that the rotation would be multiplied with the angle of rotation and therefore the result would not be the desired one. In Equation 4.4 the result of the operation is shown in order to rotate a vector in the same direction a given goal vector.

$$v_r = M_ru \tag{4.4}$$

Applying the rotation matrix to the given example the two vectors finally look like presented in the Figure 4.20.

The result in terms of data can be seen in Figure 4.21. It represents the left forearm rotated in direction of the left upper arm. The whole rotation is then carried out without moving the forearm to the head of the upper arm. Instead the center of rotation is at the head of the forearm which is in fact the connection point between those two segments.The

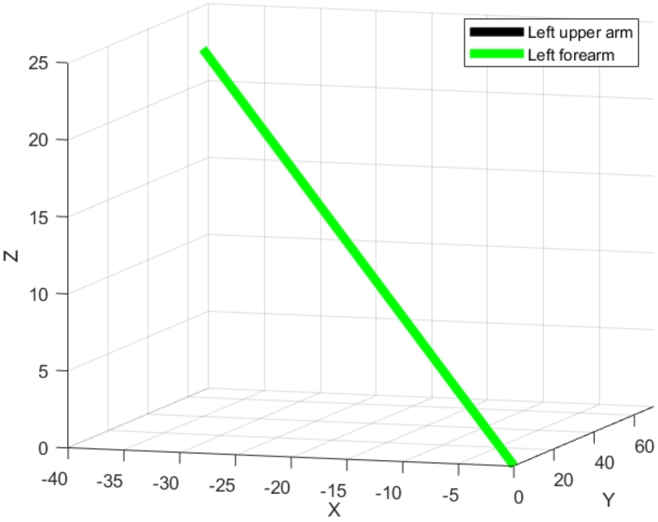


Figure 4.20: Visualisation of the rotation result in Matlab. The black and the green vector representing the forearm and the upper arm are perfectly aligned.

result of the rotation in that point is depicted in Figure 4.22. The artefacts faced in this example originate from the weighting of the voxels. In this thesis a heat map weighting approach has been applied. In order to create an anatomical correct transformation, restraints should be applied in the future.

4.6.3 Apply transform

Having the rotation matrix defined which rotates one segment in the direction of the other segment the next step is to apply this transform to the voxels which belong to it. There is also one detail which should be kept in mind. The rotation matrix only describes the rotation around the arbitrary axis but the origin of the axis has to be set to be at the joint between two segments. This step may be performed by first shifting the segment of interest to the origin of the coordinate system, apply the transform there and shift it back. Or another possibility is to apply the transform considering a specific center of rotation. Some transformations are able to take such a center into account.

Integration in visualization framework

In context of working with 3D visualization ITK is well known. Therefore we decided to work with it in terms of applying the transformation. The transformation is based on a given matrix and the center of rotation. Having both information the transformation is simply applied at the data using a *Resample* filter. The filter takes the original image and the transformation and creates the output volume. We work with a forward transformation provided by the ITK framework. Another possibility would be to use the

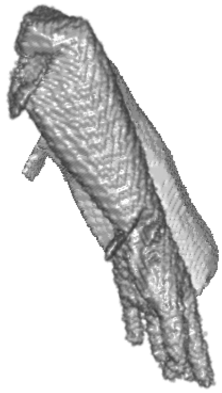


Figure 4.21: Result of the rotation of the left forearm with the center of rotation at the head of the left upper arm. Of course the left forearm has been moved to this point before.

inverse transformation as shown by Correa et al. [CSC10] which may show better results.

4.6.4 Save output

t The output of the method can be described as a volume which is a collection of volumes which represents the different parts of the armature. The volume can be saved as one file but the information about all the segments is also available. So for instance if one would like to examine the hand or the forearm of the fetus in greater detail there would also be the possibility to save those separately. In this work the save functionality of 3D Slicer is used which may be found in the menu of the program. A screenshot of the program is depicted in Figure 4.23.

4.7 Analysis

The analysis of the data may be performed in many different ways. There could be of course some automatically performed analysis steps which may be applied on the generated output data. Those could for example calculate or display the range of the volumetric data in all three dimensions. This would then correlate to the crown toe distance and the spread of the arms. Normally one would measure these parameters by viewing at a 2D slice of the ultrasound like described in Section 2.2.3. Due to the

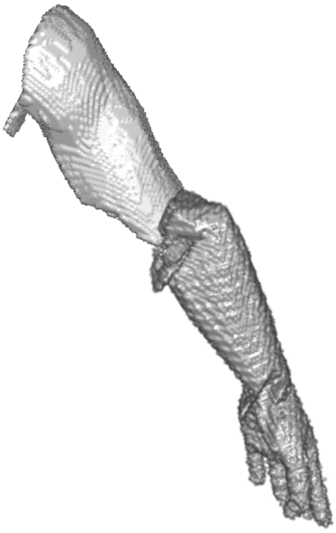


Figure 4.22: Result of the rotation of the left forearm with the center of rotation at the connection point between left forearm and left upper arm.

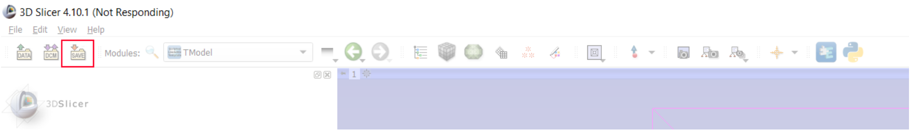


Figure 4.23: Saving option in the menu of 3D Slicer.

fact that the method analyses and transforms each segment separately the volumetric information of each part is already given for analysis. So if it is needed there could be also volumetric statistics applied like how much volume does the head or some other parts of the body have. It can even be automatically compared if the volume of the right or the left limb is significantly higher. This may be the case in some growth disorders.

In this work automatically analysis of the output data is not performed but there is a way to take measurements manually. 3D Slicer provides the user with a set of analysis tools like a ruler. The ruler can be applied in the 3D view and delivers the exact distance between two points. If the ruler is applied at the crown and the toe it will deliver the crown toe distance. Another more common distance would be the crown rump measurement. In Figure 4.24 the ruler can be seen in the toolbar and in Figure 4.25 the visualization in the 3D view of Slicer is shown. The ruling is performed interactively and can be adapted in the 3D view. Having a look at the result the artifacts prior discussed are clear to see. The artifacts at the joints like the hip, the armpit and the elbow all originate from the used weighting method. In respect to the overall goal of this thesis to support he head to toe and the finger to finger span distance measurement, the handling of these artifacts have not been in scope.

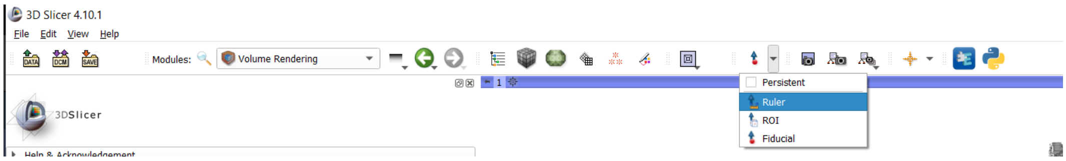


Figure 4.24: Ruler option in the 3D Slicer menu.

4.7.1 Validating the solution

The introduced method transforms a fetus in a natural pose to a standardized position. In order to validate the result some clinical reference data has to be found, where the goal is defined and can be compared to the result of the approach described in this work. Fetus ultrasound data normally is not given in such a position, therefore there are different approaches to validate the appraoch. One might use clinical data and take the measurements given from each part of the body of the fetus and sum them up to compare to the T-pose. On the other hand a practical solution would also be to use phantom data where the T-pose is already given and may be used as the goal to validate the result against. We decided to use a model of a human given in a T-pose to validate our method quantitatively and qualitatively. To load the model which was given as a mesh model the following steps had to be performed.

The phantom data represents a human in a T-pose. The model can be transformed into different fetus like poses using Blender [Ble19]. Blender enables the user to perform the rigging process and afterwards use the Pose option in order to generate different poses to be tested with the introduced workflow. The model used in this thesis has been found in the internet and is licensed by the Royalty Free License which allows all extended usages and can be fond here [The10]. It has to be mentioned that Blender works with a surface representation of the 3D models and this data would not be loadable. Therefore a processing step has to be performed to generate a voxelized model out of the mesh. The result of the voxelization can be seen in Figure 4.26. The artifacts on the head originate form the used software to create an image stack out of the .stl file. As presented in Figure 4.27 holes in the mesh are the source of the problem resulting in the artifacts at the head of the man.

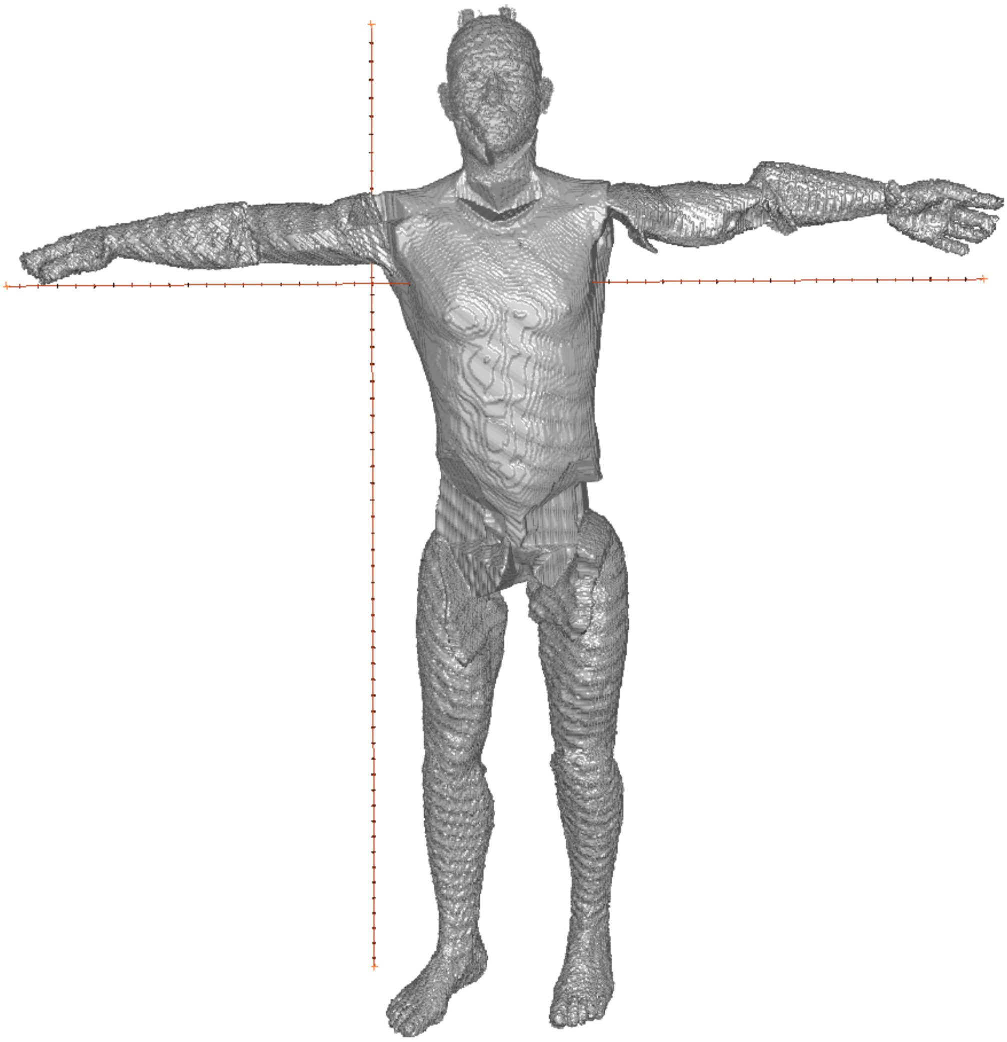


Figure 4.25: Result of two ruler operations on the T transformed model measuring the crown toe length and the spread of the arms.

STL generation

The first step towards a voxelized model representation of a mesh is to export the mesh in the *.stl format which is supported by Blender off the shelf. One only has to select the option export as stl and this step is already done.

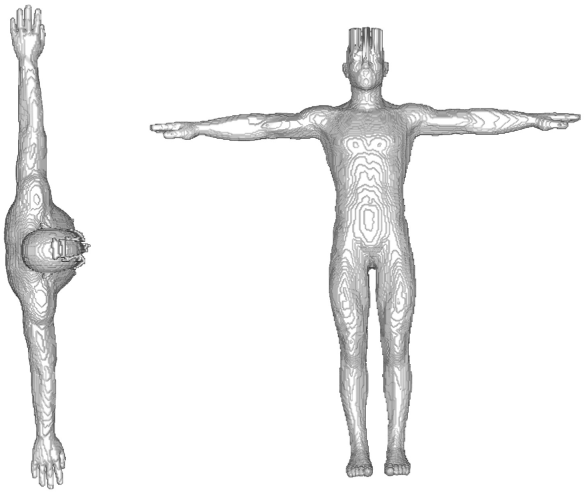


Figure 4.26: The phantom model dataset obtained from the internet [The10] after creating a voxelized representation of it. The artifacts are a side effect of the stl-to-voxel operating caused by a not perfect mesh quality.

STL to image stack

After having generated the *.stl file of the mesh model the next step is to convert it into an image stack. During the work on this thesis a public available python script has been used namely $stl - to - voxel$ [Ped15]. This Python script transforms the *.stl mash format in a stack of images which can be interpreted as voxel values. The transformation into an actual 3D volume file is done in the next step using another tool. The python script has to be run with the *.stl file and a folder as input. It then generates the images. It might be useful if the resolution is increased by altering one line of code in the file *stltovoxel.py*. The line number 99 defines the resolution of the image in x and y axis. The default value is 100 and for the data used during this thesis 400 seemed to be an appropriate value. An example input and output situation of the script is shown in Figure 4.27.

Convert image stack to *.raw file

The last step in the phantom data generation is to convert the image stack given as an output from the $stl - to - voxel$ python script into an actual volumetric dataset. This step is performed using the tool ImageJ [Eco18]. ImageJ is an open platform for scientific image analysis. The tool provides an import for the given image stack simply by selecting the option *File Import* and *ImageSequence* and selecting the first image of the image stack. The *.raw file can be saved by selecting the option *SaveAs* and *RawData*.

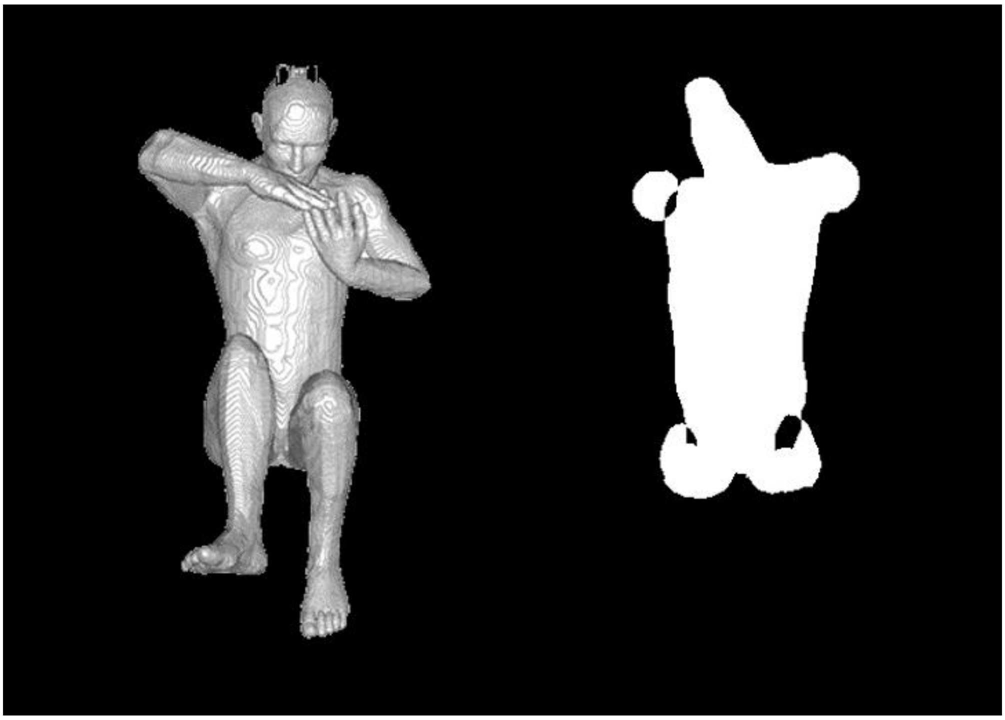


Figure 4.27: Example of the $stl-to-voxel$ python script showing the mesh representation on the left and one image of the image stack on the right.

CHAPTER 5

Implementation

In this chapter the implementation details of the Vitruvian Baby will be depicted. Figure 5.1 shows an overview of the implemented modules and the extension. The implementation has been done using Python and C++. The result is an own extension which can be used in 3D Slicer which consists of five modules.

5.1 Implementation overview

The main module is called *TModel* and carries out the procedure control. This module handles the input and the output of the data and calls the other modules which are implemented using C++. The first module in the workflow is the *Expand* module which takes care of enlarging the input volume. The second module is the threshold one. It is used in order to divide the input data by its components according to the segments in the armature. The next module is the logic for the transformation. It is invoked and provided with the IDs of the segments which should be transformed and calculates the needed angle as well as applies the transform to the input data. The *AddVolume* module is used to combine already transformed parts of the model together in order to generate the output at a glance. The workflow used to transform the data to a T-pose is pictured in Figure 5.2.

