Summer 2016

# Digital human modeling capabilities for task-based survivability

Jacob Todd Kersten
*University of Iowa*

# Digital Human Modeling Capabilities for Task-Based Survivability

by

Jacob Todd Kersten

A thesis submitted in partial fulfillment
of the requirements for the Master of
Science degree in Electrical and Computer
Engineering in the Graduate College of
The University of Iowa

August 2016

Thesis Supervisor: Adjunct Professor Tim Marler

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

_____

MASTER'S THESIS

_____

This is to certify that the Master's thesis of

Jacob Todd Kersten

has been approved by the Examining Committee
for the thesis requirement for the Master of Science degree in
Electrical and Computer Engineering at the August 2016 graduation.

Thesis Committee: _____

Tim Marler, Thesis Supervisor

_____

Mona Garvin

_____

Marc Pizzimenti

_____

Kevin Kregel

_____

Jon Kuhl

# ACKNOWLEDGMENTS

This work would have never been possible without the guidance, motivation, and tireless demand for perfection provided by Tim Marler. His relentless work ethic is both ridiculously impressive and impressively ridiculous. I can only hope to acquire a small portion of his dedication to scientific endeavors. Another individual who was necessary in the completion of this work was Nic Capdevila. He provided unconditional and unwarranted help, guidance, and availability. Immense levels of support, both intellectually and emotionally, were received from Andy Taylor, Kim Farrell, and Chris Taylor. The amount of effort provided by these individuals cannot be overstated. Additional thanks goes to all of the friends, family members, and everyone at the Virtual Solider Research Lab for all of the support and encouragement.

This work was funded by the Office of Naval Research (ONR).

# ABSTRACT

Although modeling and simulation are fertile areas for research and development within medicine, education, and human factors, there is a growing need for fully integrated organ systems as part of any digital human model (DHM). This need is particularly high in task-based survivability assessment. However, the current static geometry used in DHM is insufficient for evaluating conditions during simulated task performance. This insufficiency is due to the fact that internal viscera are inherently non-rigid objects. Therefore, undesirable, and unrealistic behaviors occur when using static models to represent internal viscera as the DHM moves through a variety of postures.

The capacity for DHMs to take on a variety of postures and positions contributes to their overall usefulness in modeling and simulation. By using static models to represent internal viscera, errors in model behavior must be tolerated, or the DHM must be limited to a posture that matches the models' configurations. With the either option being undesirable there is a need to represent internal viscera using dynamic models. A dynamic model will allow for the geometry used in representing the internal viscera to deform as the DHM.

This work proposes a computational platform for controlling the motion and deformation of internal viscera models within a DHM through two components. The first component is a new method for maintaining a relative position within a dynamic character's mesh called skin-based parenting. The second component is a system which takes a free-from deformation technique used in artistic modeling and eliminates the manual input that is usually required. This platform produces representations of internal

viscera which conform to the character's posture and motion in real-time. As such, the environmental influences that relate to the position and orientation of internal viscera models within a DHM in a variety of postures can be assessed.

# PUBLIC ABSTRACT

Digital human modeling and simulation is becoming an integral part in research within the fields of medicine, education, and human factors. As digital human modeling progresses there is a growing need to represent not just the surface of the human, but also the internal organs and other viscera, such as veins and arteries. Current methods for representing internal viscera rely on the use of static models. But static models do not perform well as a digital human model (DHM) moves through a variety of postures, due to the fact that internal viscera are inherently non-rigid objects. This work presents methods for transforming static models into dynamic models by controlling their movements and causing them to deform based on the DHM's posture. With these dynamic models, better assessments of outside factors on internal viscera can be performed.

TABLE OF CONTENTS

LIST OF FIGURES ..... viii

LIST OF EQUATIONS ..... xii

LIST OF TABLES ..... xiii

CHAPTER 1 INTRODUCTION ..... 1

1.1 Problem Statement ..... 1
1.2 Literature Review ..... 3
1.2.1 Representing Internal Viscera in DHMs ..... 3
1.2.2 Parenting Techniques ..... 5
1.2.3 Deformable Models ..... 7
1.3 Motivation ..... 10
1.4 Objectives and Hypothesis ..... 10
1.4.1 Skin-based Parenting ..... 12
1.4.2 Free-form Deformation ..... 12
1.5 Overview ..... 13

CHAPTER 2 POSITIONING ..... 15

2.1 Introduction ..... 15
2.2 Approach: Initial Placement ..... 16
2.3 Approach: Scaling ..... 17
2.4 Approach: Parenting ..... 19
2.4.1 Joint-Based Parenting ..... 19
2.4.2 Skin-based Parenting ..... 23
2.5 Results ..... 34

CHAPTER 3 MODEL DEFORMATION ..... 39

3.1 Introduction ..... 39
3.2 Approach: Overview ..... 42
3.3 Approach: Control Point Implementations ..... 46
3.3.1 Bounding-box ..... 47
3.3.2 Uniform Distribution ..... 48
3.3.3 Dimension Independent Distribution ..... 50
3.3.4 Automatic Pruning ..... 50
3.4 Auto Control-Point Determination ..... 51
3.5 Results ..... 54

CHAPTER 4 POTENTIAL APPLICATION: SHOTLINE ANALYSIS OF PERSONAL PROTECTIVE EQUIPMENT ..... 64

4.1 Introduction ..... 64
4.2 Approach ..... 65
4.3 Results ..... 69

CHAPTER 5 THREAT AND INJURY DATA ..... 74

5.1 Introduction ..... 74
5.2 Approach: Point Cloud ..... 75
5.3 Approach: Mesh Coloring ..... 77
5.4 Approach: Mesh Generation ..... 79
5.5 Discussion ..... 82

CHAPTER 6 CONCLUSION ..... 83

6.1 Summary ..... 83
6.2 Discussion ..... 84
6.3 Future Work ..... 86

REFERENCES ..... 89

LIST OF FIGURES

Figure 1: Lack of appropriate response from child object when parent object is deformed with standard parenting relationship implemented ..... 7

Figure 2: Flowchart of platform for task-based survivability assessment ..... 11

Figure 3: Underlying skeletal structure represented as collection of joints ..... 20

Figure 4: Test Case 1: A cube is parented to the left shoulder joint and initially positioned within the torso region ..... 22

Figure 5: Test Case 1: The DHM's arm has been moved and the cube has undesirably moved into the DHM's arm ..... 22

Figure 6: The DHM Santos' skin-mesh in a variety of poses ..... 24

Figure 7: Determining vectors $\boldsymbol{V0}\ \boldsymbol{V1}\ \boldsymbol{V2}$ ..... 28

Figure 8: Determining intersection points $\boldsymbol{I0}\ \boldsymbol{I1}\ \boldsymbol{I2}$ ..... 28

Figure 9: Determining weighted vectors $Vw0\ Vw1\ Vw2$ ..... 29

Figure 10: Recalculating $\boldsymbol{Vt}$ from modified centroid positions ..... 29

Figure 11: Initial reference frame for skin-based parenting model ..... 32

Figure 12: Reference frame for skin-based parenting model calculated after change in posture ..... 33

Figure 13: Another example of the calculated reference frame for a skin-based parenting model ..... 33

Figure 14: Test Case 1: The DHM's arm has been moved and the cube has moved due to the skin-based parenting relationship ..... 34

Figure 15: Test Case 1: A cube using skin-based parenting is initially positioned within the torso region near the shoulder ..... 35

Figure 16: Test Case 2: A model placed within the forearm of a DHM has a joint-based parenting relationship established with the wrist ..... 36

Figure 17: Test Case 2: The model mimics the exact rotation applied to the wrist, thus over-rotating skin-mesh ..... 36

Figure 18: Test Case 2: The model rotates and moves out of the DHM's skin-mesh .....36

Figure 19: Test Case 2: A model is placed within the forearm of a DHM and has a skin-based parenting relationship established.....37

Figure 20: Test Case 2: The model rotates with the DHM's skin-mesh rather than a specific joint, thus maintaining a realistic motion .....37

Figure 21: Test Case 2: The model does not rotate or move outside of the DHM's skin-mesh .....37

Figure 22: A parallelepiped lattice of control points on the left and an individual cell on the right .....42

Figure 23: Initial control points shown on the left, and a modified control point is shown on the right.....44

Figure 24: Offset vectors are calculated and the resulting interpolated vector $c$ is applied to the vertex position.....46

Figure 25: A single cell of the FFD algorithm being applied to a model of the lungs .....47

Figure 26: Multiple cells of the FFD algorithm working to create overall non-linear deformations .....48

Figure 27: Deformable models using a bounding-box configurations, left, compared against models using user defined control points per dimension, right.....49

Figure 28: Deformable model with all user-specified control points on left, and pruned control points on right.....51

Figure 29: A collection of internal positions that could potentially be referenced by a control point.....53

Figure 30: Vein and artery models are placed within a DHM in a default position.....54

Figure 31: Static vein and artery models do not work well with changes in posture .....55

Figure 32: Deformable vein and artery models bend and move with changes in posture.....56

Figure 33: Static model of the lungs penetrates the DHM's skin-mesh in extreme posture.....59

Figure 34: Dynamic model of the lungs deforms to stay within the DHM's skin-mesh in extreme posture ..... 59