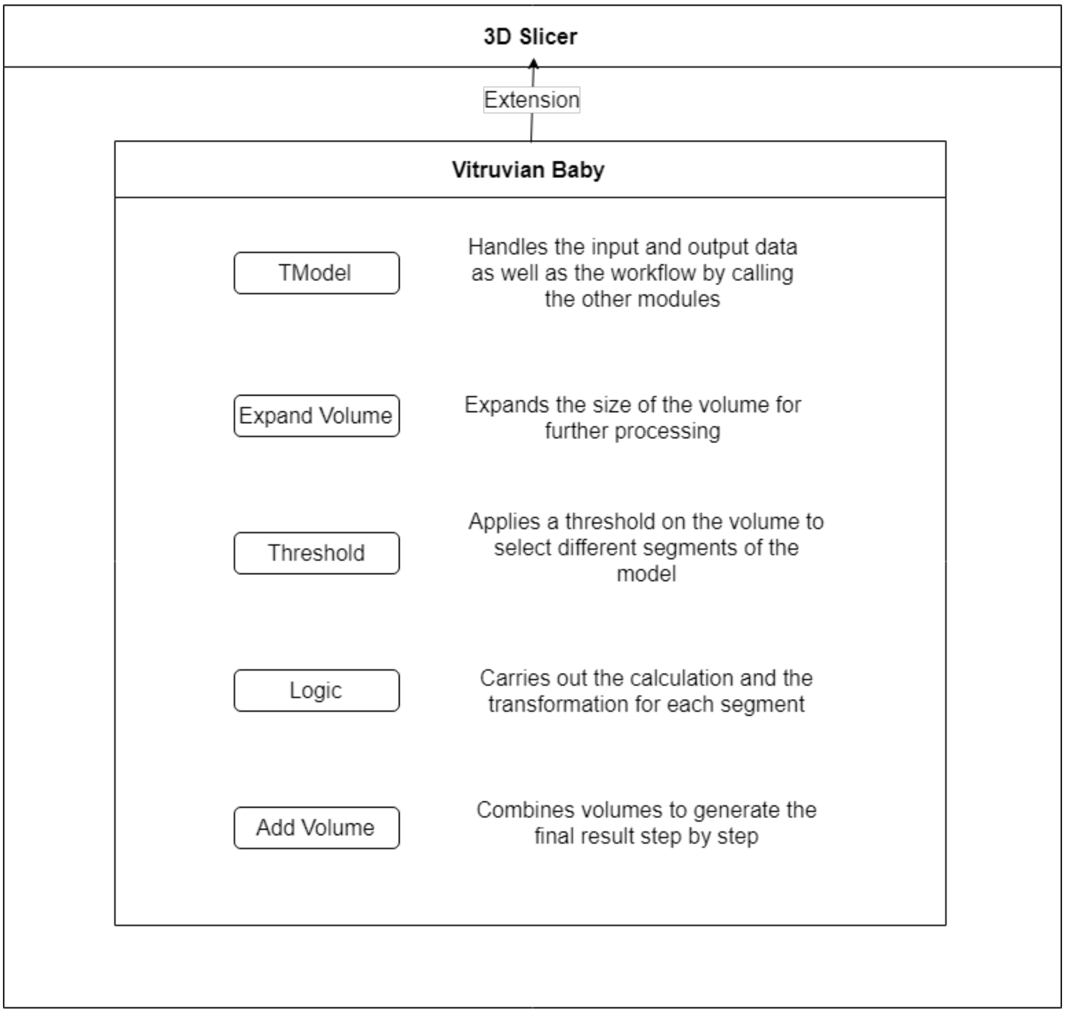

3D Slicer
Extension
Vitruvian Baby
TModel
Handles the input and output data as well as the workflow by calling the other modules
Expand Volume
Expands the size of the volume for further processing
Threshold
Applies a threshold on the volume to select different segments of the model
Logic
Carries out the calculation and the transformation for each segment
Add Volume
Combines volumes to generate the final result step by step

Figure 5.1: Overview of the implmenetation including information about the encapsulation.

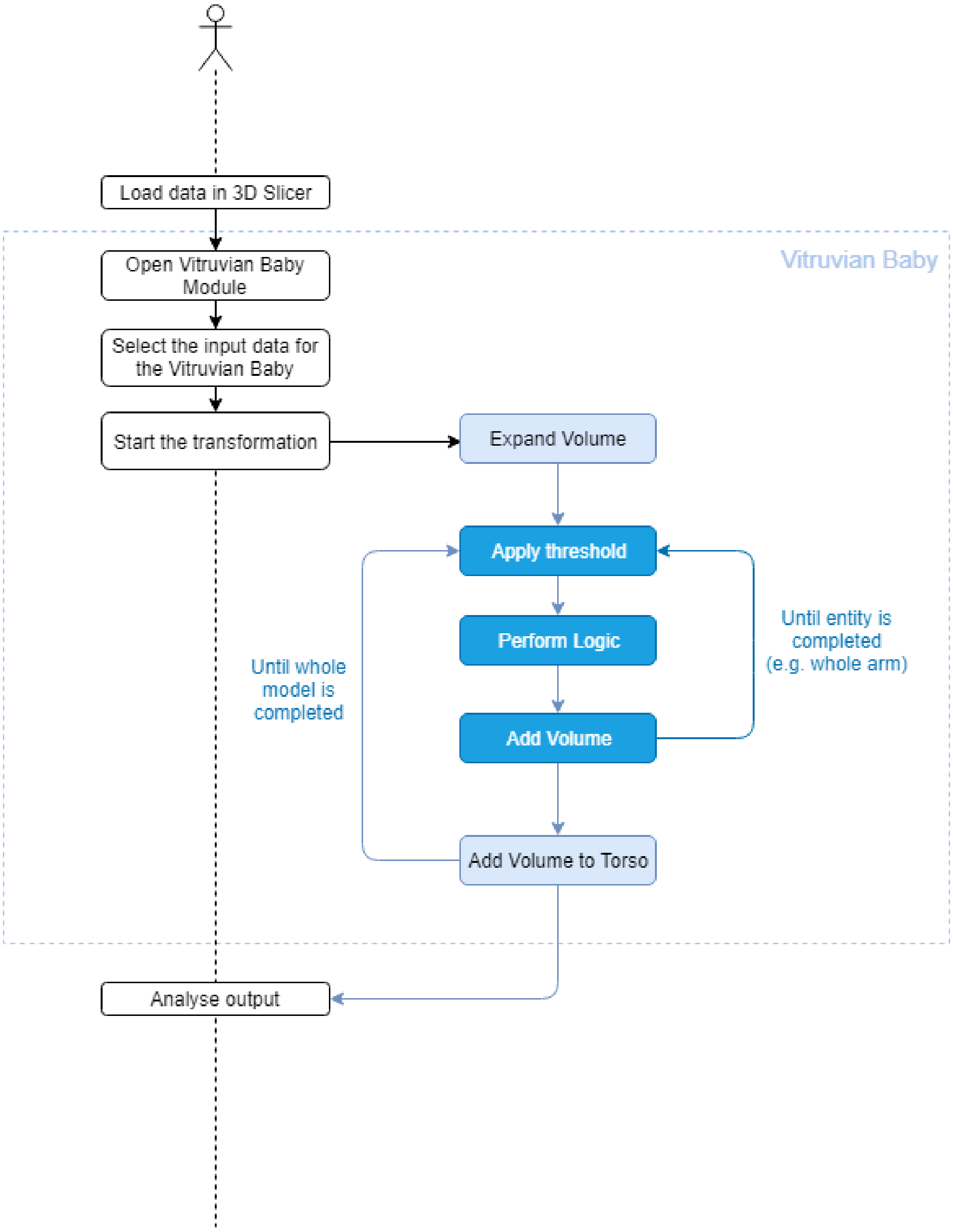


Figure 5.2: Workflow of the module including user interaction and the work done in the module.

5.2 3D Slicer extensions and modules

Implementing a 3D Slicer extension is supported very well by the software by using the *ExtensionManager* integrated in the software. The manager can be used to create and edit extensions. An extension is like a container which encapsulates the modules included. Extensions can also be shared and uploaded for other uses to work with them. In terms of modules different types have to be discriminated. In this thesis the types scripted module and command-line interface (CLI) module have been used. The scripted module is implemented in Python and is compiled directly in the software. It does not even have to be restarted. The CLI modules are implemented in C++ and provide more functionality and can also be called independently using any command line tool. In order to program CLI modules one has to build the 3D Slicer distribution on his or her own machine. Instructions about how to perform this task may be obtained from the official 3D Slicer website [PHK04].

After having set up the environment the 3D Slicer [PHK04] extension can simply be compiled and it will automatically produce a 3D Slicer version with the new extension included. Each part of the extension namely the modules are also available in the menu of the 3D Slicer program and can be tested and used independently. The program provides options for input and output handling and the user does not have to do any user interface design, although it would be possible to provide it. 3D Slicer is a very convenient platform to program for because it already includes basic implementations and libraries which are very useful when working with 3D data and also in terms of medical data handling. There are also numerous online tutorials on how to create and work with extensions and modules in 3D Slicer.

5.3 The Vitruvian Baby

The *VitruvianBaby* extension consists of a number of modules which are explained here separately. The first one which will be described is the *TModel* which is the main module and calls the other modules to perform specific tasks.

5.3.1 TModel

To interact with the method described in this thesis the *TModel* extension has to be used. It handles the input and the output data of the approach and delegates the further processing steps. It is implemented in Python and it handles the workflow of the transformation. The module handles the input and output of the other modules and calls them in order to perform the overall tasks. This module is a so called scripted module and has been created using the Extension manager of 3D Slicer and implemented using Python. On overview of the workflow order the transformation is carried out in this module is shown in Figure 5.3.

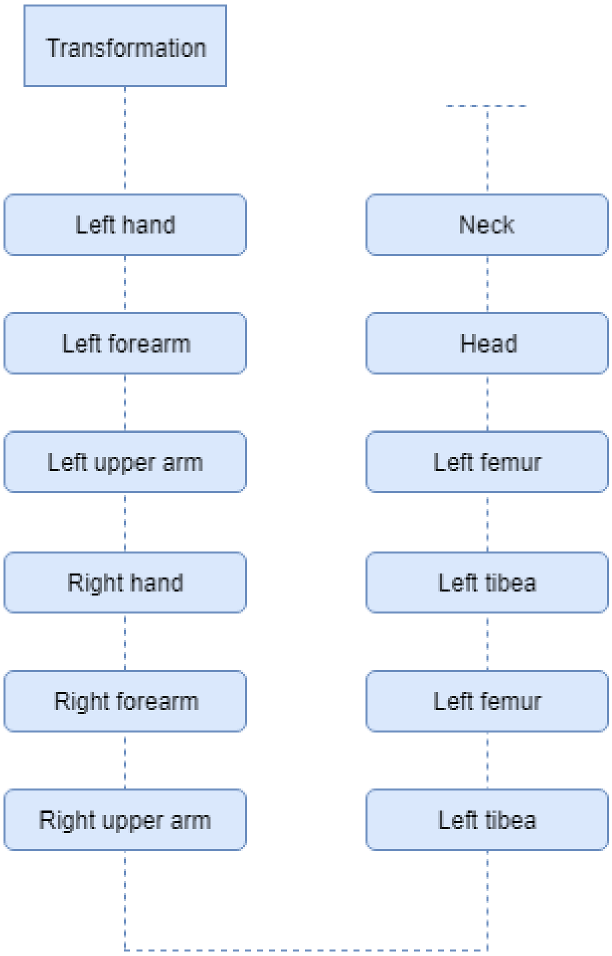


Figure 5.3: The transformation order of the model according to the armature parts.

5.3.2 Expand volume

The volumetric data is in the most cases given as a volume dataset where the data is tightly packed. Meaning that the amount of Voxels which represent air around the actual data representing the fetus is kept as little as possible. In the transformation extension of this thesis parts of the fetus are rotated and saved as an intermediate solution. In order to make these rotations possible without the limbs to be cropped by the size of the volume, the volume has to be expanded first. This step is performed using the CLI module called *ExpandVolume*.

The purpose of this module is to expand the input volume by a factor of two in each dimension. This task has to be carried out because the volume is normally tightly packed and there is not much "air" around the data. In this method the parts like the legs need some space for being rotated in the same direction as the body and a too small volume would lead to a cropping of the data. For this operation the ITK *ResampleImageFilter* and the *PasteImageFilter* are used. The input data is placed in the middle of the new volume by using the *PasteImageFilter* and afterwards the *ResampleImageFilter* is applied to create the output volume. The settings of the *ResampleImageFilter* reflect the doubled size of the input volume. In general each module has to read and write the image data. For these operations the ITK *ImageFileReader* and the *ImageFileWriter* are used.

5.3.3 Threshold volume

The data which is used as input to the method presented is already split by the segments of the armature used. Each armature part has an unique identifier. These identifiers are also used as value in the data to present the affiliation between voxels and segments. If one would like to extract only one part like the hand of the fetus he or she could use the *Threshold* module.

The *Threshold* module is also a CLI module. The module takes a volume as input and provides a volume with the same dimension and spacing as output. Other properties is the setting if the threshold should be calculated below, above or outside. For this thesis we only use the outside setting because we only need to gather specific parts of the volume which then are identified by setting the upper and lower threshold value. The linkage between the identifier and the part of the armature is shown in Table 5.1. One example of the input output situation is depicted in Figure 5.4. In this case the outside option is used with an above and below value of 21 which defines the left hand of the fetus model.

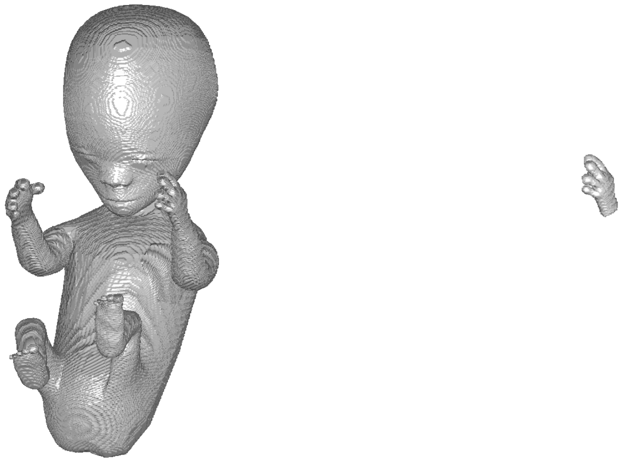


Figure 5.4: On the left side the fetus model as an input and on the right side the threshold output showing the left hand. The threshold method has been carried out using the outside option and a value of 21 for both above and below.

ID	Body part
2	Head
3	Nack
4	Left Shoulder
5	Right Shoulder
6	Torso
7	Belly
8	Left Pelvis
9	Right Pelvis
10	Right Tibea
11	Right Femure
12	Right Foot
13	Left Tibea
14	Left Femure
15	Left Foot
16	Right Upper Arm
17	Right Forearm
18	Right Hand
19	Left Upper Arm
20	Left Forearm
21	Left Hand

Table 5.1: The linkage between the unique identifier and the part of the body of the used armature.

5.3.4 Transforming data according to armature

The transformation of one part of the model is carried out by using the *Logic* module. The *Logic* module is responsible for the actual transformation of the data. The input is described as the volume representing the segment which shall be transformed, the ID of the segment which shall be rotated and the ID of the segment which is the fixed one, as well as the armature. One example would be to set the left hand as input, set the ID of the rotation segment therefore to be 21, according to Table 5.1 and the fixed part would be the left forearm and therefore 20 also given by the Table 5.1. The *Logic* module is also a CLI module.

The first step in this module is to load the armature using functionality provided by Bender [FOA+14] namely the *benderIOUtils*. These utilities are used to load the armature as a *vtkPolyData* which is an object of VTK. This dataset provides the information of the armature including the ID, the position of the head in 3D coordinates and the coordinates of the tail. First a standardized 3D vector is calculated pointing away from the connection point of the two segments. This vector is also normalized for further calculations by dividing the vector by its length. Normally the direction of the rotational part is calculated using the information about the fixed part provided by the user but there is one exception, namely the left and the right femur. Those segments are compared with the segment of the belly, because they should not point in the direction of the pelvis, where they are attached. They should point straight to the floor which is in fact the direction of the belly segment. The segments can be seen in Figure 5.5.

The rotation matrix which has to be applied is calculated according to the explanation in section 4.6. In order to apply this matrix to the given data the MatrixOffsetTransformationBase of ITK is used. This class takes a transformation matrix and a center of rotation as input and is applied to the given input data using the *ResampleImageFilter* of ITK.

5.3.5 Adding scalar volume

If the transformation of each part is carried out one would also like to put the data separated by the *Threshold* module back together. In order to carry out this task he or she may use the *AddVolume* module. This module takes two volumes as input and combines them to one output volume. The two input volumes must have the same dimensions and the output is defined as the sum of the input volumes. The task is performed by using the ITK *ConstrainedValueAdditionImageFilter* and the *ResampleImageFilter*. The addition image filter takes both 3D volumes as input and creates an output containing both volumes. The result is then produced by the *ResampleImageFilter* and written by the *ImageWriter*.

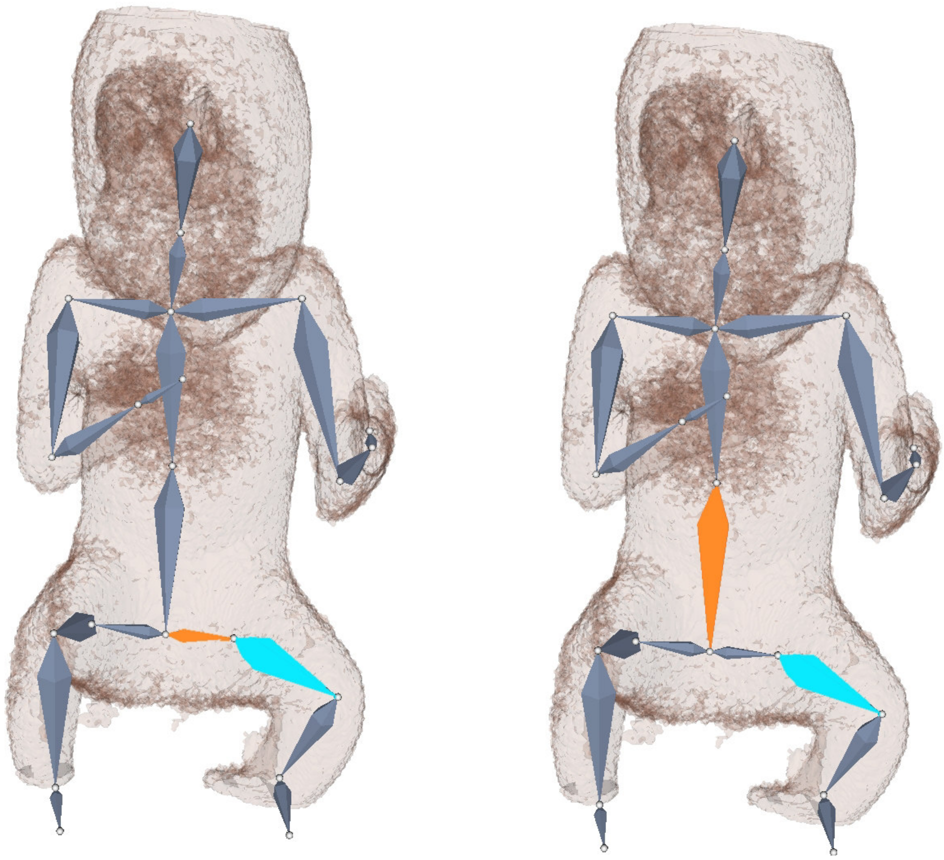


Figure 5.5: The hip is presented on the left in orange and the femur which shall be moved in blue. On the right side the belly segment is colored in orange. The image should represent why the femur should be moved in direction of the belly segment and not the hip segment.

CHAPTER 6

Results

The approach introduced in this thesis has been tested using different types of input data. One of the most interesting outcomes when testing new methods or approaches is to try to express the performance quantitatively and qualitatively. That is why we decided to test the method with a phantom dataset of a man, which has been gathered from the internet [The10]. The model is already in a T-pose and therefore the ground truth of how the output should look like. The model is adapted in Blender in order to bring it into different poses which reflect natural poses of a fetus in the belly. The input dataset is shown in Figure 6.1. The head is affected by some artifacts which occurred during the mesh to voxel transformation.

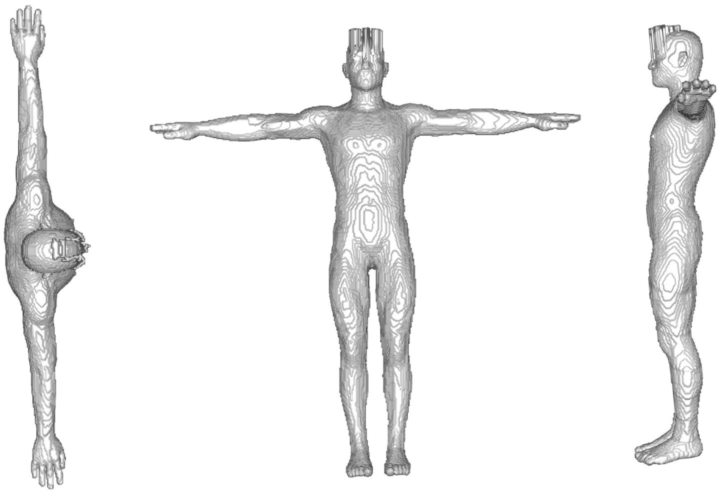


Figure 6.1: The initial pose of the phantom data used to evaluate the method against the standardized position [The10].