Figure 35: A static model of the lungs intersects the liver model as the DHM twists ..... 60

Figure 36: A Deformable model of the lungs deforms to twist around the liver as the DHM twists ..... 60

Figure 37: Separation occurs between static models of internal viscera ..... 61

Figure 38: Dynamic internal viscera organs deform to maintain filled space ..... 61

Figure 39: DHM with internal viscera performing a toe-touch ..... 62

Figure 40: DHM with internal viscera performing a twist at the belly ..... 62

Figure 41: DHM with internal viscera represented performing a lateral twist ..... 63

Figure 42: Shot line hit detection performed on DHM with internal viscera in city scene ..... 65

Figure 43: graphical user interface for shot line analysis before (top) and after (bottom) analysis ..... 67

Figure 44: shot line analysis of a PPE system with full-torso coverage ..... 68

Figure 45: shot line analysis of a PPE system with partial torso coverage ..... 68

Figure 46: Shot line assessment scene with dynamic internal viscera models ..... 69

Figure 47: Static (left) versus dynamic (right) scenarios for shot line assessment ..... 72

Figure 48: Chart comparing results of shot line assessment between static and dynamic internal viscera models ..... 73

Figure 49: Point cloud rendering of pressure data ..... 76

Figure 50: Point cloud of threat-score data ..... 76

Figure 51: A scene of models with their default appearances ..... 78

Figure 52: A scene of models with a 3D vertex shader visualizing volumetric data ..... 78

Figure 53: Marching cubes surface reconstruction (left) of point cloud data (right) ..... 81

Figure 54: Smoothed surface reconstruction geometry .....81

LIST OF EQUATIONS

Equation 1: Calculating vectors between polygonal face vertices.....26

Equation 2: Determining weighted vectors for skin-based parenting position.....26

Equation 3: Calculating centroid weights for skin-based parenting position .....27

Equation 4: Determining first reference vector for skin-based parenting reference frame .....30

Equation 5: Determining second reference vector for skin-based parenting reference frame .....30

Equation 6: Determining third reference vector for skin-based parenting reference frame .....30

Equation 7: Constructing 4x4 matrix reference frame for skin-based parenting orientation .....31

Equation 8: Calculating transformation matrix containing current rotations .....31

Equation 9: Euler angles for rotation are decomposed from a 4x4 matrix .....31

Equation 10: Calculating the offset vector for each control point.....43

Equation 11: Determining x, y, and z distances used in trilinear interpolation.....44

Equation 12: Calculating interpolation values through $x$-axis .....44

Equation 13: Calculating interpolation values through $y$-axis .....45

Equation 14: Calculating interpolation value for $z$-axis.....45

Equation 15: calculating a pseudo-random ray endpoint around the target DHM .....66

LIST OF TABLES

Table 1: Comparative features and performances of joint-based parenting and skin-based parenting .....38

Table 2: Shot line results for five trials with static models.....70

Table 3: Shot line results for five trials with dynamic models .....70

Table 4: Mean percentages and standard deviations of shot line percentages.....71

# CHAPTER 1 INTRODUCTION

## 1.1 Problem Statement

Techniques in modeling and simulation continue to increase in sophistication. The rate at which computations can be performed increases and the cost of these computations decreases. One result of this progress is the development of complex digital human models (DHMs). These models can be utilized to enhance the quality of any product or process that involves human interaction. DHMs accomplish this by allowing for the assessment of human interactions with digital representations of objects. In doing so they save time and money. Furthermore, when purpose of the product is providing safety to the user, the use of DHMs can result in systems that better reduce the overall potential for harm.

One area that can benefit from advances in simulation and modeling is task-based survivability analysis. DHMs can be assessed against environmental inputs to determine propensity for injury. These assessments necessitate accurate representations of DHMs as well as accurate representations of external threats. A specific use of task-based survivability is in the design and evaluation of personal protective equipment (PPE). A digitally designed PPE system, such as body armor, can be placed on a DHM and an assessment can be made concerning the extent to which the PPE system provides protection to the DHM. Furthermore, the modeling and simulation platform can provide

the ability to assess many PPE scenarios with varying PPE systems, threats, and DHM posture.

While full-body DHMs continue to increase in accuracy, many fall short in the simulation of internal viscera. Image capture and processing techniques have fed a growing collection of high-fidelity computational models of internal viscera. Additionally, there is substantial anecdotal knowledge of how organs move and deform within the body. However, little work has been performed to couple this knowledge with the abundant computational models to create a holistic simulation of how internal viscera move and deform within the body. As modeling and simulation are further utilized within medicine, education, and human factors research, the need for fully integrated organ systems has increased for DHMs.

The challenges involved with accurately representing internal viscera within a DHM are numerous and only increase in quantity as the DHM increases in sophistication. The scope of this thesis work aims to provide solutions to two of these problems. The first challenge is how to control the position of the internal viscera model such that it maintains an accurate relative internal position within the DHM throughout DHM motion and changes in posture. The second challenge is how to represent deformations of internal viscera such that the model minimizes intersection with the DHM skin model, minimizes intersection between internal viscera models, and eliminates undesired separation between internal viscera through DHM motion and changes in posture.

## 1.2 Literature Review

### 1.2.1 Representing Internal Viscera in DHMs

The field of DHM continues to increase in sophistication as computation increases in rate and decreases in cost. As part of this refinement, there is a desire to accurately represent internal viscera. The field of radiological science is one of the major driving forces behind this work. There is a desire to generate high-fidelity computational models, also called phantoms. These phantoms are used to assess the impact of radiation from various medical imaging methods on the organs it is imposed upon. In this field there are two primary methods for performing dosimetry calculations: stylized models and voxel-based models.

In 1980 a set of stylized models were created to mathematically represent the internal viscera of a human (Christy 1980). The models consisted of geometric primitives, such as cylinders and ellipsoids. Several refinements have been made to the initial models. However, in recent years there has been a push to replace them with voxel-based models (Zaidi and Xu 2007). Voxel-based models aim to take real medical imaging data from patients and construct three dimensional (3D) and four dimensional (4D), 3D models that change based on time, representations of internal viscera. Many voxel-based models have been developed (Zaidi and Xu 2007). However, no one model has been universally accepted for dosimetry calculations due to the inherent issues with capturing whole-body images and converting them to voxel data. One of the drawbacks of these

models is that their construction is based, almost exclusively, on static representations of the organs.

For both stylized and voxel-based models there is limited work focusing on dynamic models. Dynamic models refer to representations that can account for changes in internal viscera shape and orientation caused by external or internal forces. An example of an internal force is the cardiac cycle of the heart. As the heart beats, changes along its axes result in subtle morphological alterations that disrupt its relative position to surface reference points. An example of an external force is the change in shape and orientation internal viscera experience as a subject changes posture (e.g., standing vs. crouched). The work that has been done focuses on 4D cyclical processes, such as respiratory and cardiac cycles, as opposed to general whole-body motion (Segars, et al. 2008); (Zhu, et al. 2005).

There does exist commercially available software that provides an interactive representation of internal viscera within a DHM. Two of these software packages are *Biodigital Human* and *Cyber Anatomy*. *Biodigital Human* provides a thorough collection of models that make up the human body. However, these models are static and solely for educational purposes. *Cyber Anatomy* provides a less complete set of models. The software does allow for some model motion and deformation. However dynamic capabilities are limited to the skeleton and select muscles.

### 1.2.2 Parenting Techniques

Digitally rendered scenes are often comprised of many 3D objects. Subsets of these objects often share logical relationships with regard to their position, orientation, and scale. For example, a simplified model of the upper human torso could consist of individual models for the arms, hands, and the torso itself. In order to realistically depict motion in this scene, the arms must move relative to the torso and the hands must move relative to the arms. Scene graphs provide a method for representing these types of logical relationships (Hughes, et al. 2013). Scene graphs are generally implemented as a linked list of nodes, with each node representing one model in the scene. Additionally, each of these nodes contains transform information for the model, such as translation, rotation, and scale.

Most state of the art 3D graphical platforms have a built-in scene graph interface. With this functionality, it is programmatically trivial to impose position, rotation, and scale constraints between models in a scene. However, these scene graph interfaces are generally limited to a hierarchal ordering where any given object may have only a single parent. Having a single parent means that its transform is directly influenced by its parent transform. This parenting hierarchy is able to represent complex scenes comprised of objects with rigid attachments. But complexity increases when an object in the scene needs to receive transform influence from multiple objects. Custom designed software is generally required when multi-parent transforms are needed.

While it is possible to create code that enables an object in a scene to be geometrically influenced by multiple objects, it requires additional functionality to be implemented. Furthermore, the objects of influence still need to be identified and taken into account when establishing multi-object relationships. For objects that have complex or numerous logical relationships with other objects in the scene, a more generalized and automated system for maintaining position and rotation relationships is needed. These issues only consider static models. Additional complexity, of the level that traditional scene-graph hierarchies cannot handle, arises when attempting to establish relationships between deformable models. Figure 1 highlights one of the ambiguities that occurs when applying traditional parenting techniques to non-rigid bodies.