Fetal poses have a huge variation from case to case. Haultain and Ferguson [FH89] sketched them in their book about obstetric nursing in 1902. A sketching of different poses which might occur in the womb is shown at Figure 1.2. In respect to those poses presented in the book of Haultain and Ferguson [FH89] we have tried to reassemble relevant poses to perceive the performance of our method. The results are assessed in three steps. The first step is to measure the time needed for the manual step in the method namely the rigging. The second measurement is the overall overlapping of the voxels in the resulting T-pose and the goal T-pose. The last measurement is the similarity of the head to toe measurement and the span measurement from finger to finger between the resulting T-pose and the goal T-pose.

In the following we present the results of an analysis of the different poses in terms of performance. Therefore we show two charts which represent the qualitative and the quantitative performance of the method. In Figure 6.2 the needed time is shown and the used poses can also be seen. Figure 6.3 shows the performance in terms of overlapping voxels in percent between the given T-pose and the resulting T-pose of the described method.

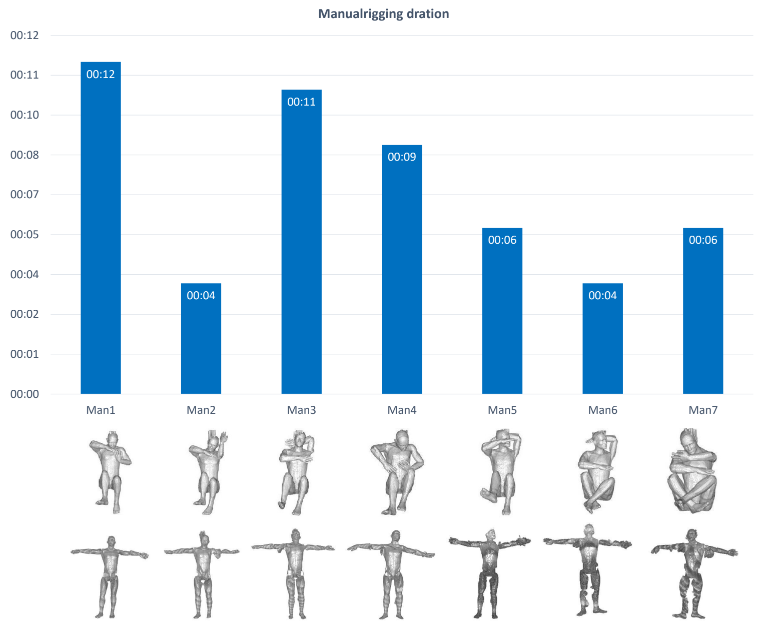


Figure 6.2: Chart representing the duration needed for rigging of the data in minutes. The legend shows which pose has been used as input.

One of most interesting measurements though is the difference of measurements of the ini-

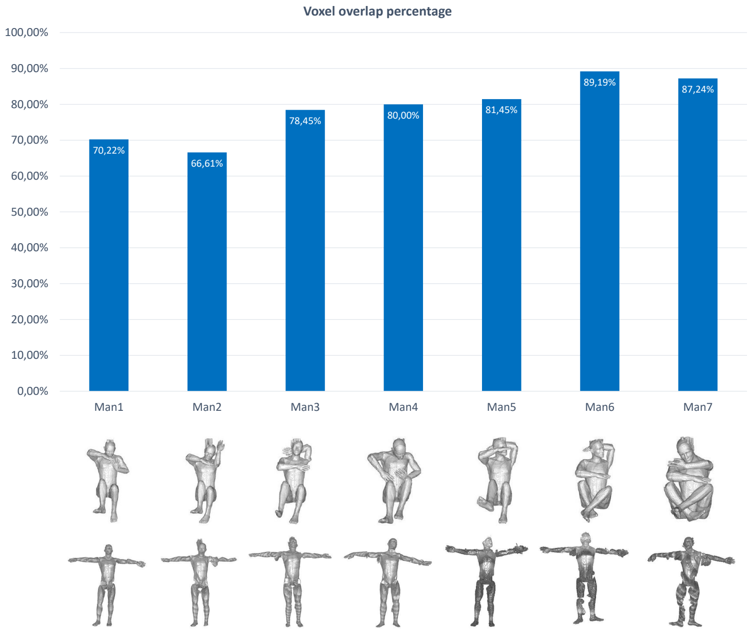


Figure 6.3: Chart representing the overlapping of the resulting T-pose and the initial T-pose in percent. The legend shows which pose has been used as input.

tial T-pose and the resulting T-pose. The major benefit of using our approach may be to automatically measure the fetus in terms of head to toe and finger to finger span measurement. Therefore we analyse the similarity of those measurements in percent. The results of the head to toe measurement similarity in percent and those of the finger to finger span are shown in Figure 6.4. All the quantitative results will also be discussed in Section 6.1.7.

Having a look at the charts presented it is very hard to find a clustering or grouping of poses which perform better then others. If we focus on the measurements all of the head to toe measurements are over 90% and the worse measurement of the finger to finger span is around 85%. When focusing on the rigging time it has to be said that the author worked through the examples in ascending order which might explain why the Man 1 model needed the most amount of time. On the other hand the overlapping is very hard to explain because the most folded example namely the Man 7 has one of the best voxel based overlapping. The next section 6.1 shows the results structured by first showing a volume visualization of the pose and the pose with the armature included. The last image then shows the resulting T-pose.

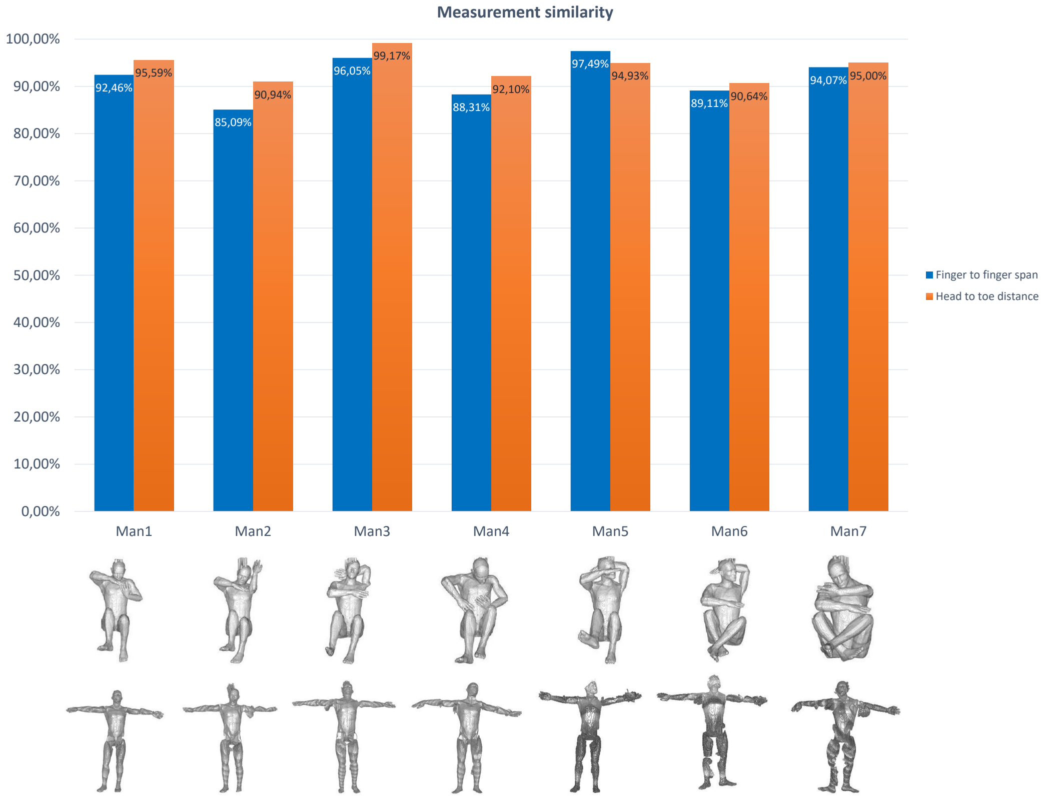


Figure 6.4: Chart representing the head to toe measurement similarity in blue and the finger to finger measurement similarity in orange, between the result of the method and the goal. The legend shows the results of the method and the input pose.

6.1 Images of the results

In this part of the thesis the results are depicted and described qualitatively. The first part is composed of the phantom models used to analyze the performance of the method. There are seven different poses which have been used. Afterwards the fetus models are shown.

6.1.1 First fetus pose

The first pose used to analyze the performance of the novel method is a very simple fetus pose. It can be classified as a breech position. The fingers are touching each other and the angle between the femur and the body is about 90 degrees. In Figure 6.5 the touching fingers can be seen and in Figure 6.6 the top view of the model is presented. In Figure 6.7 the angle between femur and body can be seen clearly.
Figure 6.8 shows the result of the unfolding process. The T-pose is very clear to see. Although the fingers touched each other in the beginning the separation delivered good results where they are clearly distinguishable. The angle between the femur and the body had a higher impact. The result shows that the reformation leads to a deformation in terms of the torso. A part of the torso is now attached to the leg where it should not be. Overall the T-pose looks very good and the deformation would not lead to major differences in the fetus measurements.

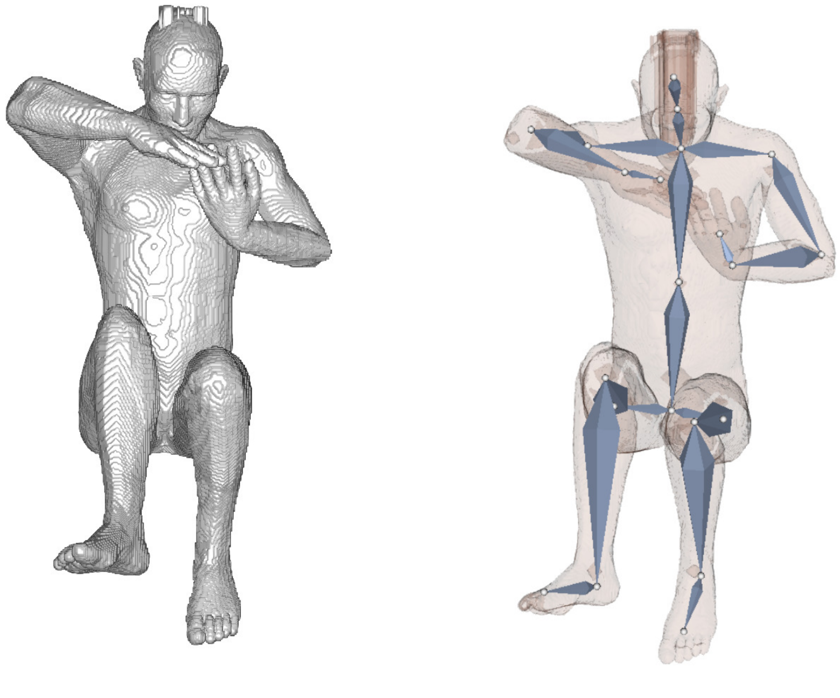


Figure 6.5: Front view of the first fetus pose. On the left the volume visualization of the prototype and on the right the visualization of the included armature.

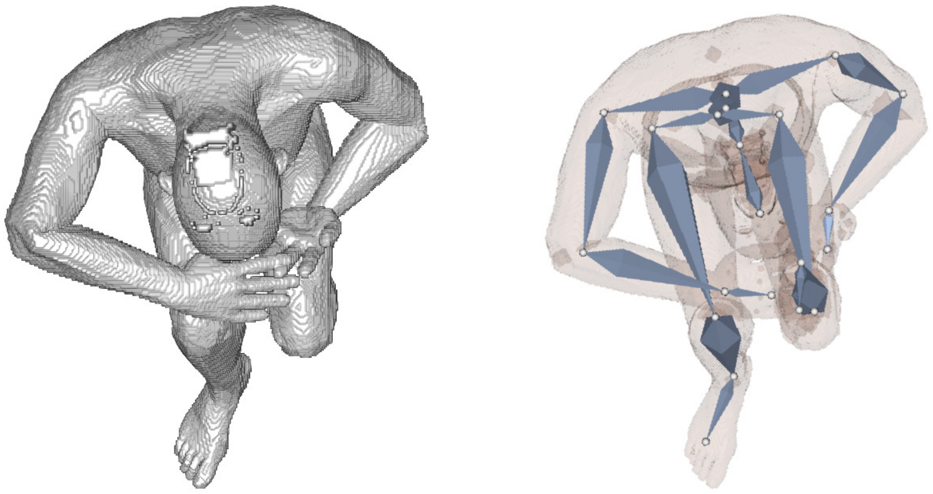


Figure 6.6: Top view of the same formation also volume and armature view.

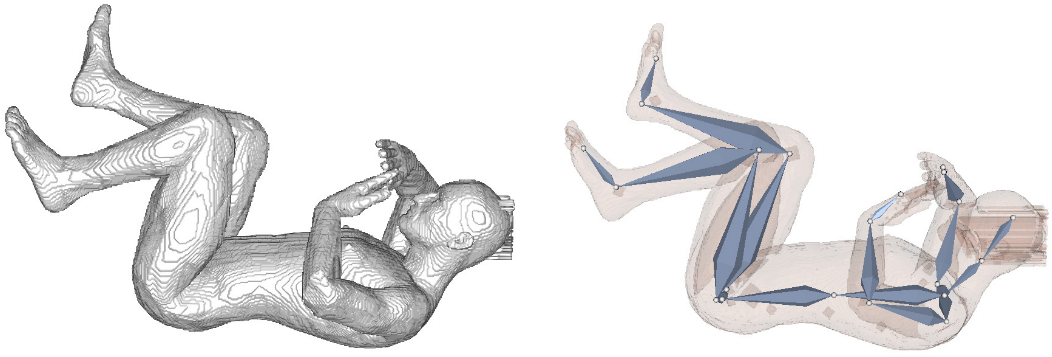


Figure 6.7: Side view of the pose also with visible armature and volume visualization.

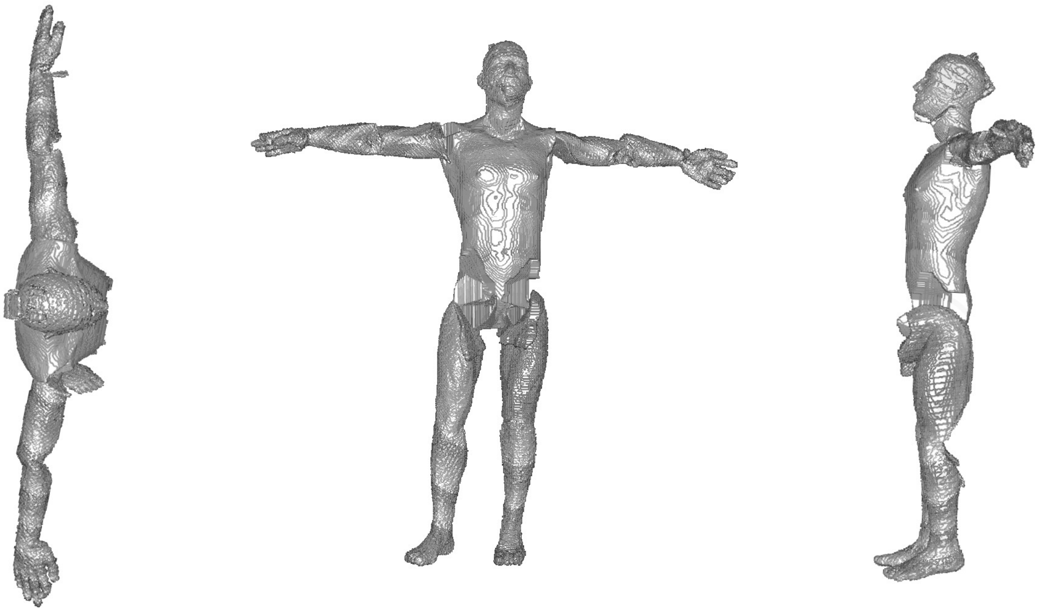


Figure 6.8: The result T-pose of the first fetus pose from left to right: Top view, front view and side view.

6.1.2 Second fetus pose

The second pose presented is very similar to the first one. Here we tried a different arm pose and the legs are very close to each other. In Figure 6.9 the arm pose can be seen and Figure 6.10 shows the top view of the model. A side view of the input is presented in Figure 6.11.

The result of the transformation is shown in Figure 6.12. Overall the T-pose is clear to see and it is a quite good result but there are some considerable deformations involved. First of all the head is split, second the left arm of the model took some part of the body with it. The legs are again taking some part of the torso with them. Although the standardized measurement as used in fetal diagnostics would still be applicable.

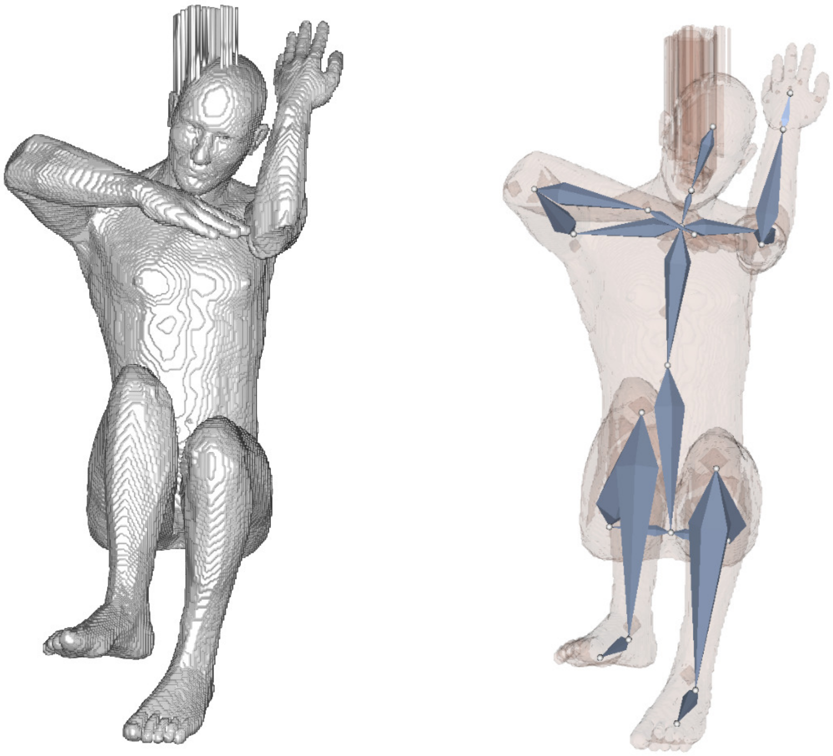


Figure 6.9: Front view of the second fetus pose. On the left the volume visualization of the prototype and on the right the visualization of the included armature.

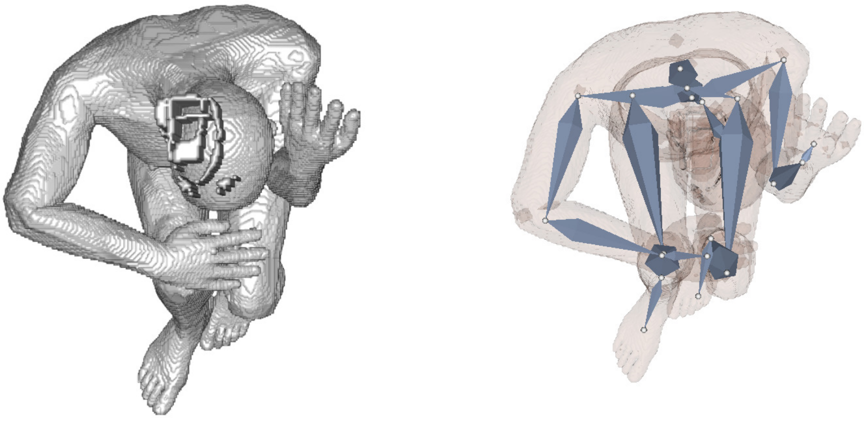


Figure 6.10: Top view of the same formation also volume and armature view.

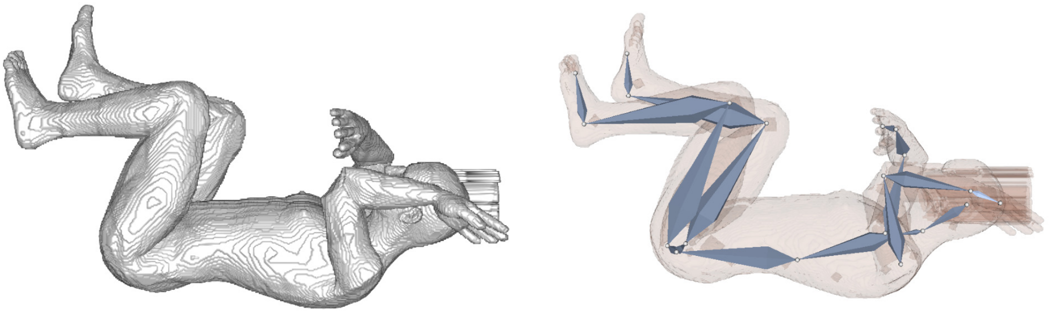


Figure 6.11: Side view of the pose also with visible armature and volume visualization.

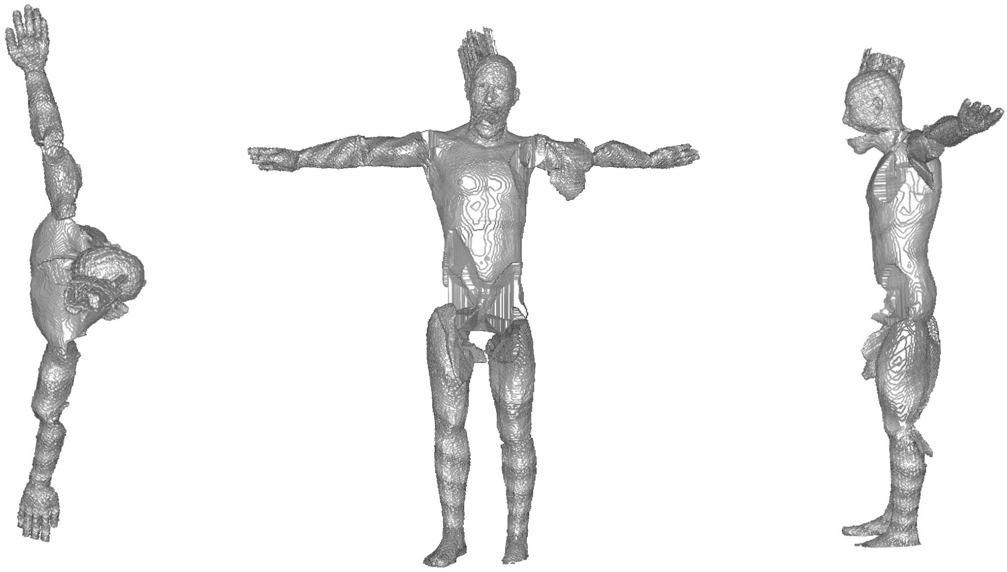


Figure 6.12: The result T-pose of the second fetus pose from left to right: top view, front view and side view.

6.1.3 Third fetus pose

The third position differs in the pose of the arms. They are resting before and behind the body. In Figure 6.13 the pose of the arms is clear to see. In Figure 6.14 the arms before and behind the body are shown. Having a look at Figure 6.15 the angle of the legs is visualized.

The resulting pose after the application of the method is shown in Figure 6.16. The T-pose is again very clear. Although the left arm touched the head it does not suffer of any deformations. The result shows similar deformations as the first and the second pose, namely the missing parts of the torso.

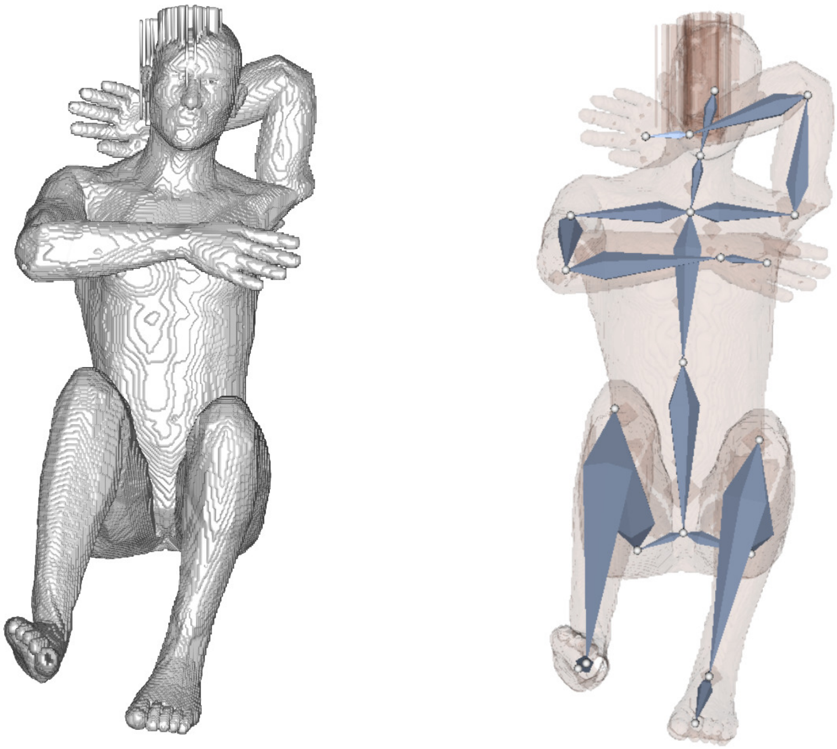


Figure 6.13: Front view of the third fetus pose. On the left the volume visualization of the prototype and on the right the visualization of the included armature.

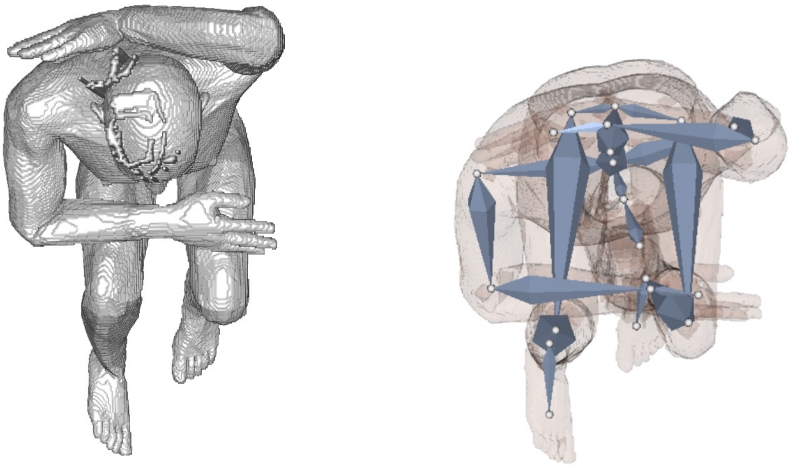


Figure 6.14: Top view of the same formation also volume and armature view.

Figure 6.15: Side view of the pose also with visible armature and volume visualization.

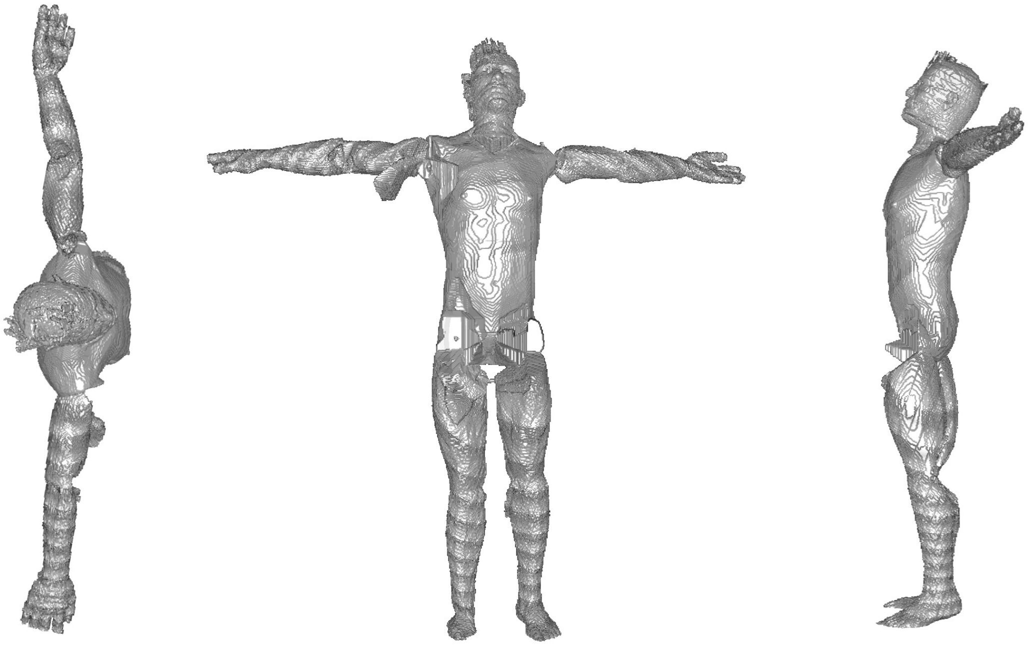


Figure 6.16: The result T-pose of the third fetus pose from left to right: top view, front view and side view.

6.1.4 Fourth fetus pose

The fourth pose is used to analyze a more buckled pose, where the arms are resting in front of the body and the head is also tilted forward. In Figure 6.17 the pose of the arms is shown. Figure 6.18 enables a view of the tilted head. Figure 6.19 shows a side view of the used pose.

Figure 6.20 shows the result of the transformation. The result is affected by the same deformations as the models before. In this result a not perfectly straight pose of the arms can be seen. They are more pointing downwards than straight. Overall the T-pose can be seen clearly.

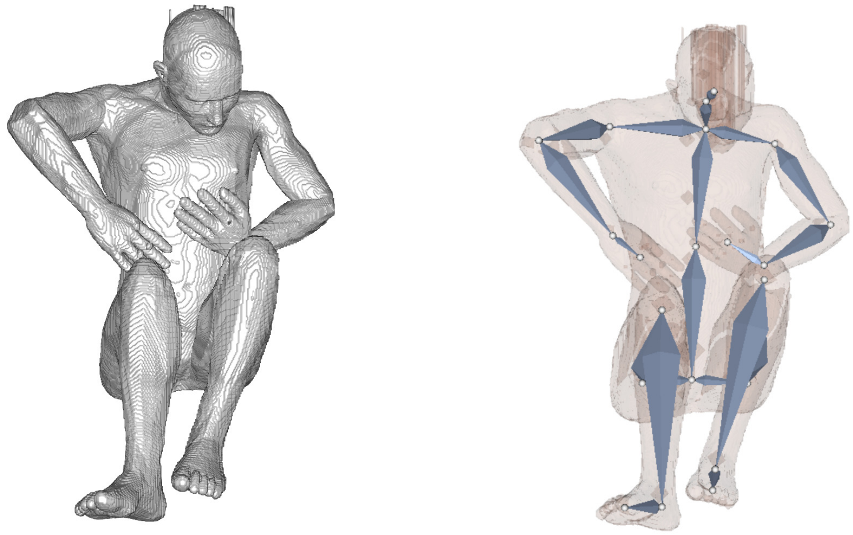


Figure 6.17: Front view of the fourth fetus pose. On the left the volume visualization of the prototype and on the right the visualization of the included armature.

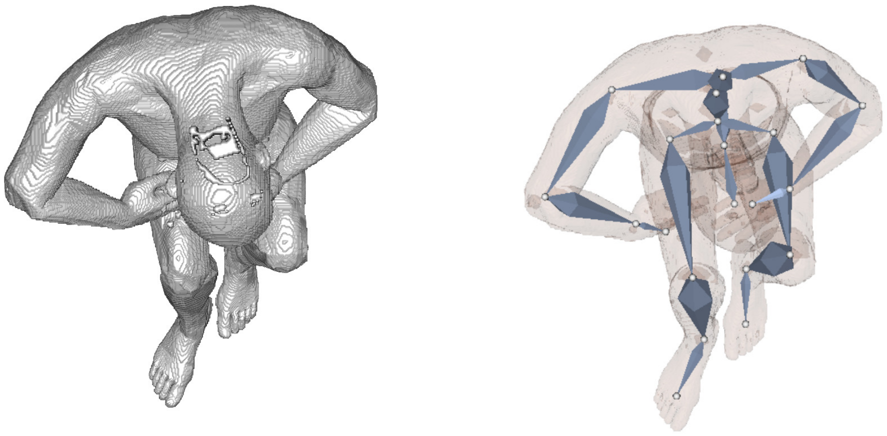


Figure 6.18: Top view of the same formation also volume and armature view.

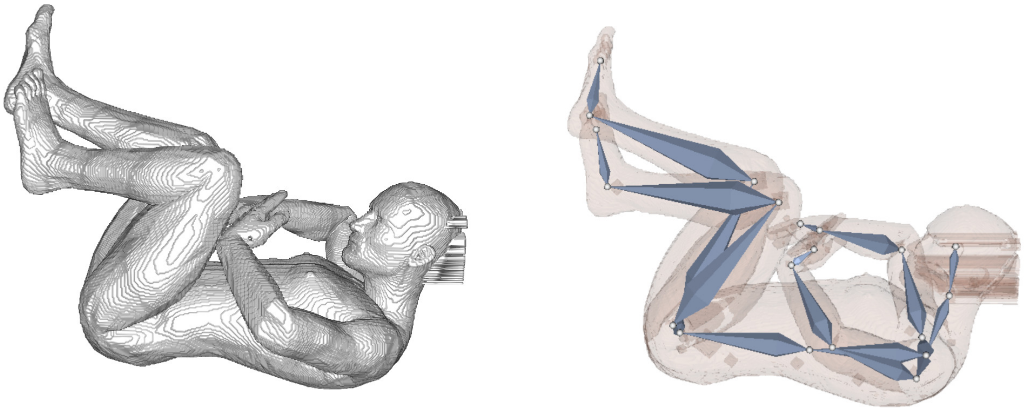


Figure 6.19: Side view of the pose also with visible armature and volume visualization.

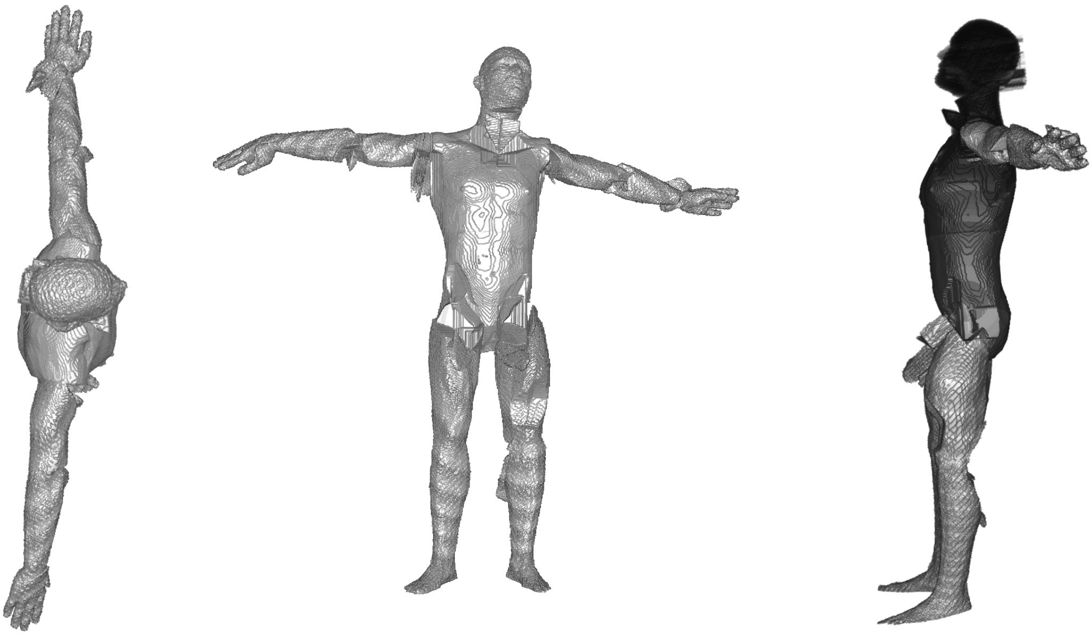


Figure 6.20: The result T-pose of the fourth fetus pose from left to right: top view, front view and side view.

6.1.5 Fifth fetus pose

The fifth pose is used to try different angles for the arms and the feet. The left arm of the model is touching the head and the right tibia has another angle than the left one. The angle of the legs can be seen in Figure 6.23. The touching arm is visualized in Figure 6.22 and the overall pose from the front is shown in Figure 6.21.

The result after the transformation is displayed with Figure 6.24. In this example the transformation of the left arm lead to a deformation of the head. This is a bad effect because the head circumference measurement would definitely be affected by it. Overall the T-pose is well achieved.

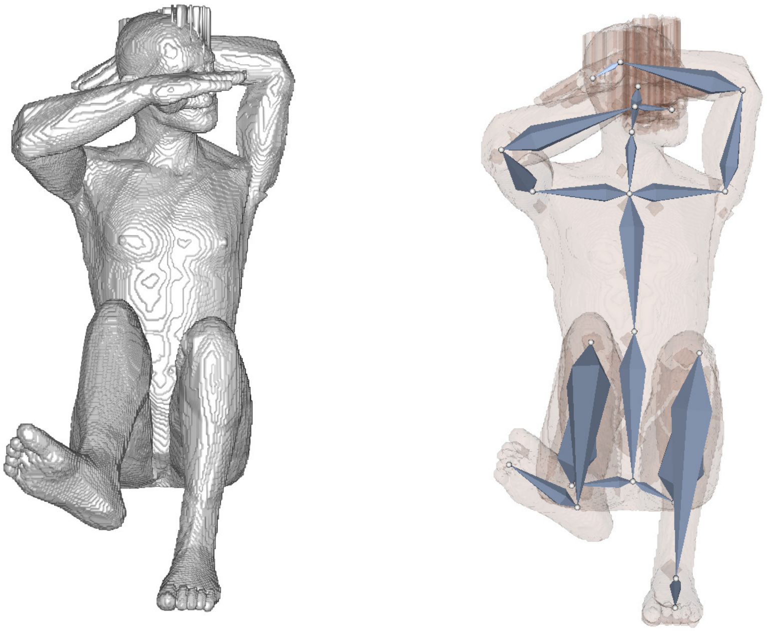


Figure 6.21: Front view of the fifth fetus pose. On the left the volume visualization of the prototype and on the right the visualization of the included armature.