A system more sophisticated than traditional parenting techniques is needed in order to depict scenes involving deformable models with complex relationships. Ideally, this system would incorporate a simple interface where objects could be placed in the desired position relative to the objects with which it interacts, and relationships could be automatically established and enforced. This work proposes one possible implementation of such a system

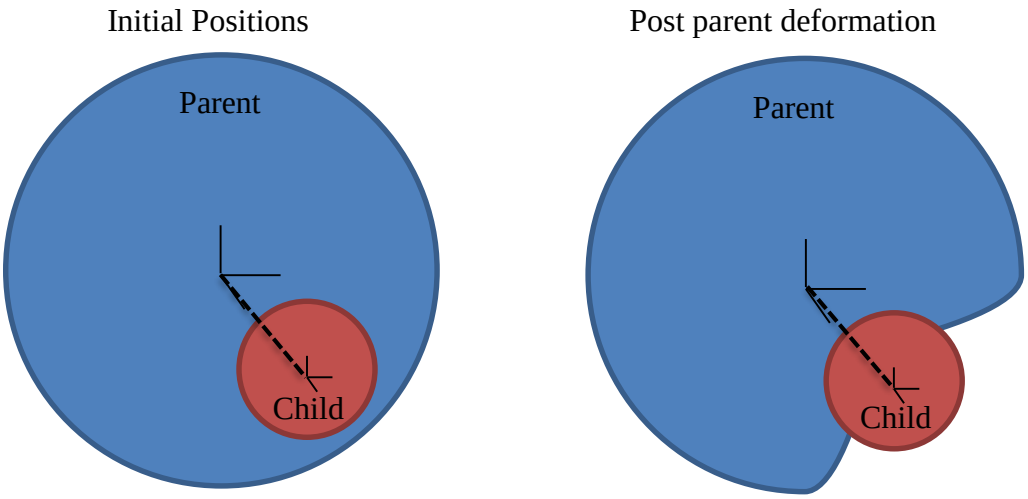


Figure 1: Lack of appropriate response from child object when parent object is deformed with standard parenting relationship implemented

### 1.2.3 Deformable Models

In the field of computer science there is a need to simulate real-world objects. A large portion of these objects are non-rigid and the standard static model representation is insufficient. Static models easily have real-time capabilities of global translation, rotation, and scale. As such, the positions of vertices relative to one another remain the same. In order to accurately represent non-rigid bodies, the relative positions between vertices must be mutable. Moreover, in the area of computer simulation, the modification of vertex positions must be performed at a rate that does not inhibit interactivity.

An early method for model deformation was presented by Barr in 1984. This method reduced deformations to bend, twist, taper, compression, and expansion (Barr 1984).

Subsequently, a variety of methods developed that build upon the initial deformation technique. Upon surveying these methods, they can be classified into three groups: artistic object modeling/animation, image segmentation, and interactive mechanical simulation (Meier, et al. 2005). Each of these fields imposes different constraints and challenges. Therefore, the methods adopt different characteristics to suit the needs of the field in which they will be applied. Of these fields, interactive mechanical simulation and artistic object modeling/animation are the most relevant to this work.

Algorithms designed for interactive mechanical simulation aim for volume conservation and realistic propagation of deformations. The most widely used algorithm in this field is the spring-mass model (Meier, et al. 2005). In this algorithm the desired object is reduced to a collection of masses, each of which are connected to a subset of its neighbors by springs. To determine the current state of deformation, one simply calculates the force equilibrium for each node using the Newtonian law of motion. A major drawback for this method is that large global deformations are not well handled. This is due to the inherent locally structured connections between masses and requires additional global links or complex tuning of link parameters in order to be mitigated. Another drawback, especially for use with DHMs, is the tendency for mass-spring models to oscillate. There is no intuitive method for setting the parameters of each mass and spring, and there is a high tendency for oscillation if the parameters are not properly tuned.

Deformation methods developed for artistic object modeling/animation consist primarily of modifications to the free-form deformation (FFD) technique presented by Sederberg

and Parry (Sederberg and Parry 1986). The FFD process involved encasing a model in a parallelepiped lattice of control points. Before deformation, each vertex in the model is mapped to a set of surrounding control points. Once the initial mapping is completed, each vertex's position is then computed by evaluating a tensor product trivariate Bernstein polynomial defined with respect to the current control points' positions. Several extensions of FFD focus on improving the "free-form" nature of the algorithm. It has been suggested that the parallelepiped structure for control points inherently limits certain types of deformation (Coquillart 1990).

The complexity of deformations created with FFD is limited by the density of control points in the enclosing lattice. In some cases, the desired deformation requires a large number of control points. As the number of these control points increases, manual manipulation becomes difficult since each point must be moved to reflect the desired deformation. In addition, there is the possibility that a number of the control points will be obscured by the model itself. Work has been done in attempt to alleviate these difficulties of working with FFD. A form of direct manipulation has been proposed that enables the user to specify the location of a set of vertices in the model and then compute the required FFD to meet those constraints (Hsu, Hughes and Kaufgamn 1992). Even with these improvements, FFD remains a deformation technique that requires a significant amount of manual input.

## 1.3 Motivation

Given the state of the art in representing internal viscera within DHM, there is a need to bridge the gap between static representations of internal viscera and dynamic representations in order to better perform task-based survivability assessments. Some work has been completed to simulate motion of internal viscera within a DHM, but this work is limited to fixed motions that do not take into account the general changes in posture and orientation of the DHM. To fill this gap, it is necessary to improve upon current methods for establishing logical relationships between internal viscera models and the DHM's overall position, orientation, and posture. Additionally, static representations of internal viscera must be replaced with dynamic representations. Moreover, these dynamic representations must support a variety of DHM postures and positions while requiring minimal user input. Finally, an overall modeling and simulation platform for assessing task-based survivability must be constructed.

## 1.4 Objectives and Hypothesis

The overall objective of this work is to construct a modeling and simulation platform for task-based survivability. This platform will support the process to formulate task-based survivability assessments (Figure 2). Given the large scope of this objective, only key elements of this platform will be targeted. The primary objective of this thesis work is to create a system for representing internal viscera within a DHM with dynamic models that move and deform in realistic manners as the DHM changes position and posture. This objective will be accomplished by first developing a novel method for establishing

position and orientation relationships between models and the skin-mesh of a DHM, referred to as skin-based parenting. Second, an FFD algorithm will be implemented and coupled with the skin-based parenting method to enable automatic deformation of internal viscera models as the DHM undergoes changes in posture. The secondary objective of this thesis work is to establish methods for performing task-based survivability assessments. This objective will be accomplished by first presenting a shot-line assessment tool, and second presenting an initial investigation into the visualization of injury and threat data. The hypothesis of this thesis is that increased accuracy in the modeling of internal viscera through the completion of the primary objective mentioned will increase the overall value and effectiveness of task-based survivability analysis.

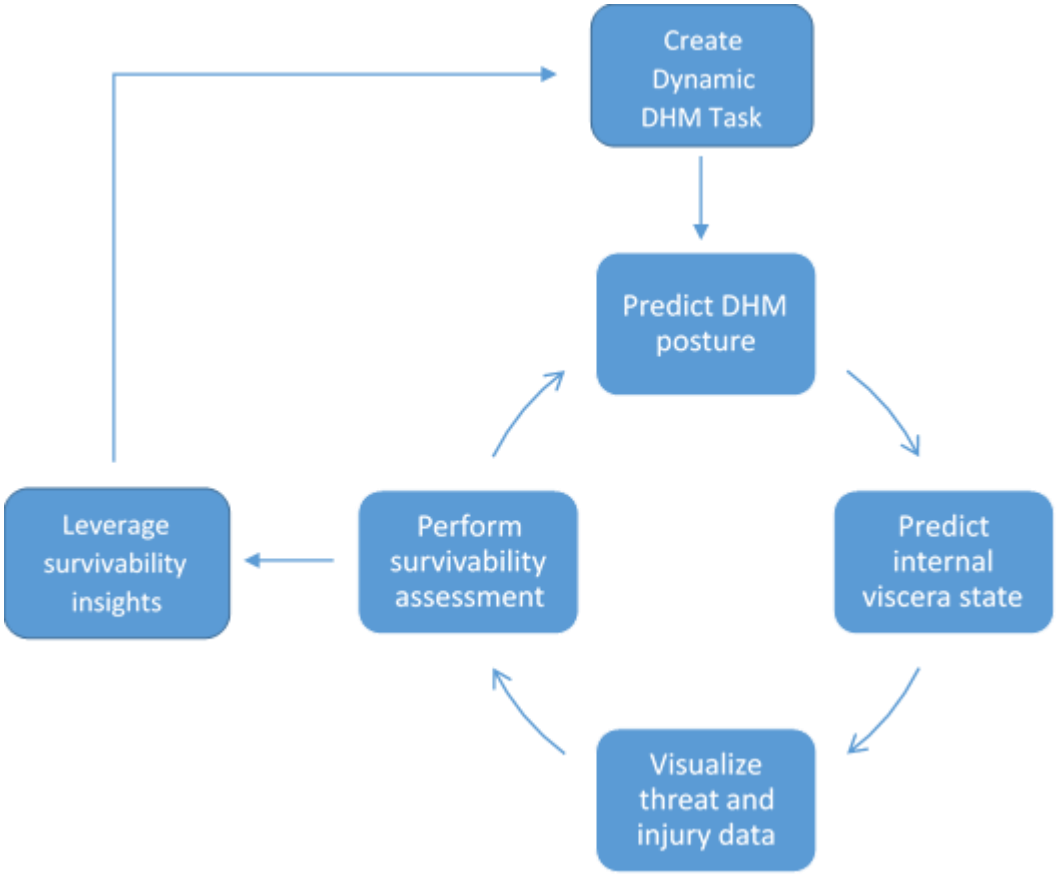


Figure 2: Flowchart of platform for task-based survivability assessment

### 1.4.1 Skin-based Parenting

An algorithm is developed which uses skin-mesh data local to a desired internal position to calculate and enforce position and orientation relationships. Furthermore, the algorithm requires only an initial internal position from the user. This algorithm reduces manual input required in establishing complex position and orientation relationships through traditional parenting methods. Additionally, the algorithm is agnostic of the underlying skeletal structure used to deform and position the DHM's skin-mesh, and could be used in other non-DHM applications involving deformable models.

### 1.4.2 Free-form Deformation

An FFD algorithm based on the algorithm introduced by Sederberg and Parry (Sederberg and Parry 1986) is implemented. This algorithm is used to deform internal viscera models. While traditionally this algorithm requires a significant amount of user input to generate deformations, a method for automatically coupling this algorithm with the skin-based parenting algorithm is presented. Through this coupling, the user input required for the FFD algorithm is reduced to an initial value indicating the desired resolution of the deformations. With the resolution specified, the internal viscera models are automatically deformed as the DHM transitions through a variety of postures.

## 1.5 Overview

Chapter 2 presents the need for maintaining position and orientation relationships when representing internal viscera models. The process for determining initial placement of the internal viscera models is described. Next, traditional parenting methods and their limitations within the focus of this work are presented. Finally, a novel method for maintaining position and orientation relationships through utilization of a DHM's skin-mesh is documented.

Chapter 3 covers the concept and approach for implementing an FFD algorithm that deforms the internal viscera models. A method for coupling this FFD algorithm with the skin-based parenting system is presented, and the results of coupling these two systems are shown. The resulting deformable representations of the internal viscera models are compared against their static counterparts.

Chapter 4 introduces shot line assessment, a method of statistical projectile tracing, for personal protective equipment evaluation and uses this application as a platform to compare static versus dynamic internal viscera models. An implementation of shot line assessment is presented, and an experiment is performed with both static internal viscera models and dynamic internal viscera models. The results are discussed, highlighting the limitations of static internal viscera models.

Chapter 5 presents an initial investigation into methods for visualizing injury and threat data. The first method presented is point cloud rendering, for which the benefits and drawbacks are discussed. The drawbacks of point cloud rendering lead to the visualization method of mesh coloring. The advantages of mesh coloring are presented and its limitations lead to the presentation and discussion of mesh generation.

Chapter 6 consists of a summary, discussion, and future work.

# CHAPTER 2 POSITIONING

This chapter provides the procedure and results of constructing a system for maintaining an accurate internal position relative to a DHM's skin mesh. Initial positions for several internal viscera are determined through examination of literature and medical diagrams, as well as through review and input from medical professionals. A traditional joint-based parenting method is implemented and evaluated. Then a novel parenting method that utilizes the positional data contained within the DHM's skin mesh is presented and evaluated. Finally, the presented skin-parenting method is compared against the traditional joint-based method.

## 2.1 Introduction

In order to accurately represent internal viscera within a DHM, it is necessary to establish and maintain a correct internal position. This requires two steps. First, the internal viscera model must be placed within the DHM at the correct position and with the correct orientation. Second, the internal viscera model must maintain a correct internal position relative to the DHM's skin surface as the DHM changes position and posture. Initial placement and orientation of the models can be a one-time manual process aided by medical images and text as well as guidance from medical professionals. This approach can lend insight to organ configuration within a limited number of postures. However, it is infeasible for the models to be manually positioned and oriented for every possible posture that the DHM can achieve.

Traditional methods for maintaining position and orientation constraints are sufficient for establishing relationships between objects that have well-defined, rigid attachments. But, these methods fall short with objects, such as internal viscera, that do not have well-defined attachment points. This chapter provides the design and implementation of a novel method for maintaining internal position and orientation of models within a DHM. Furthermore, this system requires only an initial placement and orientation of the models within the DHM. No attachment points or constraints are needed, and the internal position is automatically calculated and enforced as the DHM changes in position and posture.

## 2.2 Approach: Initial Placement

The first step in maintaining accurate position and orientation for internal viscera is to determine the correct position and orientation within the DHM in a default posture. Initial positions for internal viscera are determined through consulting medical illustrations, images, and professionals. This step is non-trivial and is susceptible to criticism for a number of reasons. First, rarely are two medical illustrations of internal viscera identical. Second, precise placement, size, and orientation of organs varies between persons. Third, the positions and orientations of internal viscera are impacted by the posture and orientation, relative to gravity, of the person. These issues have been mitigated to some degree through assumptions of initial orientation and posture of the DHM, as well as the assumption that the initial placement is accurate for the DHM representing a single

person. Furthermore, the initial placements have received several rounds of revisions through a counsel of medical experts.

## 2.3 Approach: Scaling

A step that overlaps with the initial placement of the internal viscera is to determine the correct size of internal viscera relative to the anthropology of the DHM in which they are being simulated. That is, how can the internal viscera models be scaled to fit DHMs of varying height, weight, and shape? An initial attempt to quantify sizes of internal viscera based on anthropometric variations by means of a literature-based is presented.[1]

The first consideration for a scaling effort is determining whether to create separate scaling models for male and female or to use a single model for both genders. For this initial effort it was deemed unnecessary to construct separate models when scaling according to body index (D'Oronzio, et al. 2012) (Ungerleider and Clark 1939). As the available data in literature do not support generalized models for each individual internal organ, two models were generated for specific internal viscera. A model for the heart is developed from a study using x-ray images of 1,500 chests to measure the transverse diameter of the heart (D'Oronzio, et al. 2012). The result of the study is a table of heart sizes is created by comparing the collected heart measurements to the recorded height and weight of the subjects. A statistically-refined model is produced from the table

[1] All research for internal viscera scaling based on anthropology was performed by Samantha Wagner

presented in the study. This model derives the transverse diameter of the heart from the body index of weight over height, with a 0.9882 correlation coefficient.

$$Heart\ Diameter(in) = 8.6566\left(\frac{in^2}{lb}\right) \times \left(\frac{Weight\ (lb)}{Height\ (in)}\right)^{0.4895}$$

Similarly, a model is developed for the lungs based on a study that uses 2,500 x-ray images to measure the heart and chest diameter to create heart and lung coefficients (Cowan 1964). The resulting heart and lung coefficients of the study are used to derive the transverse diameter of the chest. The transverse diameter of the chest is equivalent to the width of the lungs at full inspiration. The model generates the transverse diameter of the chest from the transverse diameter of the heart with a 0.9204 correlation coefficient.

$$Lung\ Width\ (in) = 1.9381\ \times\ (Heart\ Diameter)^{1.083}$$

The remaining internal viscera models are scaled using the lung scaling model. The scaling occurs by first calculating the DHM's body index. Next the heart diameter is determined and scaled accordingly. Then the lungs and remaining internal viscera are scaled. As these scaling models are based on pre-existing study data, they do not contain comprehensive size attributes. For example, a parameter currently missing that is necessary for accurate scaling is the lung height. Further scaling work would benefit from a study specialized in acquiring internal viscera dimensions based on anthropology.

## 2.4 Approach: Parenting

The final step in maintaining accurate position and orientation for internal viscera is to have the ability to determine the correct position and orientation within the DHM in any posture. Repeating the manual process discussed in the previous section for every posture the DHM is capable of achieving is both inefficient and impractical. The alternative is to establish relationships which cause the models to move and rotate, in a desired fashion, as the DHM transitions between various postures.

### 2.4.1 Joint-Based Parenting

The position, and posture of a DHM is often driven by an underlying skeletal structure that is an idealized version of a real human's skeletal system. For example, the Santos model [Yang, ほか 2007] created at the University of Iowa uses kinematic joints connected by rigid links as shown in Figure 3. Since these joints control the overall position and posture of the DHM, it is logical to consider them good candidates for establishing parent-child relationships with internal viscera models. A relationship can be

established between the model and a specific joint, and as the joints are moved and rotated, so too will the internal viscera model move and rotate.