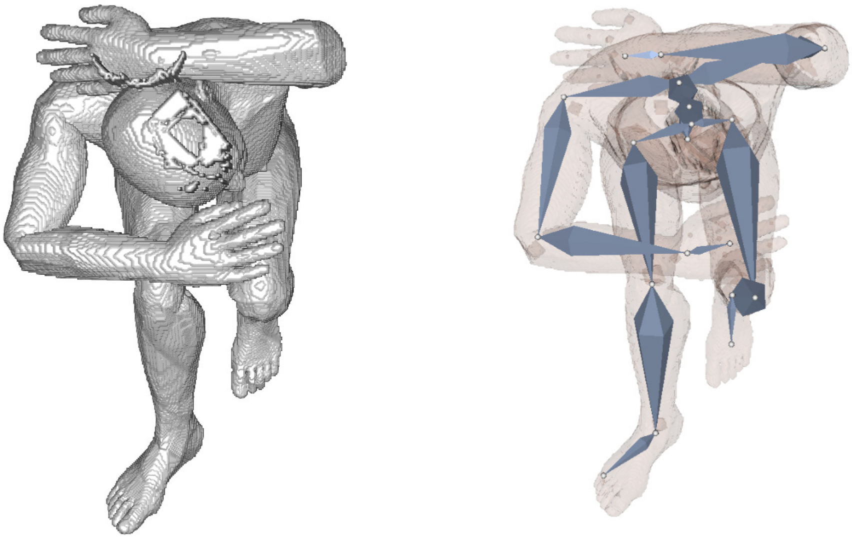


Figure 6.22: Top view of the same formation also volume and armature view.

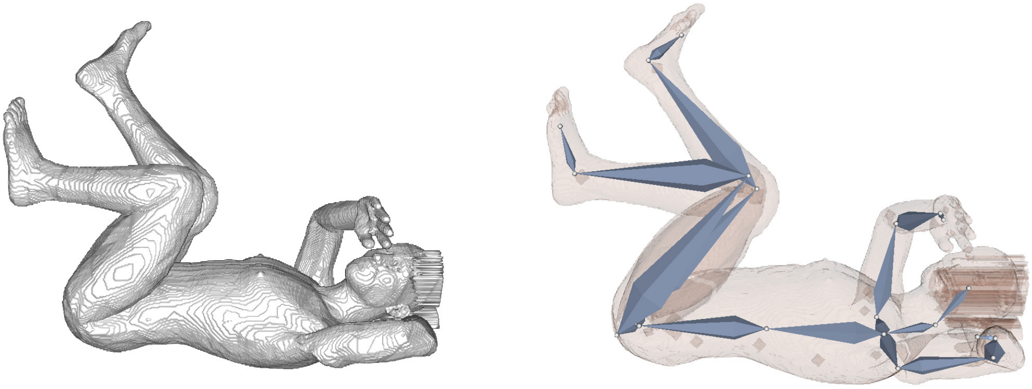


Figure 6.23: Side view of the pose also with visible armature and volume visualization.

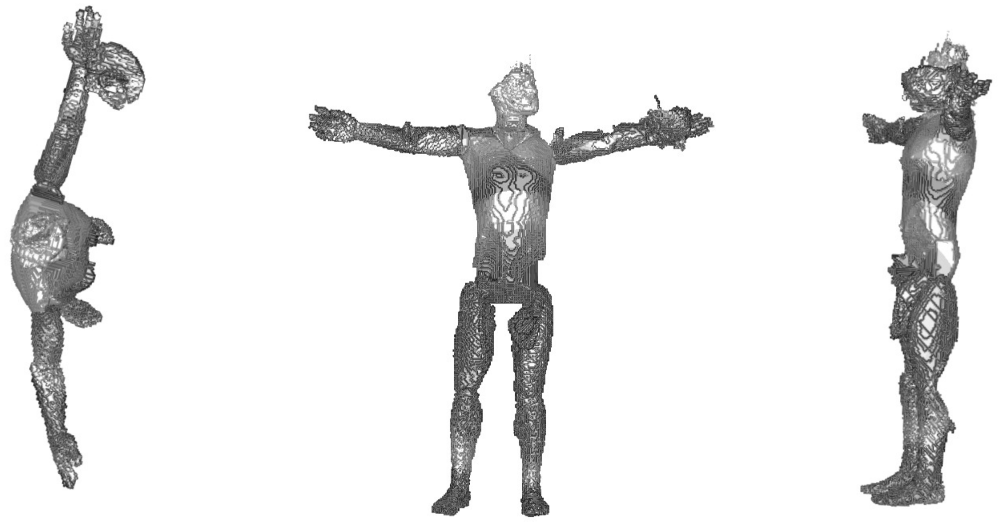


Figure 6.24: The result T-pose of the fifth fetus pose from left to right: top view, front view and side view.

6.1.6 Sixth fetus pose

The sixth pose is also a breech pose, where the legs are crossed. The arms are similar to the fifth pose and the head is pointing to the left. Figure 6.25 shows the frontal view of the model and Figure 6.26 represents the pose of the arms. The side view is presented in Figure 6.27.

The result of this pose is presented in Figure 6.28. There are some issues with this model. The left hand takes some parts of the head. The right leg looses a bit the connection between tibia and femur. On the other hand the the T-pose is also very well represented by this result.

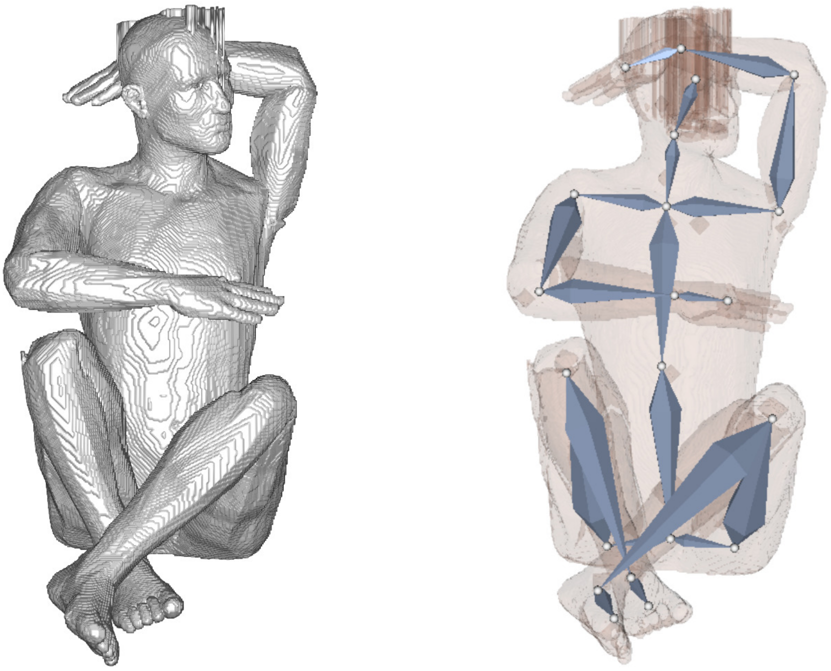


Figure 6.25: Front view of the sixth fetus pose. On the left the volume visualization of the prototype and on the right the visualization of the included armature.

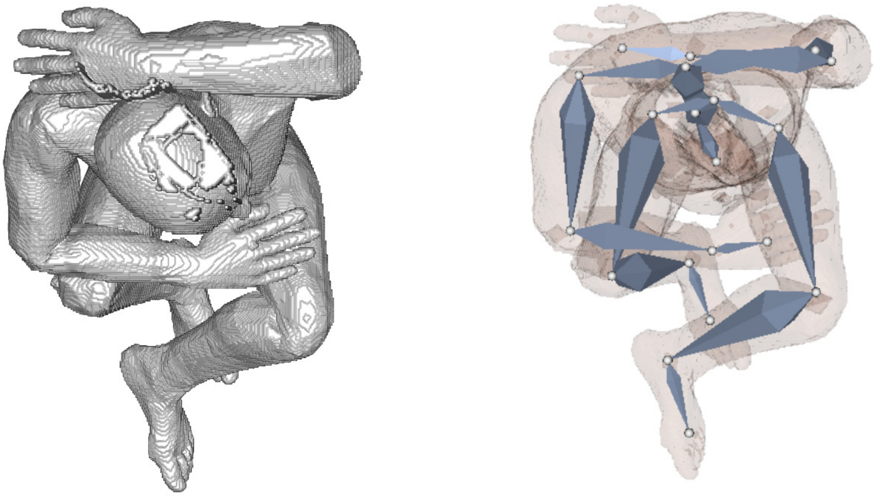


Figure 6.26: Top view of the same formation also volume and armature view.

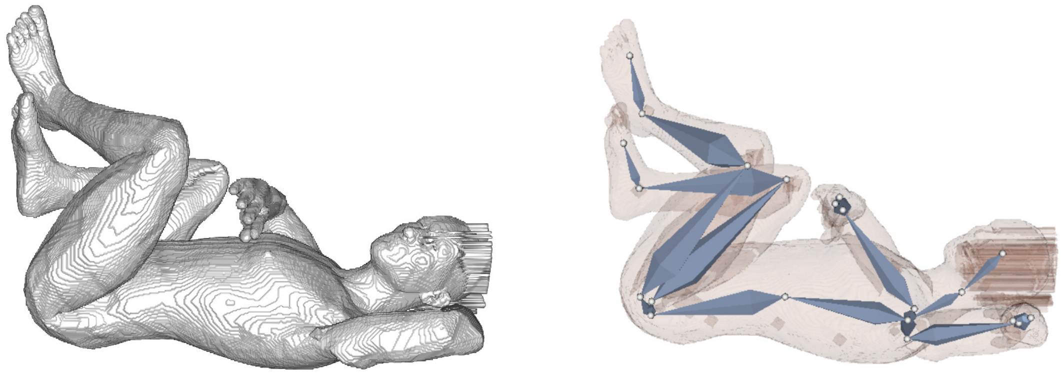


Figure 6.27: Side view of the pose also with visible armature and volume visualization.

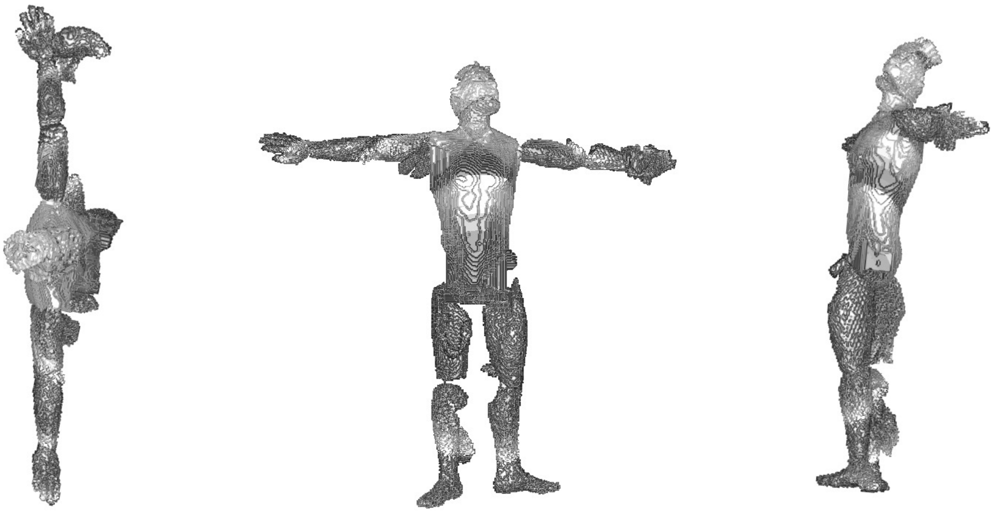


Figure 6.28: The result T-pose of the sixth fetus pose from left to right: top view, front view and side view.

6.1.7 Seventh fetus pose

The seventh pose is the most folded one. Both legs and arms are folded. The head is tilted too. Figure 6.29 shows the frontal view of the model presenting the crossed legs and the folded arms. The top view also reveals the tilted head and is shown in Figure 6.30. The side view is depicted in Figure 6.31 also shows the narrow angle between the torso and the legs.

Figure 6.32 presents the result of the transformation. In this example deformations of the torso occur, for example the femur takes a part of the torso. The left arm also takes some part of the torso. The overall T-pose is although also very well established in this result.

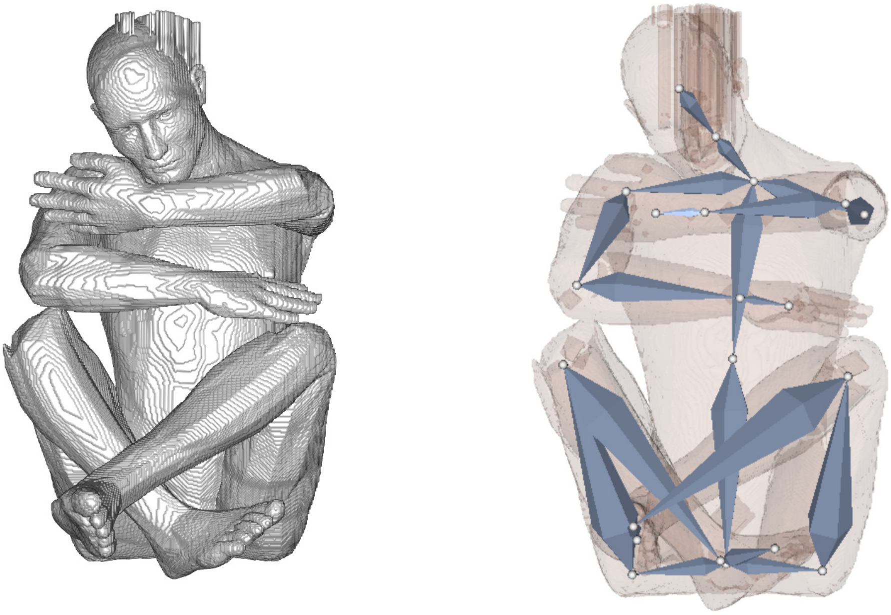


Figure 6.29: Front view of the seventh fetus pose. On the left the volume visualization of the prototype and on the right the visualization of the included armature.

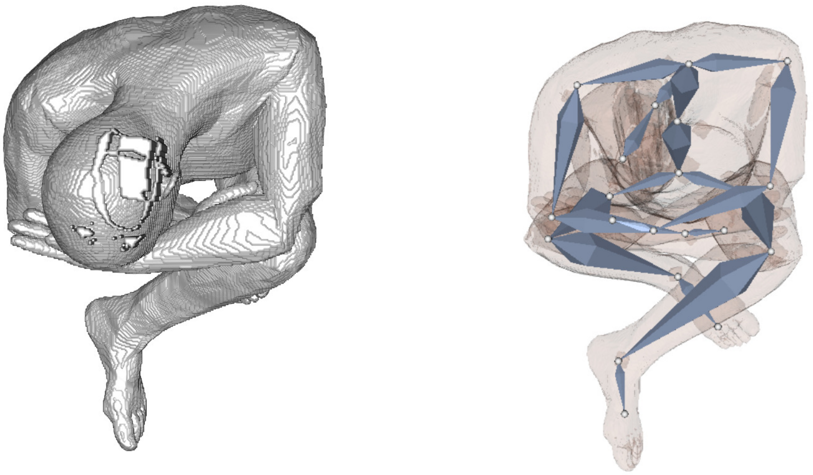


Figure 6.30: Top view of the same formation also volume and armature view.

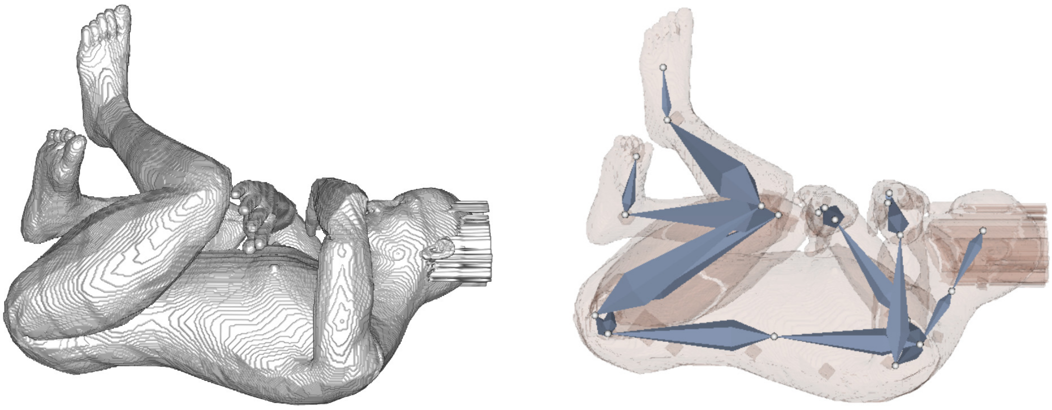


Figure 6.31: Side view of the pose also with visible armature and volume visualization.

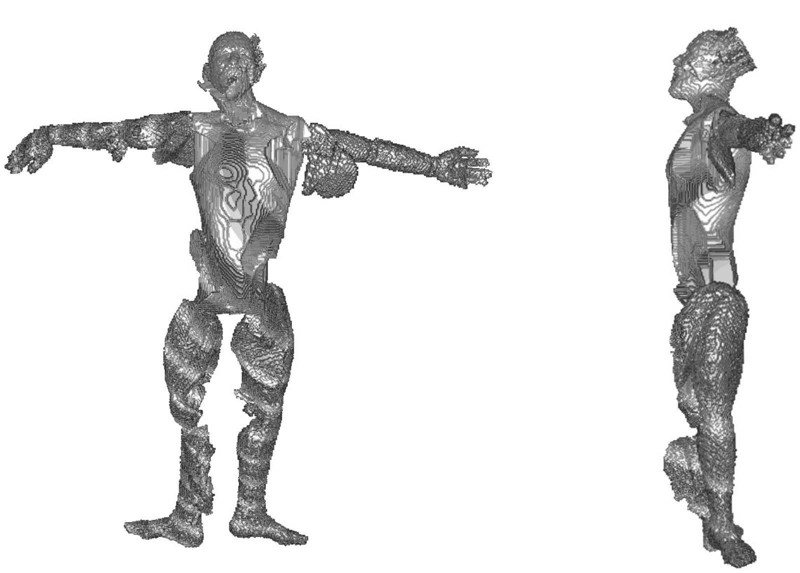


Figure 6.32: The result T-pose of the seventh fetus pose from left to right: top view, front view and side view.

Performance statistics

As stated in the begin of this chapter performance analysis is important when presenting a novel method or approach and therefore we tried to measure how good the algorithm performs. The performance is measured using different operations. First there is a time consuming part involved in the process of having the T-pose transformed data namely the rigging, described in Section 4.4. Therefore the time being spent in the program Bender [Ble19] has been measured from the step of loading the data until the result has been saved for further processing. The overview of the times is shown in Table 6.1. The average duration of the whole process being done in Bender is **seven** Minutes. The rigging is performed by the author and therefore maybe someone less familiar would perform slower.

Phantom	Starting time	End time	Duration
Man1	15:12	15:24	00:12
Man2	16:46	16:50	00:04
Man3	15:41	15:52	00:11
Man4	16:12	16:21	00:09
Man5	16:24	16:30	00:06
Man6	16:32	16:36	00:04
Man7	16:38	16:44	00:06

Table 6.1: Overview of the duration needed to rig the seven phantom models.