The first step for establishing a joint-based relationship is to place the internal viscera model within the DHM at the desired position and orientation. Next, a joint must be

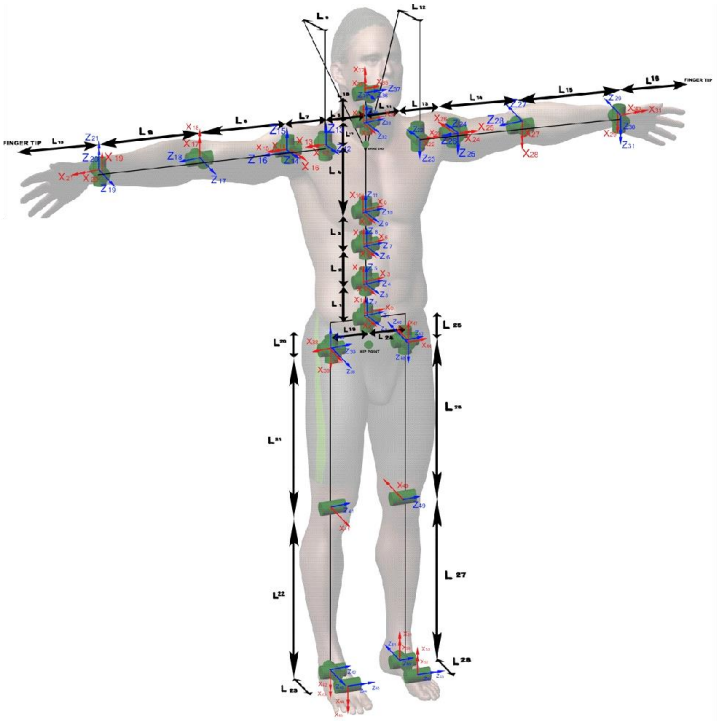

Figure 3: Underlying skeletal structure represented as collection of joints

selected as the parent to this model. Intuitively, the joint which is closest to the model's initial position can be selected as the parent. With a parent-child relationship defined between a joint and the model, any resulting changes in position or rotation of the joint will be reflected in the position and orientation of the child.

In cases where the closest joint does not produce the motion desired from the internal viscera model, other joint parents can be experimentally chosen to determine the best joint parent. This process of manually determining the correct parent for an internal viscera model quickly becomes tedious as the number of models increases. Furthermore, it is often challenging for a user to determine which joint should be used as the parent, such as when the initial position of the model places it in equal distance from several joints. Additionally, there are cases where a single parent joint is insufficient in creating the desired motion for a model. This can be seen when we attempt to place a model, represented by a cube, around the DHM's shoulder in Test Case 1. Due to the cube's initial positioning within the shoulder region of the DHM, the left should joint is selected as the model's parent. In Figure 4 we can see the position of the model is near the shoulder blade. But in Figure 5, when the arm is moved down toward the DHM's side, the cube moves from near the should blade and down into the arm near the triceps. In cases like this, it is necessary to consider position and rotation relationships between the model and several reference joints. This capability is not inherent in traditional parent-child relationships. Furthermore, if the traditional parenting method was expanded to

allow for multiple parents, the manual process of determining which parents to use for a given internal viscera model would still be cumbersome.

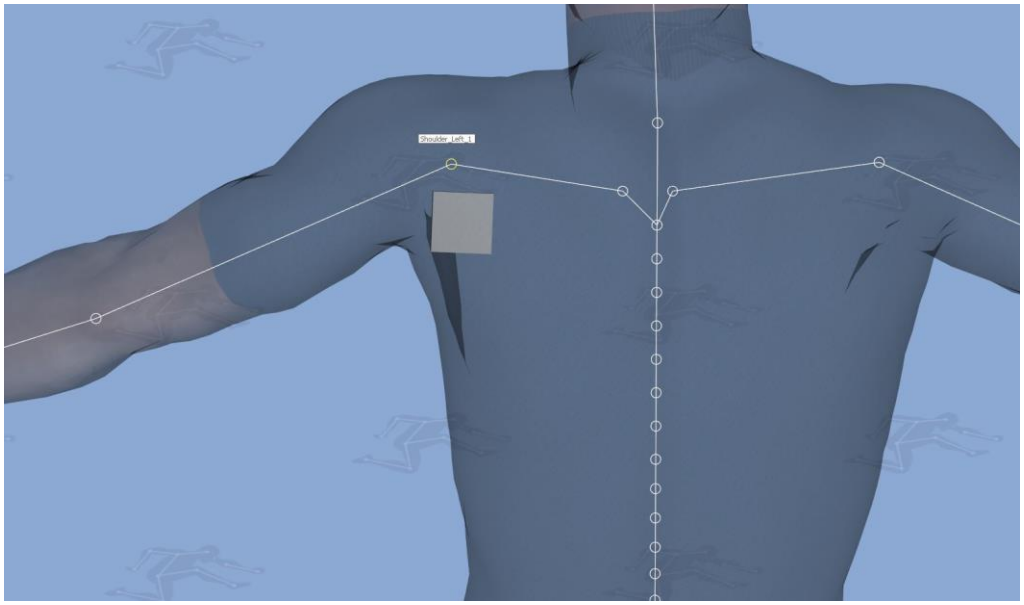

Figure 4: Test Case 1: A cube is parented to the left shoulder joint and initially positioned within the torso region

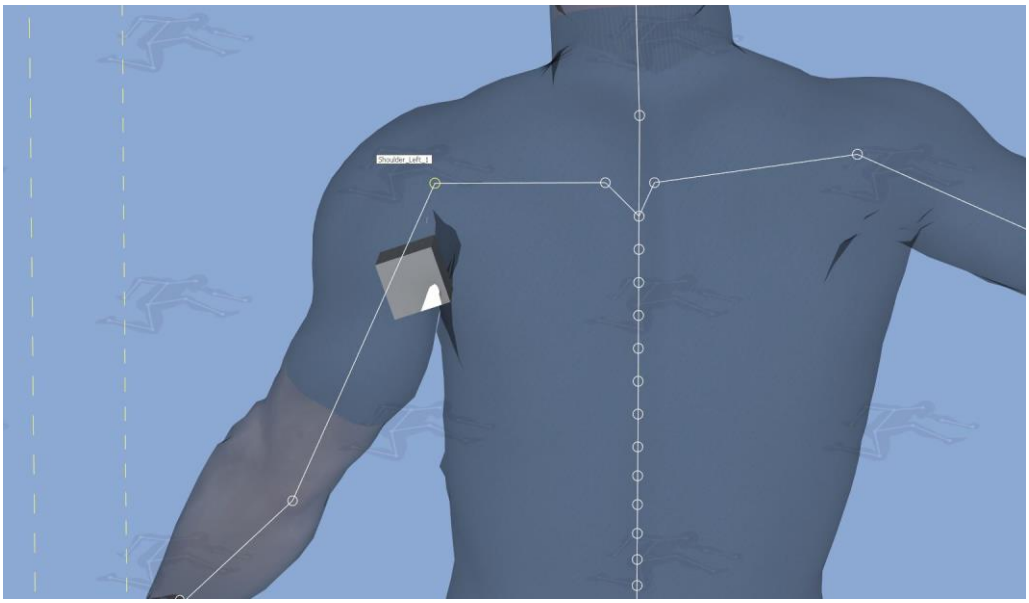

Figure 5: Test Case 1: The DHM’s arm has been moved and the cube has undesirably moved into the DHM's arm

### 2.4.2 Skin-based Parenting

A skin-mesh representing the skin of a human often accompanies a DHM. This mesh is a visual representation of the DHM and is comprised of a high number of polygonal faces. A skin-mesh, like most digital models, is inherently static. This means that the polygonal faces do not change their positions relative to one another. Thus, the default skin-mesh cannot be put into a variety of postures.

In order to make the skin-mesh dynamic, it needs to undergo a process called character skinning. A summary of character skinning is presented by (Lewis, Cordner and Fong 2000). In short, this process involves defining positional relationships between the vertices of a mesh and the underlying skeletal structure of the character. At the end of this process every vertex in every polygonal face in the skin-mesh has a list of corresponding skeletal joints, each having a weight assigned to it. This information is used to transform the position of each vertex as its corresponding skeletal joints change position and rotation. The result being a dynamic skin-mesh that moves and deforms as the underlying structure is changed. The DHM can then be put into a variety of postures as shown in Figure 6.

A skinned mesh contains a substantial amount of information about how the mesh should move relative to changes in its underlying skeletal structure. The presence of this data and the intuitive observation that the internal viscera being represented should, generally, not move outside of the skin-mesh led to an effort to utilize the skin-mesh itself to

establish logical relationships for the internal viscera models. The positions of polygonal faces within the DHM's skin-mesh would be used to track and enforce the position and orientation of an internal viscera model. This technique will be referred to as skin-based parenting.

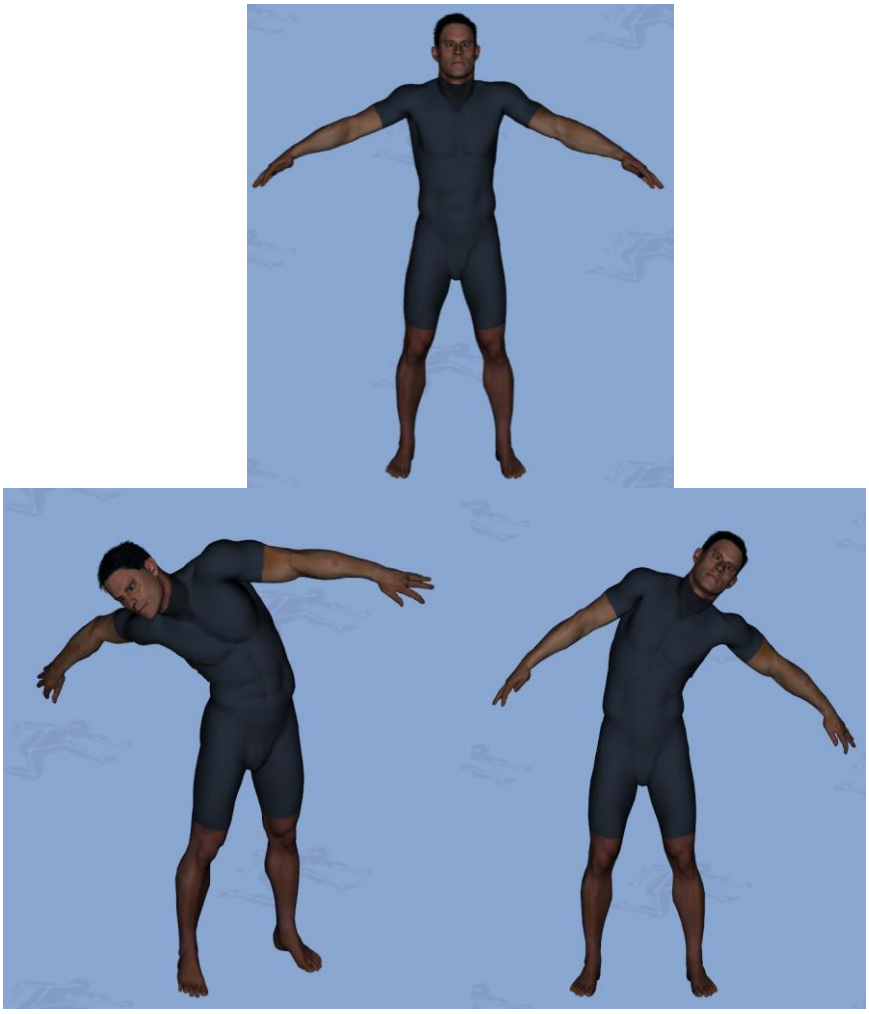

Figure 6: The DHM Santos' skin-mesh in a variety of poses

The first, and only, user inputs required to establish a skin-based parenting relationship are the initial position $\boldsymbol{V_t}$ and orientation $\boldsymbol{Q_t}$ of the internal viscera model. $\boldsymbol{V_t}$ is a 3D

vector in global space, and $\boldsymbol{Q_t}$ is a 3D vector containing Euler angles. The position $\boldsymbol{V_t}$ must satisfy the requirement of being inside the DHM's skin-mesh. Once the initial position and orientation have been defined the developed algorithm automatically establishes and enforces the logical relationship between that position and the DHM's skin-mesh.

First, a set of three rays are cast away from $\boldsymbol{V_t}$ towards the DHM's skin mesh. These three rays satisfy the condition that they form a plane on which $\boldsymbol{V_t}$ lies. A ray-intersection algorithm is used to determine the polygonal faces in the skin-mesh that each ray intersects. Since $\boldsymbol{V_t}$ satisfies the condition that it is within the mesh of the DHM, each ray will intersect exactly one polygonal face on the skin-mesh. Each of the polygonal faces in the skin-mesh is made up of three vertices, and each vertex is identified by a vertex index. The three vertex indices are stored for each of the three rays in order to access them later. The positons of these vertices will be used to update the internal position as the skin-mesh moves and deforms.

In order to calculate the updated position as the skin-mesh moves and deforms, additional information must be calculated. The centroids $\boldsymbol{C_0 C_1 C_2}$ of each set of three vertices are calculated. The initial position $\boldsymbol{V_t}$ is guaranteed to lie on a plane with $\boldsymbol{C_0 C_1 C_2}$, but it is not guaranteed to be in the center of the three centroids. Therefore, a weight for each centroid needs to be calculated in order to maintain the desired relative position of $\boldsymbol{V_t}$ between $\boldsymbol{C_0 C_1 C_2}$. The three weighted vectors are used to determine the position of $\boldsymbol{V_t}$ based on the position of the centroids. In order to calculate these weights a series of

calculations are needed. First, the vectors between each of the centroids, seen in Figure 7, are calculated:

$$\boldsymbol{V_0 = C_2 - C_1}$$

$$\boldsymbol{V_1 = C_0 - C_2}$$

$$\boldsymbol{V_2 = C_1 - C_0}$$

Equation 1: Calculating vectors between polygonal face vertices

Next, intersection points $\boldsymbol{I_0\ I_1\ I_2}$ on each of those vectors are determined by tracing a line from the opposite centroid through $\boldsymbol{V_t}$ as seen in Figure 8. Using these intersection points, the vectors $\boldsymbol{V_{w0}\ V_{w1}\ V_{w2}}$, shown in Figure 9, are calculated:

$$\boldsymbol{V_{w0} = I_0 - C_1}$$

$$\boldsymbol{V_{w1} = I_1 - C_2}$$

$$\boldsymbol{V_{w2} = I_2 - C_0}$$

Equation 2: Determining weighted vectors for skin-based parenting position

The weights $w_0\ w_1\ w_2$ are calculated so that the position of $\boldsymbol{V_t}$ relative to $C_0, C_1,$ and $C_2$ is maintained when an updated position $\boldsymbol{V_t}'$ is calculated:

$$w_0 = \frac{|\boldsymbol{V_{w0}}|}{|\boldsymbol{V_0}|}$$

$$w_1 = \frac{|\boldsymbol{V_{w1}}|}{|\boldsymbol{V_1}|}$$

$$w_2 = \frac{|\boldsymbol{V_{w2}}|}{|\boldsymbol{V_2}|}$$

Equation 3: Calculating centroid weights for skin-based parenting position

With these weights and their corresponding centroid positions, the target internal position can be recalculated even when the centroids change their positions. This is done by determining the intersections of lines between each of the weighted vectors $\boldsymbol{V_{w0}}\ \boldsymbol{V_{w1}}\ \boldsymbol{V_{w2}}$ and their corresponding centroids $\boldsymbol{C_0 C_1\ C_2}$. These lines and their intersections can be seen in Figure 10. Through this calculation, the relationship between the position of the internal viscera position and the skin-mesh of the DHM is established. Thus future positions of $\boldsymbol{V_t}$ can be calculated, as shown in Figure 10.

This method for determining a "weighted" centroid of a triangle is just one of many potential solutions. This method is chosen for two reasons. First, it is simple to implement. Second, this method results in an updated position $\boldsymbol{V_t}'$ that always lies within the boundary of the triangle. Staying within the boundary of the triangle guarantees that the internal position remains within the DHM's skin mesh.

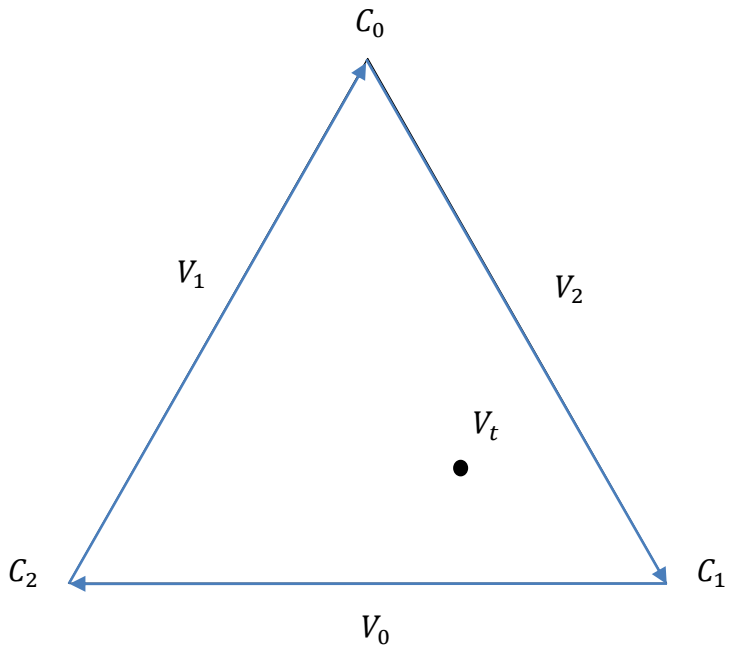


Figure 7: Determining vectors $\boldsymbol{V_0}\ \boldsymbol{V_1}\ \boldsymbol{V_2}$

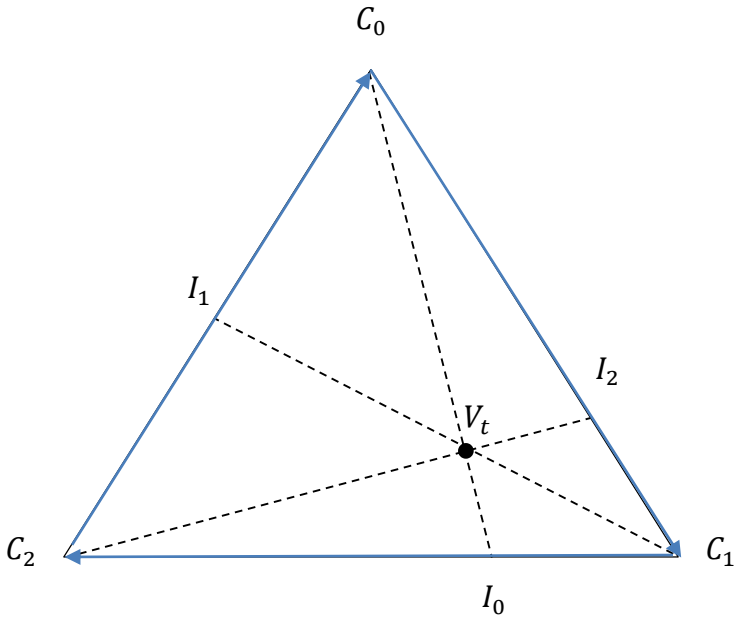


Figure 8: Determining intersection points $\boldsymbol{I_0}\ \boldsymbol{I_1}\ \boldsymbol{I_2}$