The second and maybe most important measurement of performance is how well the transformed data represent the original T-pose. Therefore the result has been registered with the T-pose input and the scalar volume difference has been produced. The result of one of those measurements is shown in Figure 6.33. It somehow looks like that there is lots of data left and therefore the overlapping is low, but it has to be considered that the data is volumetric and therefore the number of voxels remaining is important and not if the rest still looks like a human. The analysis results are presented in Table 6.2. The average of the overlap percentage is 79,02% which depends on how well the armature is placed in the data.

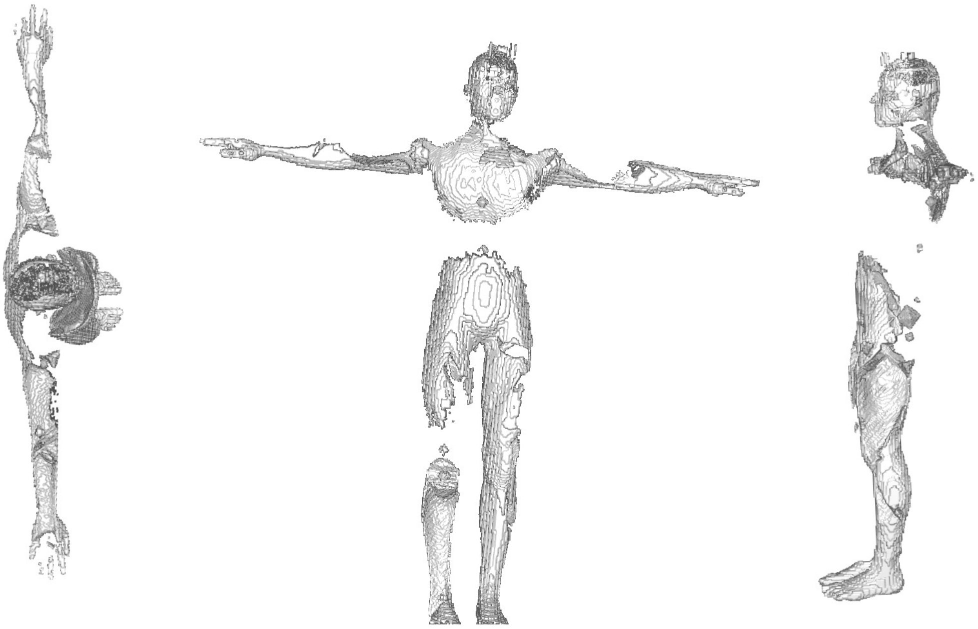


Figure 6.33: The result of the difference of the fourth fetus pose from left to right: top view, front view and side view.

Phantom	Total amount voxels T-pose	Amount of voxels left after substraction	Percentage Left	Percentage Overlap
Man1	2654642	790637	29,78%	70,22%
Man2	2654642	886471	33,39%	66,61%
Man3	2654642	572079	21,55%	78,45%
Man4	2654642	530825	20,00%	80,00%
Man5	2654642	492437	18,55%	81,45%
Man6	2654642	286991	10,81%	89,19%
Man7	2654642	338764	12,76%	87,24%

Table 6.2: The table shows the total amount of voxels in the T-pose and the amount of voxels left after subtraction of the result of the method and the T-pose. The left two columns represent the percentage values of voxels left afer subtraction and the overlap percentage which represents the performance of the algorithm.

We think that the measurement similarity of the head to toe and the finger to finger span is one of the most important qualitatively result of our approach. Therefore we measured those distances in each of the man phantoms and present the results in Table 6.3. The average of the head to toe measurement similarity is **94,05%** and the finger to finger span measurement similarity is **91,80%**.

Phantom	Head to Toe Measurement Overlapping	Finger to Finger Measurement Overlapping
Man1	95,59%	92,46%
Man2	90,94%	85,09%
Man3	99,17%	96,05%
Man4	92,10%	88,31%
Man5	94,93%	97,49%
Man6	90,64%	89,11%
Man7	95,00%	94,07%

Table 6.3: The table shows the measurement similarity of the head to toe and the finger to finger span between the results of our approach and the original T-pose.

6.1.8 Fetus data

The fetus data presented in this section is also gathered from the internet. One dataset is a simple 3D model of a fetus [Spe] and the other one a simulation of a whole fetus ultrasound scan produced by Cortes et al. [CKM+16]. This section is structured just as the one with the man phantoms. The difference between the datasets is that in this case there is no ground truth that can be compared so there is no T-pose which is the right solution. Therefore the only performance analysis can be done in terms of time needed for rigging the data.

Figure 6.34 shows the front of the fetus model. It is clear to see that the model is different to the man models used before, the limbs are really small and the head is bigger. Figure 6.35 shows the top of the model and it is clear to see that the head is very big. The side view of the model is shown in 6.36.

The result of the method is shown in Figure 6.37. The T-pose is very clear to see but there are some considerable deformations involved. The arms are ripping out parts of the torso and also the legs do so. The measurements from head to toe and from finger to finger could be analysed very well but the circumferences of the abdomen and head may not be very precise.

The following data shows the phantom model which represents how the output of a full fetus ultrasound scan may look like. The data has been pre-filtered in order to make is usable for the method presented. The original data is shown in Figure 6.38.

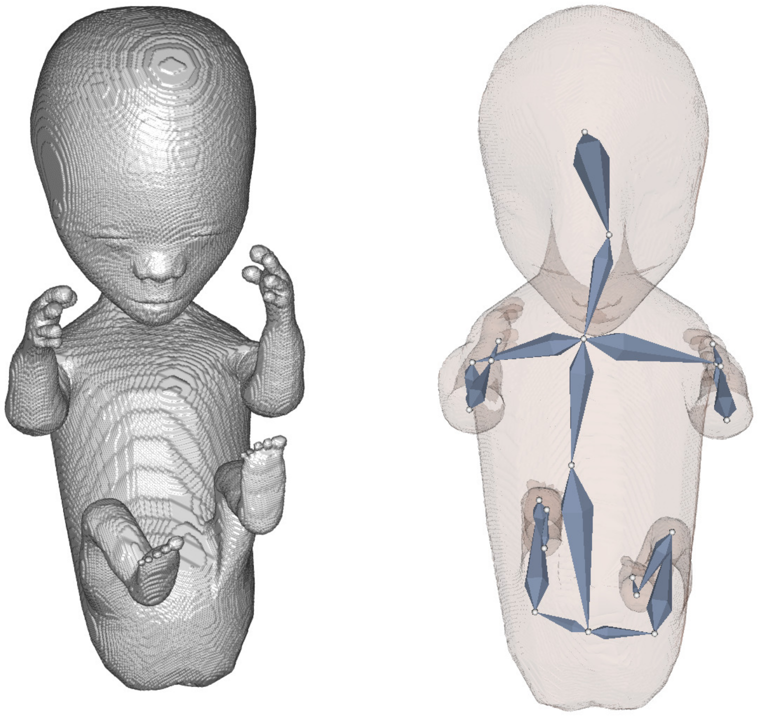


Figure 6.34: Front view of the fetus model [Spe]. On the left the volume visualization of the prototype and on the right the visualization of the included armature.

The result of the preprocessing step and the result of the fetus formation is shown here. Figure 6.39 represents the fetus from the front and the incorporated armature. The feet are missing in the model therefore parts of the limbs are used as the feet information. The top view of the model is shown in Figure 6.40 and the side view is shown in Figure 6.41.

The result of the method is shown in Figure 6.42. The T-pose is also very clear to see. Measurements like the head to toe distance and the finger to finger span should be quite precise but the deformations of the torso may make the circumference measurements inapplicable. Overall the result looks quite good and the T-pose is clear to see.

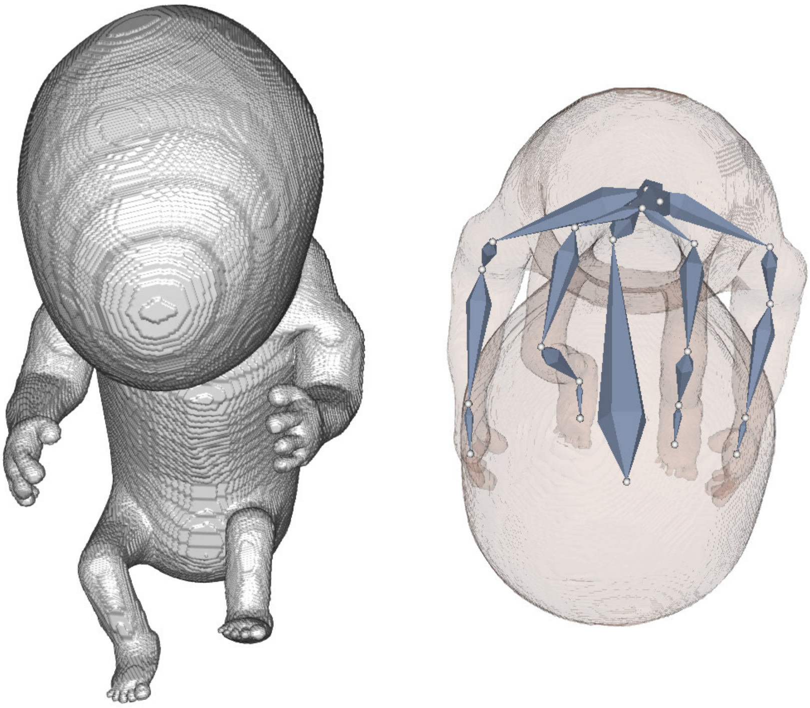


Figure 6.35: Top view of it also volume and armature view.

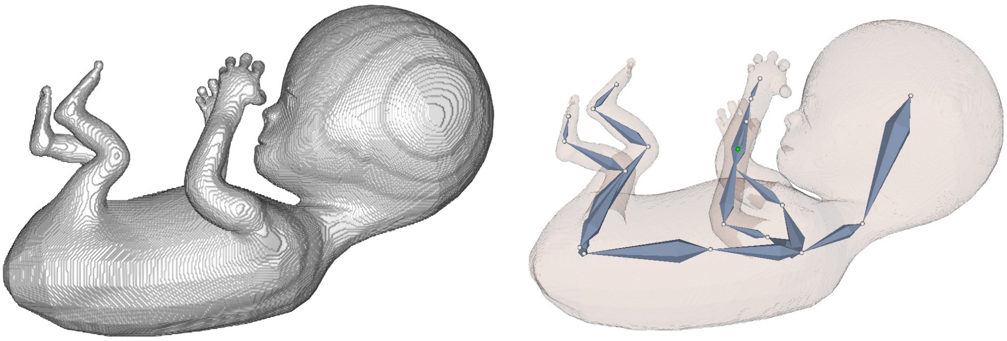


Figure 6.36: Side view of the pose also with visible armature and volume visualization.

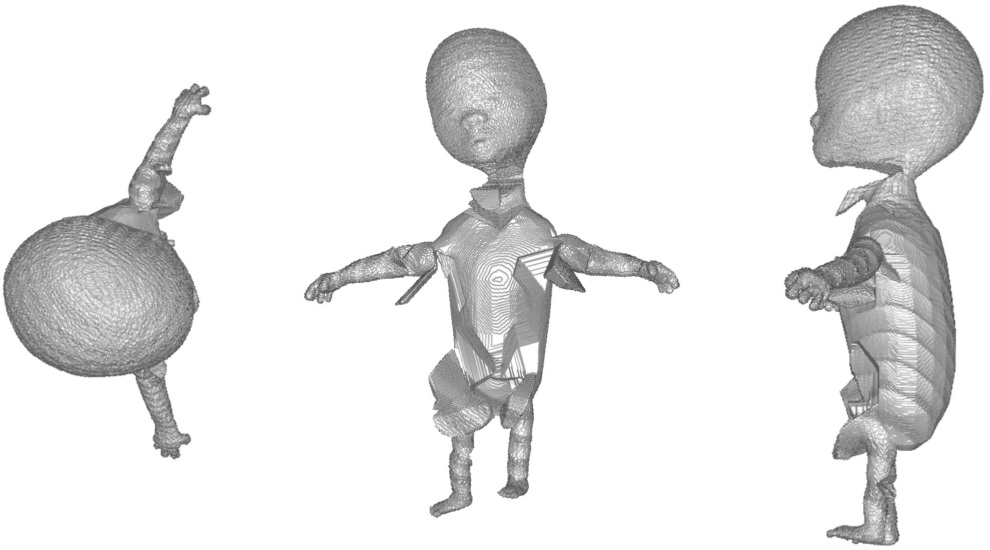


Figure 6.37: The result T-pose of the fetus model from left to right: top view, front view and side view.

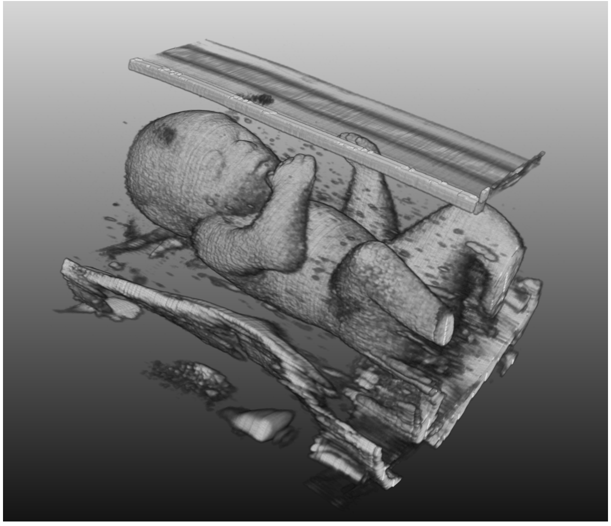


Figure 6.38: Original phantom model created by Cortes [CKM+16] which represents a phantom of a full fetus 3D ultrasound scan.

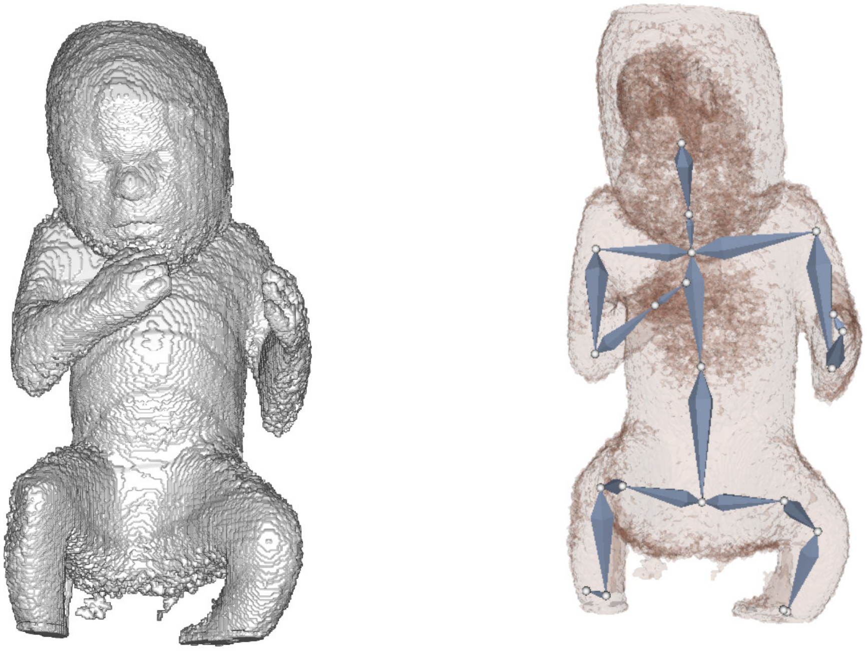


Figure 6.39: Front view of the fetus phantom provided by Cortes [CKM+16]. On the left the volume visualization of the prototype and on the right the visualization of the included armature.

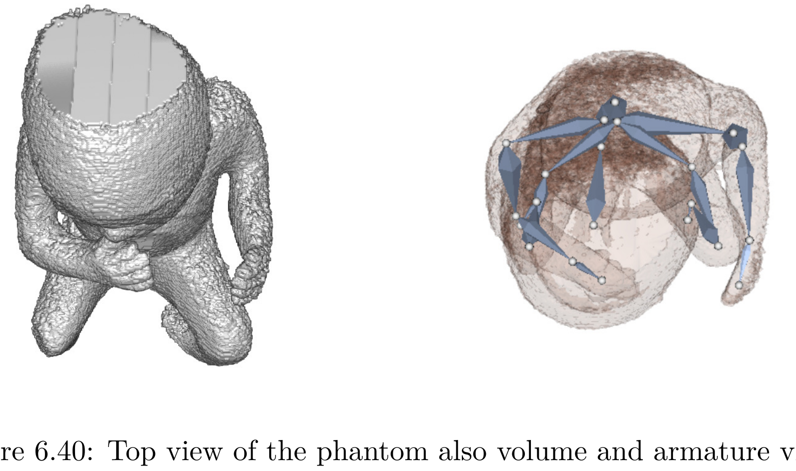


Figure 6.40: Top view of the phantom also volume and armature view.

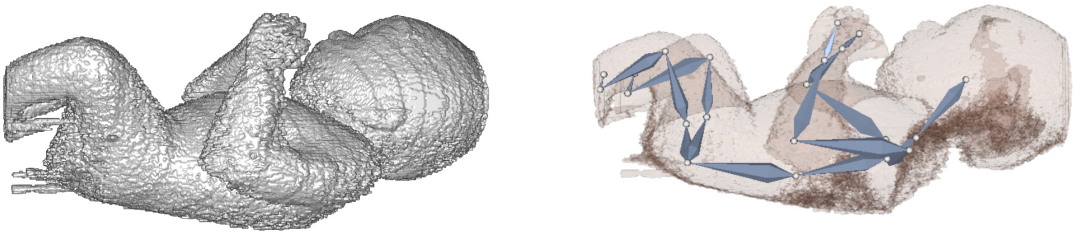


Figure 6.41: Side view of the pose also with visible armature and volume visualization.

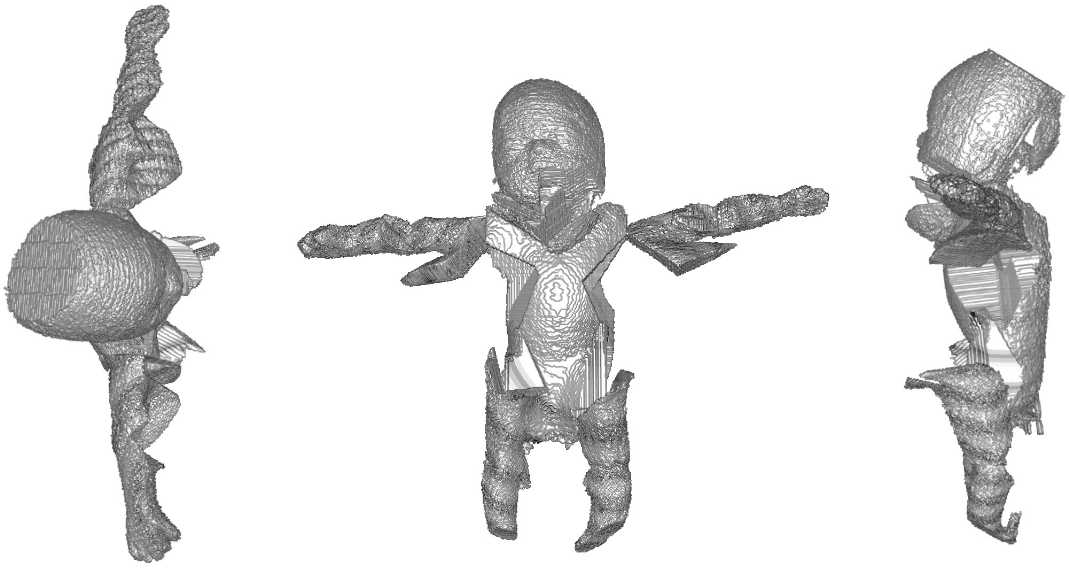


Figure 6.42: The result T-pose of the fetus phantom from left to right: top view, front view and side view.

The analysis of the method in terms of fetus like or phantom fetus models is represented by the time the end user needs in order to perform the manual step of rigging. Table 6.4 represents the time needed to perform this manual step. The average time to do so is **eight** minutes.

	Starting time	End time	Duration
Fetus Model	18:50	18:56	00:06
Fetus US Phantom	19:00	19:11	00:11

Table 6.4: Table representing the time needed for rigging the fetus model and the fetus ultrasound phantom.

6.2 Influence analysis

This section of the results handles findings which somehow show where the limits of the method are. There are some influences which lead to major artifacts and unwanted results. Those limiting cases are presented here.

6.2.1 Noise

The first problem which has been identified is the influence of noise in the data. In case of the phantom ultrasound investigation result created by Cortes et al. [CKM$^+$16] noise which is usual for an ultrasound investigation has been added. If this data is directly loaded into the Bender application the result looks like shown in Figure 6.43 and Figure 6.44. This data cannot be used in order to rig the data and produce a meaningful output. Therefore the preprocessing step has been established. The surrounding of the data has to be free of any noise. In fact the data representing the fetus itself may have some noise included which can be seen in the results of the fetus phantom. The noise can be removed by the introduced preprocessing steps which can be seen in section 4.3. The here shown examples makes it clear that the preprocessing steps are very important in order to make the method work.

6.2.2 Weighting issues

In some cases especially when the pose of the fetus is really packed and the limbs are very close to the body the weighting of the data does not perform very well. The problem is that if the arms are connected with the body the armature in the arms will compete with the armature in the body, namely the torso and the belly. The voxels in between those armature parts will be divided and the armature in the upper part will get a piece of the body which should not happen. Another example is given when the tighs are very close to he body they will somehow share some parts of the belly or of the pelvis. This phenomena is also likely to occur if the arms touch the legs. One example of this weighting issue is presented in Figure 6.45. As discussed in the section 4.5 one solution to this problem might be to use a different method to calculate the weighting. The result

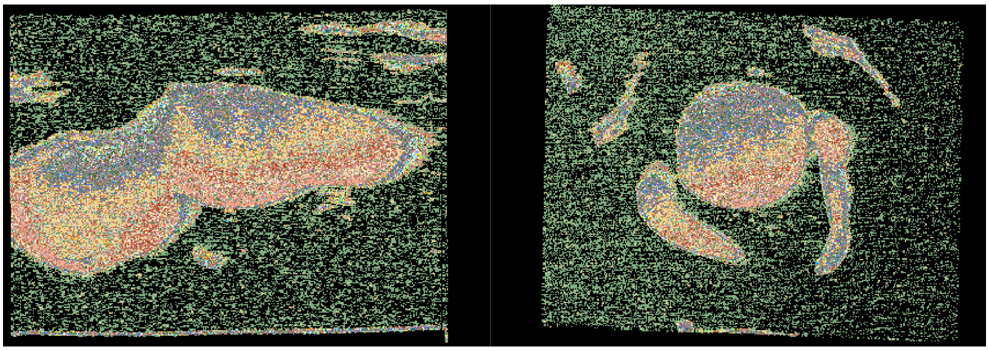


Figure 6.43: Different views of the noise in the phantom ultrasound data provided by Cortes et al. [CKM+16] in Bender [FOA+14].

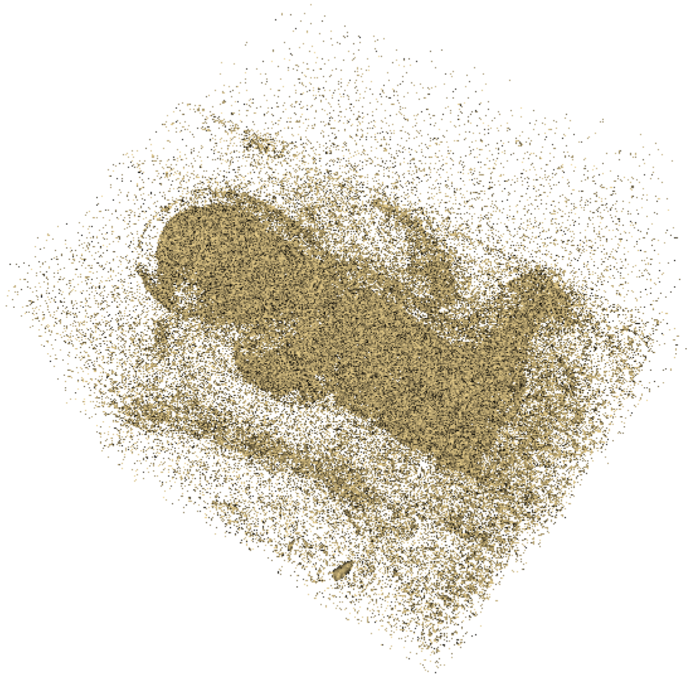


Figure 6.44: 3D visualisation of the fetus phantom with the noise included in Bender [FOA+14].

should be improved by introducing assumptions about the shape of the segments like the arms are somehow cylindrical. This would reduce the problem of the arms and also the legs taking parts of the torso with them.

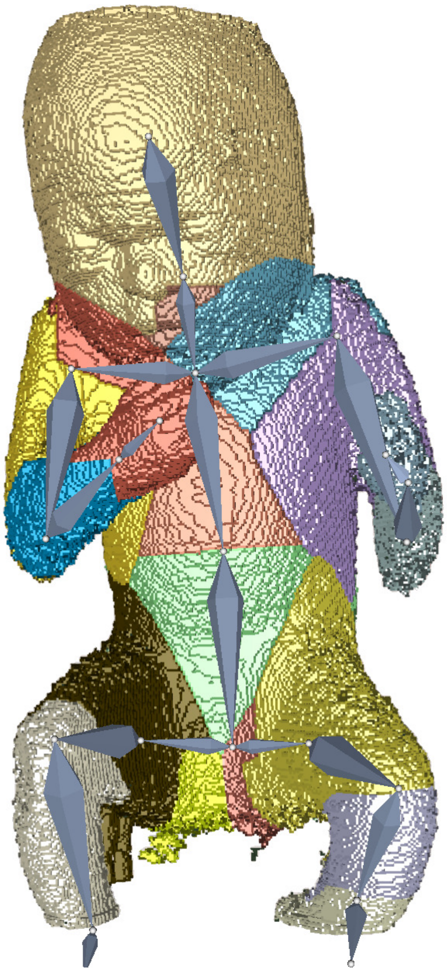


Figure 6.45: Visualization of the weighting of the fetus phantom model. The arms and the legs have too much voxels assigned to them.

6.2.3 Limited space for rotation

Another issue which has been faced during the evaluation phase was that the size of the volume to be worked with is somehow limited. The male phantom models have a resolution of 402x402 and around 250 in the depth which depends on the pose of the model. In order to perform all the rotations without limbs getting cut off by the volume borders first the volume is expanded by a factor of two which results in a volume size of around 800 times 800 and 500 in the depth. This is somehow also the limitation for the *threshold* function in 3D Slicer. In some cases like the examples five, six and seven the expanding of the volume to the doubled size has not been enough. The initial result may be seen in Figure 6.46. If such datasets have to be analysed they have to be scaled down before they can be used. The scaling is done in case of example five, six and seven and results in a lower resolution of the output.

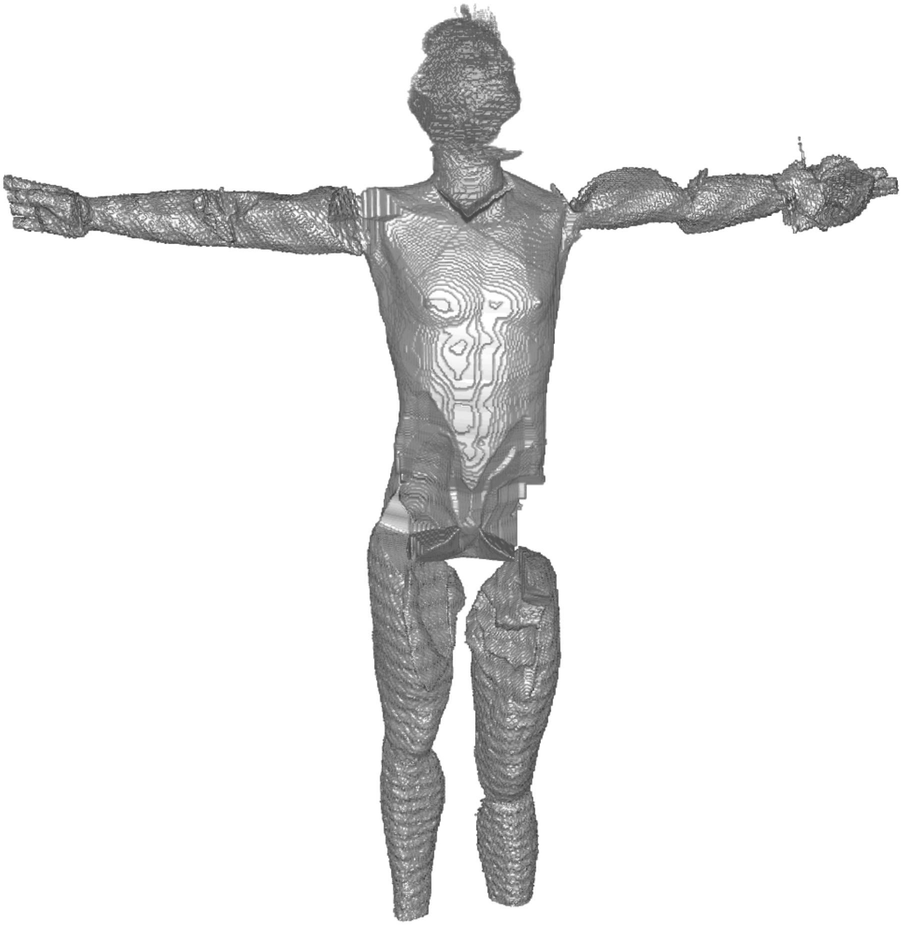


Figure 6.46: The limited space for the rotation of the limbs leads to a cut off of the legs in case of the phantom man five

6.3 Limitation and strengths

The results presented in this chapter clearly show the strengths and the limitations of our new and novel approach. The main goal of the method is to present the fetus in a T-pose. This goal has been reached in every case we applied our method. We see the main strength in our approach that it is reliable and reproducible. The measurement of finger to finger span and the head to toe distance has a very high precision. Regarding the limitations we have to state that the deformations of the torso are limiting the application of the approach. The measurement of the abdominal circumference would be highly affected by these deformations. Noise in the data as well as structures in the dataset, that do not belong to the fetus are also a problem and have to be removed by the preprocessing step. We think the resulting T-pose is clear to see in each example we tried and we think that the comparability and the standardization of the pose helps to analyze differences between fetuses and to control the development of the fetus during the pregnancy.

CHAPTER 7

Conclusion

In this chapter the work presented will be summarized and discussed. Limitations which have been identified will be verbalized as well as potential applications in a clinical context. At the end future work which may be performed will be stated.

7.1 Summary

The work in this thesis can be summarized as a novel approach to transform fetal data of an ultrasound investigation to a standardized position. Standardization always opens the possibility of comparison not only inter-case but also intra-case [MMNG15, OKG^{+}06, NUX02] for example to compare scans of different development stages of the fetus. The result of the transformation can be analyzed automatically in terms of size of the fetus. The span from the left fingertip to the right fingertip can be computed automatically as well as the size of the fetus from the head to the toe. Getting rid of occlusions caused by a very bent or curved pose of the fetus also enables gynecologists to take measurements which have not been measured before in terms of prenatal analysis.

The first step for applying this method is to analyze the data which is considered as the input. The data has to be cleared of all disturbances like noise in the data or components which are not connected to the fetus. The fetus has to be available in the data as a whole. In order to perform these tasks a threshold operation as well as a connected component analysis is applied. In terms of creating a whole volume dataset the compound volume generation introduced by Viola et al. may be applied [VBS^{+}13]. After clearing the data and making it ready for transforming the fetus the data has to be rigged. Rigging is the process of placing a skeleton or an abstraction of the fetal skeleton in the data in order to create a hierarchy.

The data has to be weighted or skinned according to the armature which has been placed in the data beforehand [DdL13]. Skinning means that the voxels are assigned to the nearest armature segment in order to be able to transform the data. The weighting is performed by creating a heat map between the segments and assigning the voxels to the nearest segment of the armature.

The next step is the actual transformation which is the most novel component of this thesis. The transformation itself is performed completely automatically. It only takes the armature and the skinned volume as an input and automatically creates the transformed volume in the T-pose. The transformation is done by analysing the hierarchy of the armature. One may think of how the parts of the skeleton are aligned when the human body is in a T-pose. For example the forearm is perfectly aligned with the upper arm meaning that the segment that represents the forearm has to face in the exact same direction as the upper arm. This system also applies for the legs and for the head as well as the neck. Considering this fact the transformation has to calculate the angle which has to be used for the transformation and the center of rotation in order to rotate each segment in the same direction as the adjacent one. Starting from the most far away point namely the hands and the feet back to the torso the transformation results in a perfect T-pose for each case. The quality of the result of course depends on the given data as well as the positioning of the armature in the data.

The last step which can be performed semi-automatically is the analysis of the data. There are different options to perform this task and it also depends on which measurements the medical personal is interested in. The goal is that the analysis is much easier if the fetus is unfolded and not in the natural folded position. Having the data in a standardized position may result in being able to compare different fetuses and check if the growth is normal and healthy. Being able to also compare the growth of one fetus over time in an easier way is a part of the goal.

We assessed the performance of our solution by using a phantom model of a man [The10]. The quality of the result has been assessed by analyzing the percentage of voxels which are overlapping with the goal T-pose and the average was 79,02%. Even better are the results of analyzing the similarity of the measurements head to toe and finger to finger span between the result of our approach the and the goal T-pose. The average finger to finger span measurement similarity is 91,08% and the similarity of the head to toe measurement is even higher, 94,05%.

Another important aspect of the method is how much time it takes to perform the manual rigging task. We assessed the time needed for this task while rigging the seven man phantom models. It took us in average seven minutes to perform the task. We also applied our method to more fetus like data namely a fetus model and a phantom model of a 3D ultrasound investigation. The results also look very promising.

7.2 Research questions and requirements

As stated in Chapter 1 the main research question is, if we can provide a transformation of the fetus in 3D to a standardized position, to analyze the fetal development. We can definitely answer this question with yes. We provide a method which partly guides the user through the process of transforming 3D ultrasound data of a fetus to the end result, namely the data presented in a T-pose. The usage of different tools for different steps of the workflow might be inconvenient for the users but we already think of incorporating all features into our 3D Slicer [PHK04] extension. Our main contribution with the presented work is the transformation of the data. The process is fully automatized and delivers fast and precise results. The second most important contribution of this thesis is the development of the strategy and the workflow to perform the transformation.

In this section we would also like to reflect the requirements we discussed in section 1.5. We have shown that the method performs with a minimal amount of user interaction. The user needs some special domain knowledge, namely some anatomical knowledge of the joints of a fetus and how to maneuver in a 3D data viewer. The comparability of the method depends highly on the rigging the user performs but the T-pose of the result is always very similar. Currently the analysis has to be carried out manually but it could be automatized. The result of the method includes some deformations dependent on the given data and the rigging process, but the measurements of finger to finger span and head to toe length is only minimally affected by them. The system could be installed in a clinical environment but the system would need a proper graphical processing unit and central processing unit.

We have shown that it is possible to carry out the transformation itself automatically while the rigging and the preprocessing as well as the analysis has to be done manually with the current solution. The main part namely the transformation of the data had to be implemented while the preprocessing, the rigging and the weighting have been reused. Currently the volume resolution determines the time needed for the transformation. The transformation of a 400 × 400 × 255 voxel volume would need around four minutes on a machine with a 8700k central processing unit and a NVIDIA Titan X graphical processing unit.

7.3 Discussion and Limitations

This work is a prototype of a novel approach and has to be tested in an medical environment to find out if it is useful and applicable. User studies could be performed in a clinical environment to assess if the application is self explanatory and if the clinical personnel is able to perform the needed manual tasks involved. It has to be clear that including a manual step which needs domain information about the anatomy of the fetus and also knowledge about navigation in 3D space may lead to a negative reaction of the medical staff because they are normally used to a 2D view of slices. In general the more automation included in the transformation process the better, because it improves for example the reproducibility and the acceptance of the gynecologists because they would not have to spend too much time interacting with the system.. Therefore the fully automated transformation of the data is a step into the right direction. Another important aspect is the time needed to perform different tasks. It has to be analysed if the gynaecologist is willing to spend around ten minutes for a manual task per fetus examined. The time needed to perform a task may be a key fact which determines if the method is being used or not. The performance of around 79% is clearly a positive sign to work further on the method and try to deploy it in a clinical environment.

On the other hand it has to be said that there are also some limitations when using this method. The data has to be cleared from all disturbances. This might be one of the most important limitations because ultrasound often tends to be very noisy and the fetus can often not be discriminated very clearly. Another limitation is the positioning of the armature in the data. Including a manual task always has a great effect on the outcome. If the user is not well trained or underestimates this task the result will not be usable at all. The weighting currently used is also a limitation which has to be taken into account. If any parts of the body are connected the voxels between segments of the armature will be shared in a way which might not be beneficial. One limitation which has to be focused when working with very large volumetric data is that the time needed to transform the data scales not linearly with the size of the volume.

Considering the fact that we apply a forward transformation to the data we should definitely consider that an inverse transformation as shown by Correa et al. [CSC10] would maybe lead to a better result. The artifacts that are present in our work may not only be present because of the prior discussed weighting issues, they can also originate from the fact that we use a forward transformation. We definitely think of adding this type of transformation in the future and compare the results when using both transformation types.

7.4 Applications in a clinical context

The main goal of this thesis was to analyse and implement a way to interactively transform 3D ultrasound data of a fetus investigation into a T-pose. Clearly one clinical application where this method may be used is in case of prenatal investigations. The method may not only be used in normal prenatal checks but could also be interesting for medical scientific research. It would be groundbreaking if the analysis of fetal data in a standardized position would lead to an insight of genetic disorder without having to perform a mother fluid analysis. Presenting the data of an ultrasound investigation in a transformed way may also be helpful for maybe non clinical persons like the parents. They would be able to perceive their fetus in a way where there are no occlusions and every part of it is visible. In case of abnormal growth the communication using the imaging data can also be improved.

The application of a transformation into a standardized position is a general concept and may not only be applicable in case of a fetus transformation. It can also be useful in other application where there are occlusions or where it would be interesting to standardize the data. One example might be to unroll or unfold the intestinal tract to be able to measure the length or see if there are any abnormalities. The approach might also be helpful to transform organs into a standardized position to be able to measure them or compare them with each other. The application might be very diverse and has to be further researched.