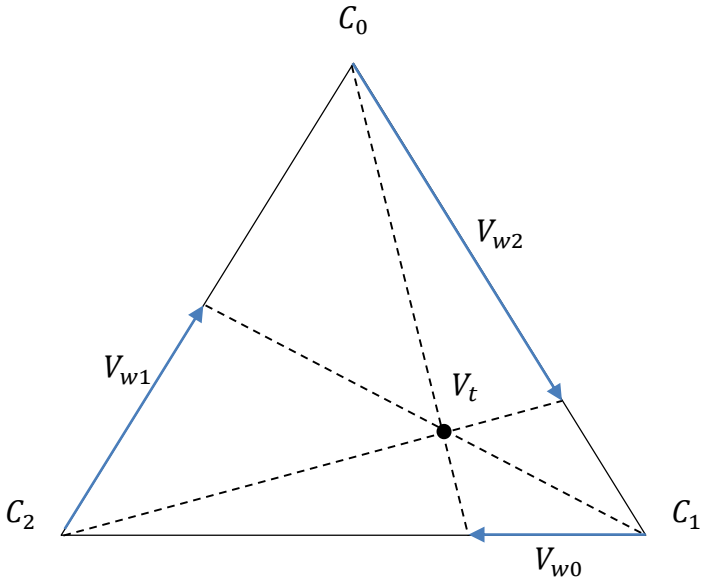


Figure 9: Determining weighted vectors $V_{w0}$ $V_{w1}$ $V_{w2}$

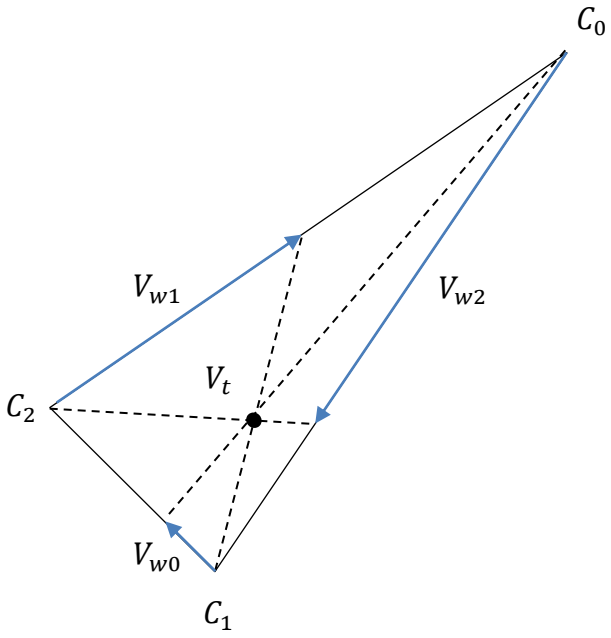


Figure 10: Recalculating $\boldsymbol{V_t}$ from modified centroid positions

In order to maintain an orientation relationship between the internal viscera model and the DHM's skin-mesh, an initial reference frame must be established. This reference frame is composed of three vectors that are all perpendicular to one another. The first vector of the reference frame is determined by calculating the vector perpendicular to the plane of the three centroid positions $\boldsymbol{C_0C_1C_2}$:

$$\boldsymbol{V_{r0} = (C_1 - C_0) \times (C_2 - C_0)}$$

Equation 4: Determining first reference vector for skin-based parenting reference frame

The second vector is calculated by determining the vector that starts at the position $\boldsymbol{V_t}$ and ends at centroid position $\boldsymbol{C_0}$:

$$\boldsymbol{V_{r1} = C_0 - V_t}$$

Equation 5: Determining second reference vector for skin-based parenting reference frame

The third vector is calculated by determining the cross product of $\boldsymbol{V_{V0}}$ and $\boldsymbol{V_{r1}}$ :

$$\boldsymbol{V_{r2} = V_{r0} \times V_{r1}}$$

Equation 6: Determining third reference vector for skin-based parenting reference frame

A 4x4 matrix $M_r$ is constructed from the calculated reference vectors $\boldsymbol{V_{r0}}$, $\boldsymbol{V_{r1}}$, and $\boldsymbol{V_{r2}}$:

$$M_r = \begin{bmatrix} V_{r0}x & V_{r1}x & V_{r2}x & 0 \\ V_{r0}y & V_{r1}y & V_{r2}y & 0 \\ V_{r0}z & V_{r1}z & V_{r2}z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 7: Constructing 4x4 matrix reference frame for skin-based parenting orientation

By repeating this operation each time the skin-mesh deforms or moves, causing the centroids to move, a current reference frame can be computed. To determine the desired orientation of the internal viscera model based on the current centroid locations, first the updated reference frame $\boldsymbol{M_r}'$ is calculated following the same steps as the calculation of $\boldsymbol{M_r}$, but with the use of the current centroid positions. Then $\boldsymbol{M_r}'$ can be multiplied by the original reference frame $\boldsymbol{M_r}$ transposed:

$$\boldsymbol{M_c} = \boldsymbol{M_r}^{\mathrm{T}} * \boldsymbol{M_r}'$$

Equation 8: Calculating transformation matrix containing current rotations

Next, Euler angles $\theta_x\ \theta_y\ \theta_z$ are extracted from the resulting matrix:

$$\theta_x = \tan^{-1}(M_{c32}, M_{c33})$$

$$\theta_y = \tan^{-1}(-M_{c31}, \sqrt{{M_{c32}}^2 + {M_{c33}}^2})$$

$$\theta_z = \tan^{-1}(M_{c21}, M_{c11})$$

Equation 9: Euler angles for rotation are decomposed from a 4x4 matrix

These angles are used to set the orientation for the internal viscera model and reflect the local change in orientation of the three centroids. As the skin-mesh of the DHM moves and deforms, the three reference centroids move. Through the changes in these centroids an updated reference frame can be determined. The desired rotation can be determined by comparing the current reference frame with the initial reference frame. An initial reference frame is shown in Figure 11, and updated reference frames are shown in Figure 12 and Figure 13.