One aspect during the search for related work was that sometimes it can also be useful to transform the data recorded from animals to a standardized position. If for example a rat is screened it may not always hold still or is positioned completely straight. It might roll in some way and in that case a transformation to a straight position could be helpful for further analysis and comparison of the data.

7.5 Future work

One of the most important features which shall be added for a more convenient and reliable solution would be to automatically incorporate the armature in the data. The process of rigging can be very tedious and annoying and would be a very interesting part to be automatized. It also has to be taken into account that the result of the automatic rigging might not be sufficient enough to work with it. Therefore a semi-automatic approach might be the best solution. A domain expert should get the opportunity to have a look at the rigging result and if needed he or she shall be able to adapt the positioning of the segments and joints in the data. One option to solve this problem might be to having a look at the topology toolkit by Tierny et al. [TFM^{+}17]. It might be used to analyse the topological structure of the input data and present a way to analyze the data and create a skeleton used as basis for the transformation.

Currently the transformation of the data is done segment by segment in a sequential way. During the evaluation and the write up it was clear that this task may be subject of an optimization by performing the task in parallel. The transformation could be done for both arms and legs as well as the head and the neck at the same time because they are not dependent. This would lead, in an optimal situation, to an algorithm needing only about a fourth of the time than the original one.

The tasks performed in this implementation of the method are currently mainly performed on the central processing unit of the machine. One improvement especially when working with volumetric data and medical data is to perform the calculations and transformations using the graphics processing unit. This unit is optimized by performing tasks in parallel and therefore the calculation would be much faster.

Another very important and helpful adaption would be to re-implement the weighting or skinning of the data. The currently used implementation has some downsides leading to major artefacts in the output like in the case of the fetus phantom by Cortes et al. [CKM+16]. If the data is very close to each other and the limbs touch the torso or each other, the weighting algorithm does not know the border between them and divides the voxels evenly between the segments, leading to the artefacts. Other approaches might take regular structures into account for example the limbs are somehow cylindrical. The benefit of that approach would be that it gets rid of the problems currently arising due to limbs touching the body of the fetus.

One feature which could be added is to automatically analyse the output of the algorithm. If standardized measurements could be included in the data off the shelf the clinical personal would not have to take them every time by hand. Also new and innovative measurements like a real measurement of the volume of the head or of the torso might be interesting. Automatically performed growth analysis over time or between different fetuses are also very interesting. An automatically performed weight estimation based on the volume of the fetus could also be a very interesting information for preparing the birth of the baby.

The last point we would like to discuss is the development of an interface where the user is able to easily assess the development of the fetus over the whole time of the pregnancy. We think that it would be rather easy to use the resulting data we provide with our method to develop a program which presents the development. The user should be able to see the data of each week of the pregnancy where a 3D ultrasound screening has been performed and the tool might display an analysis in terms of growth. The tool should help clinicians as well as the parents to control and see the development of the future child.

Acronyms

2D two dimensional. xiii, 1–3, 13, 14, 16, 34, 53, 59, 112

3D three dimensional. xiii, xiv, 1, 2, 4, 8, 9, 12–16, 18, 22–30, 44–47, 51, 54, 58, 61, 63, 68, 72, 98, 101, 105, 110–112, 114

4D four dimensional. 1, 13, 23, 24

CLI command-line interface. 68, 70, 72

CRL crown-rump length. 3, 16

CT computed tomography. 36

DICOM Digital Imaging and Communications in Medicine. 8, 39

GPU graphics processing unit. 25, 26, 42, 50

HC head circumference. 3, 16

IK inverse-kinematics. 21, 26, 27, 35, 36

ITK Insight Segmentation and Registration Toolkit. 8, 54, 58, 70, 72

LBS linear blend skinning. 31, 32

MRI magnetic resonance imaging. 3, 4, 14, 36

VOCAL Virtual Organ Computer-aided Analysis. 14

VTK visualization toolkit. 54, 72

Bibliography

[ADCB^{+}15] M. Alessandrini, M. De Craene, O. Bernard, S. Giffard-Roisin, P. Allain, I. Waechter-Stehle, J. Weese, E. Saloux, H. Delingette, M. Sermesant, and J. D'hooge. A pipeline for the generation of realistic 3d synthetic echocardiographic sequences: Methodology and open-access database. *IEEE Transactions on Medical Imaging*, 34(7):1436–1451, 2015.

[And14] Johan Andruejol. Bender modules. https://public.kitware.com/Wiki/Bender/Documentation/2.0/Modules/, 2014. Accessed: 2019-03-09.

[Bas11] Ahmet A. Baschat. Examination of the fetal cardiovascular system. *Seminars in Fetal and Neonatal Medicine*, 16(1):2–12, 2011.

[BCPC^{+}15] E. Bonet-Carne, M. Palacio, T. Cobo, A. Perez-Moreno, M. Lopez, J. P. Piraquive, J. C. Ramirez, F. Botet, F. Marques, and E. Gratacos. Quantitative ultrasound texture analysis of fetal lungs to predict neonatal respiratory morbidity. *Ultrasound in Obstetrics and Gynecology*, 45(4):427–433, 2015.

[Bla15] Bruce Blaus. Fetal ultrasound. `https://commons.wikimedia.org/wiki/File:Fetal_Ultrasound.png`, 2015. Licence: `https://creativecommons.org/licenses/by-sa/4.0/deed.en`.

[Ble19] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2019.

[BP07] Ilya Baran and Jovan Popović. Kre. *ACM Transactions on Graphics*, 26(99):72, 2007.

[BTST12] Gaurav Bharaj, Thorsten Thormählen, Hans Peter Seidel, and Christian Theobalt. Automatically rigging multi-component characters. *Computer Graphics Forum*, 31(2):755–764, 2012.

[BUB10] Ulaş Bağci, Jayaram K. Udupa, and Li Bai. The role of intensity standardization in medical image registration. *Pattern Recognition Letters*, 31(4):315–323, 2010.

[CKM$^+$16] Camilo Cortes, Luis Kabongo, Ivan Macia, Oscar E. Ruiz, and Julian Florez. Ultrasound image dataset for image analysis algorithms evaluation. In *Innovation in Medicine and Healthcare*, pages 447–457. Springer, 2016.

[CSC10] Carlos D. Correa, Deborah Silver, and Min Chen. Constrained illustrative volume deformation. *Computers and Graphics (Pergamon)*, 34(4):370–377, 2010.

[CSYB05] Nicu D. Cornea, Deborah Silver, Xiaosong Yuan, and Raman Balasubramanian. Computing hierarchical curve-skeletons of 3d objects. *Visual Computer*, 21(11):945–955, 2005.

[DdL13] Olivier Dionne and Martin de Lasa. Geodesic voxel binding for production character meshes. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '13, pages 173–180, New York, NY, USA, 2013. ACM.

[DS05] C.Correa W. Hurt P. Mason J.Ziria D. Silver, K.yaws. Volumetric manipulation for dosimetry simulations, 2005.

[DZL13] Su Zhen Dong, Ming Zhu, and Fen Li. Preliminary experience with cardiovascular magnetic resonance in evaluation of fetal cardiovascular anomalies. *Journal of Cardiovascular Magnetic Resonance*, 15(1):1–12, 2013.

[Eco18] ImageJ Ecosystem. Imagej. `https://imagej.net/Welcome`, 2018. Ecosystem, ImageJ. Accessed: 2019-03-10.

[FBKC$^+$12] Andriy Fedorov, Reinhard Beichel, Jayashree Kalpathy-Cramer, Julien Finet, Jean-Christophe Fillion-Robin, Sonia Pujol, Christian Bauer, Dominique Jennings, Fiona Fennessy, Milan Sonka, John Buatti, Stephen Aylward, James V Miller, Steve Pieper, and Ron Kikinis. 3d slicer as an image computing platform for the quantitative imaging network. *Magnetic Resonance Imaging*, 30(9):1323–1341, 11 2012.

[FH89] J. H. Ferguson and F. W. N. Haultain. Handbook of obstetric nursing. In *Handbook of Obstetric Nursing*, pages 150–153. Young J. Pentland in Edinburgh, 1889.

[FOA$^+$14] Julien Finet, Ricardo Ortiz, Johan Andruejol, Andinet Enquobahrie, Julien Jomier, Jason Payne, and Stephen Aylward. Bender: An open source software for efficient model posing and morphing. In *Biomedical Simulation*, pages 203–210. Springer, 2014.

[GS99] Nikhil Gagvani and Deborah Silver. Parameter-controlled volume thinning. *CVGIP: Graph. Models Image Process.*, 61(3):149–164, May 1999.

[HDHP84] F P Hadlock, R L Deter, R B Harrist, and S K Park. Estimating fetal age: computer-assisted analysis of multiple fetal growth parameters. *Radiology*, 152(2):497–501, 8 1984.

[Hoe16] Rama Karl Hoetzlein. Gvdb: Raytracing sparse voxel database structures on the gpu. *High Performance Graphics*, 2016.

[HZ17] Qinghua Huang and Zhaozheng Zeng. A review on real-time 3d ultrasound imaging technology. *BioMed Research International*, 2017, 2017.

[IDTS06] Nikolas M. Ivancevich, Jeremv J. Dahl, Gregg E. Trahey, and Stephen W. Smith. Phase-aberration correction with a 3-d ultrasound scanner: Feasibility study. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 53(8):1432–1439, 2006.

[JK17] Xun Jin and Jongweon Kim. A 3d skeletonization algorithm for 3d mesh models using a partial parallel 3d thinning algorithm and 3d skeleton correcting algorithm. *Applied Sciences*, 7(2):139, 2 2017.

[JKT13] Andrei C. Jalba, Jacek Kustra, and Alexandru C. Telea. Surface and curve skeletonization of large 3d models on the gpu. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1495–1508, 2013.

[KCE+12] D.F Keefe, D. Coffey, A. Ellingson, L. Thorson, F. Korsakov, M. Ewert, H. Hagh-Shenas, and D. Nuckley. Visualizing motion data in virtual reality: Understanding the roles of animation, interaction, and static presentation. *Computer Graphics Forum*, 31(3pt3):1215–1224, 2012.

[KMM+18] J. Kreiser, M. Meuschke, G. Mistelbauer, B. Preim, and T. Ropinski. A survey of flattening-based medical visualization techniques. *Computer Graphics Forum*, 37(3):597–624, 2018.

[KQ03] C. Kirbas and F. K. H. Quek. Vessel extraction techniques and algorithms: a survey. In *Third IEEE Symposium on Bioinformatics and Bioengineering, 2003. Proceedings.*, pages 238–245, March 2003.

[KST+14] Jan Kretschmer, Grzegorz Soza, Christian Tietjen, Michael Suehling, Bernhard Preim, and Marc Stamminger. Adr-anatomy-driven reformation. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2496–2505, 2014.

[KvdMF+17] Pieter Kruizinga, Pim van der Meulen, Andrejs Fedjajevs, Frits Mastik, Geert Springeling, Nico de Jong, Johannes G Bosch, and Geert Leus. Compressive 3d ultrasound imaging using a single sensor. *Science Advances*, 3(12):e1701423, 12 2017.

[LCEC09] Pam Loughna, Lyn Chitty, Tony Evans, and Trish Chudleigh. Fetal size and dating: Charts recommended for clinical obstetric practice. *Ultrasound*, 17(3):160–166, 2009.

[LCIB13] Bradley C. Lowekamp, David T. Chen, Luis Ibáñez, and Daniel Blezek. The design of simpleitk. *Frontiers in Neuroinformatics*, 7(December):1–14, 2013.

[LDB95] Andreas Lee, Joshp Deutinger, and Gerhard Bernaschek. Three dimensional ultrasound: abnormalities of the fetal face in surface and volume rendering mode. *BJOG: An International Journal of Obstetrics & Gynaecology*, 102(4):302–306, 1995.

[MA19] Medical Imaging (MITA) and Technology Alliance. Dicom, 2019.

[McD06] John McDonald. Rotation About an Arbitrary Axis. In , editor, *CGEMS - Computer Graphics Educational Materials*. The Eurographics Association, 18-7-2006.

[MeV] MeVis Medical Solutions. MeVisLab. `https://www.mevislab.de/`. AG, MeVis Medical Solutions. Accessed: 2019-04-12.

[MHP$^+$14] Markus Müller, Linn Emilie Sẽvil Helljesen, Raphael Prevost, Ivan Viola, Kim Nylund, Odd Helge Gilja, Nassir Navab, and Wolfgang Wein. Deriving anatomical context from 4d ultrasound. In Ivan Viola Ropinski, Katja Buehler, and Timo, editors, *Eurographics Workshop on Visual Computing for Biology and Medicine*, page 173–180. The Eurographics Association, 2014.

[MMNG15] Haichao Miao, Gabriel Mistelbauer, Christian Našel, and M Eduard Gröller. Cowradar: Visual quantification of the circle of willis in stroke patients. *Eurographics Workshop on Visual Computing for Biology and Medicine*, 2015.

[MPG$^+$17] Hannes Martinke, Christian Petry, Stefan Großkopf, Michael Suehling, Grzegorz Soza, Bernhard Preim, and Gabriel Mistelbauer. Bone Fracture and Lesion Assessment using Shape-Adaptive Unfolding. In Stefan Bruckner, Anja Hennemuth, Bernhard Kainz, Ingrid Hotz, Dorit Merhof, and Christian Rieder, editors, *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2017.

[MTDL05] D. Moeglin, C. Talmant, M. Duyme, and A. C. Lopez. Fetal lung volumetry using two- and three-dimensional ultrasound. *Ultrasound in Obstetrics and Gynecology*, 25(2):119–127, 2005.

[MTLT88] Nadia Magnenat-Thalmann, Richard Laperrière, and Daniel Thalmann. Joint-dependent local deformations for hand animation and object grasping. *Proceedings on Graphics Interface '88*, pages 26–33, 1988.

[NM10] Megumi Nakao and Kotaro Minato. Physics-based interactive volume manipulation for sharing surgical process. *IEEE Transactions on Information Technology in Biomedicine*, 14(3):809–816, 2010.

[NPH$^+$03] T. R. Nelson, D.H. Pretorius, A. Hull, M. Riccabona, M.S. Sklansky, and G. James. Sources and impact of artifacts on clinical three-dimensional ultrasound imaging. *Ultrasound in Obstetrics and Gynecology*, 16(4):374–383, 2003.

[NUX02] L.G. Nyul, J.K. Udupa, and Xuan Zhang. New variants of a method of mri scale standardization. *IEEE Transactions on Medical Imaging*, 19(2):143–150, 2002.

[OKG$^+$06] S. Oeltze, A. Kuß, F. Grothues, A. Hennemuth, and B. Preim. Integrated Visualization of Morphologic and Perfusion Data for the Analysis of Coronary Artery Disease. In Beatriz Sousa Santos, Thomas Ertl, and Ken Joy, editors, *EUROVIS - Eurographics /IEEE VGTC Symposium on Visualization*. The Eurographics Association, 2006.

[PB14] Bernhard Preim and Charl P. Botha. *Visual Computing for Medicine*. Elsevier, 2014.

[Ped15] Christian Pederkoff. stl-to-voxel. `https://github.com/cpederkoff/stl-to-voxel`, 2015. Github, Accessed: 2019-03-10.

[PHK04] S. Pieper, M. Halle, and R. Kikinis. 3D Slicer. In *2004 2nd IEEE International Symposium on Biomedical Imaging: Nano to Macro (IEEE Cat No. 04EX821)*, volume 1, pages 632–635, April 2004.

[PSS$^+$03] Stephen M. Pizer, Kaleem Siddiqi, Gabor Székely, James N. Damon, and Steven W. Zucker. Multiscale medial loci and their properties. *International Journal of Computer Vision*, 55(2-3):155–179, 2003.

[RLT$^+$15] Helmut Ringl, Mathias Lazar, Michael Töpker, Ramona Woitek, Helmut Prosch, Ulrika Asenbaum, Csilla Balassy, Daniel Toth, Michael Weber, Stefan Hajdu, Grzegorz Soza, Andreas Wimmer, and Thomas Mang. The ribs unfolded - a ct visualization algorithm for fast detection of rib fractures: effect on sensitivity and specificity in trauma patients. *European Radiology*, 25(7):1865–1874, 2015.

[SBS$^+$17] Veronika Solteszova, Åsmund Birkeland, Sergej Stoppel, Ivan Viola, and Stefan Bruckner. Output-sensitive filtering of streaming volume data. *Computer Graphics Forum*, 36(1):249–262, 2017.

[SFH00] R. L. Schild, R. Fimmers, and M. Hansmann. Fetal weight estimation by three-dimensional ultrasound. *Ultrasound in Obstetrics and Gynecology*, 16(5):445–452, 2000.

[SFW$^+$14] Ari Shapiro, Andrew Feng, Ruizhe Wang, Hao Li, Mark Bolas, Gerard Medioni, and Evan Suma. Rapid avatar capture and simulation using commodity depth sensors. *Computer Animation and Virtual Worlds*, 25(3-4):201–211, 2014.

[SHW$^+$12] Veronika Solteszova, Linn Emilie Saevil Helljesen, Wolfgang Wein, Odd Helge Gilja, and Ivan Viola. Lowest-variance streamlines for filtering of 3d ultrasound. *Eurographics Workshop on Visual Computing for Biology and Medicine*, 2012.

[Spe] Specialboy. Beby fetus infant 3d model. `https://www.3dcadbrowser.com/3d-model/baby-fetus-infant`. Specialboy, Accessed: 2019-03-15.

[SS04] V. Singh and Deborah Silver. Interactive volume manipulation with selective rendering for improved visualization. *IEEE Symposium on Volume Visualization and Graphics*, M:95–102, 2004.

[TFM$^+$17] Julien Tierny, Guillaume Favelier, Michael Michaux, Charles Gueunet, and Joshua A. Levine. The topology toolkit. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):832–842, 2017.

[The10] The Squidifier. Detailed man. `https://www.turbosquid.com/3d-models/free-obj-mode-human/544305`, 2010. The Squidifier, Accessed: 2019-03-10.

[VBS$^+$13] I Viola, Å Birkeland, V Solteszova, L Helljesen, H Hauser, S Kotopoulis, and K Nylund. High-quality 3d visualization of in-situ ultrasonography. *Eurographics*, pages 1–4, 2013.

[Vin90] Leonardo Da Vinci. Vitruvian Man (Le proporzioni del corpo umano secondo Vitruvio), c.1490. Pen and ink with wash over metalpoint on paper, Gallerie dell'Accademia, Venice.

[Viz] VizLab. Volume skeleton matlab toolbox. `http://coewww.rutgers.edu/www2/vizlab/liliu/research/skeleton/VSK`. coewww.rutgers.edu, Accessed: 2019-03-10.

[VLP05] Tomaž Vrtovec, Boštjan Likar, and Franjo Pernuš. Automated curved planar reformation of 3d spine images. *Physics in Medicine and Biology*, 50(19):4527–4540, 2005.

[WD10] Bricker L Neilson JP Whitworth, M and T Dowswell. Ultrasound for fetal assessment in early pregnancy. *Cochrane Database of Systematic Reviews*, (4), 2010.

[YCSV07] S. Yagel, S. M. Cohen, I. Shapiro, and D. V. Valsky. 3d and 4d ultrasound in fetal cardiac scanning: A new look at the fetal heart. *Ultrasound in Obstetrics and Gynecology*, 29(1):81–95, 2007.