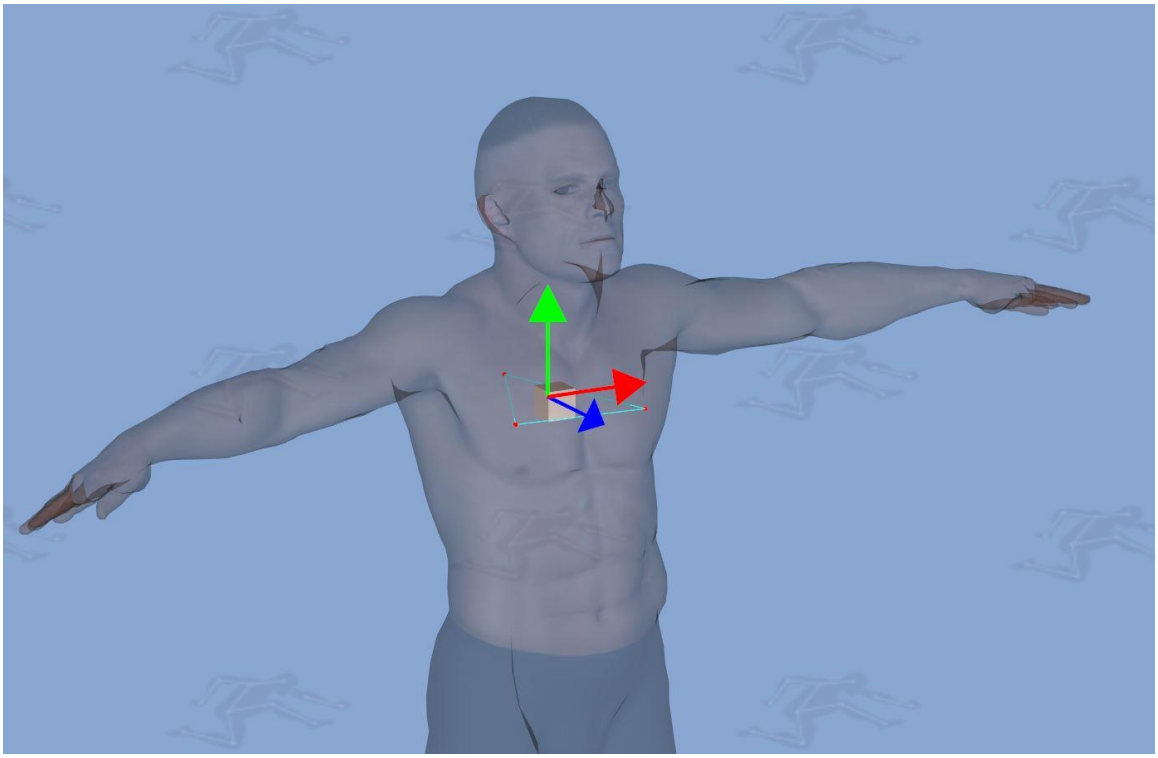

Figure 11: Initial reference frame for skin-based parenting model

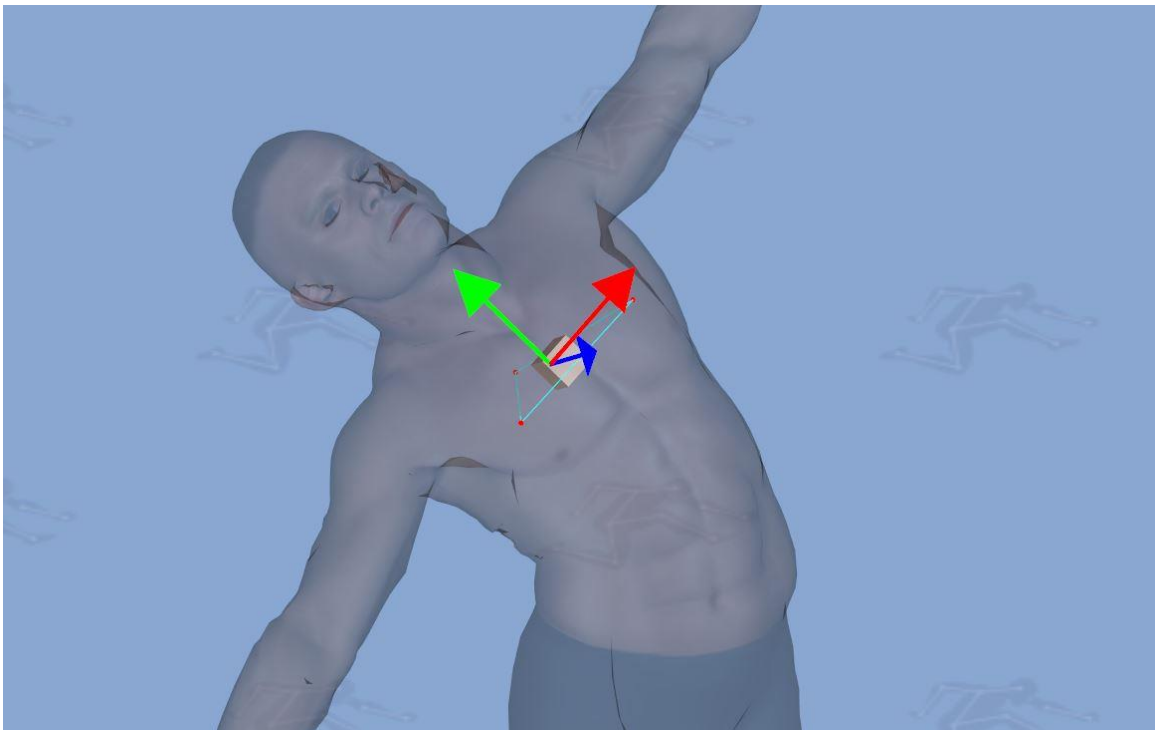

Figure 12: Reference frame for skin-based parenting model calculated after change in posture

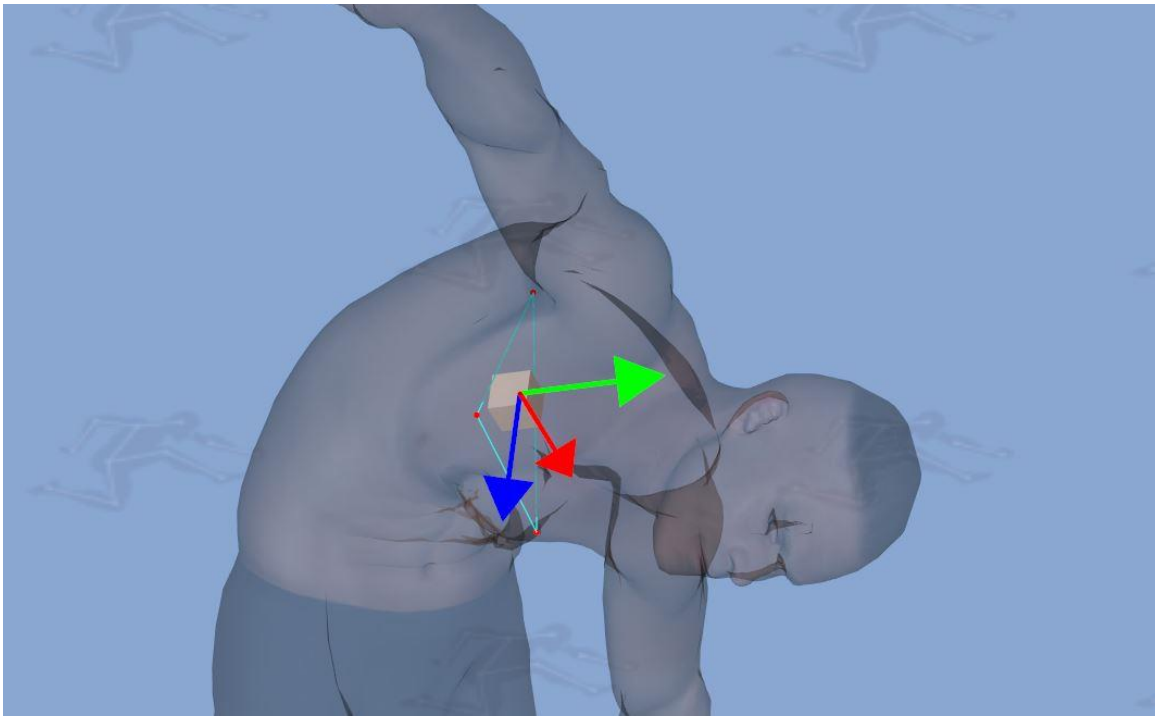

Figure 13: Another example of the calculated reference frame for a skin-based parenting model

## 2.5 Results

With both position and orientation relationships maintained, the internal viscera models move and rotate in a way that naturally follows the motion and deformation of the DHM's skin-mesh. This is evident in Figure 14 and Figure 15 where skin-based parenting is used in place of the joint-based parenting example mentioned in Test Case 1 in section 2.4.1.

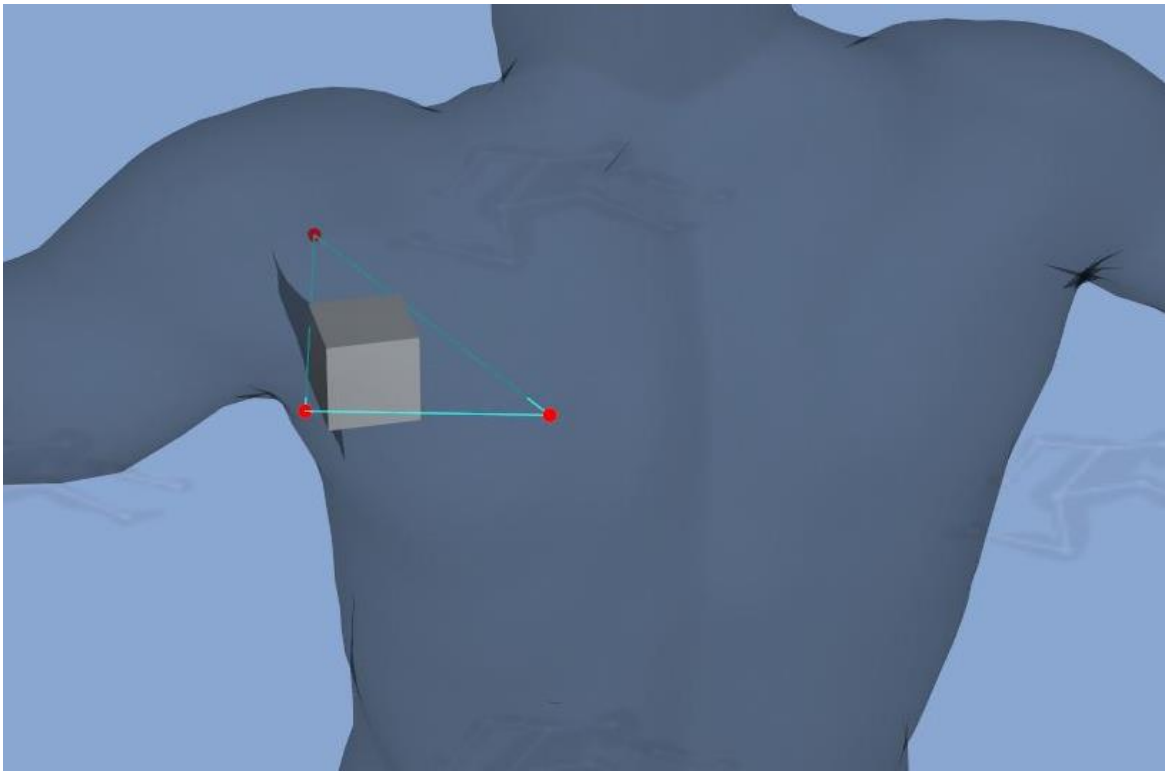

Figure 14: Test Case 1: The DHM's arm has been moved and the cube has moved due to the skin-based parenting relationship

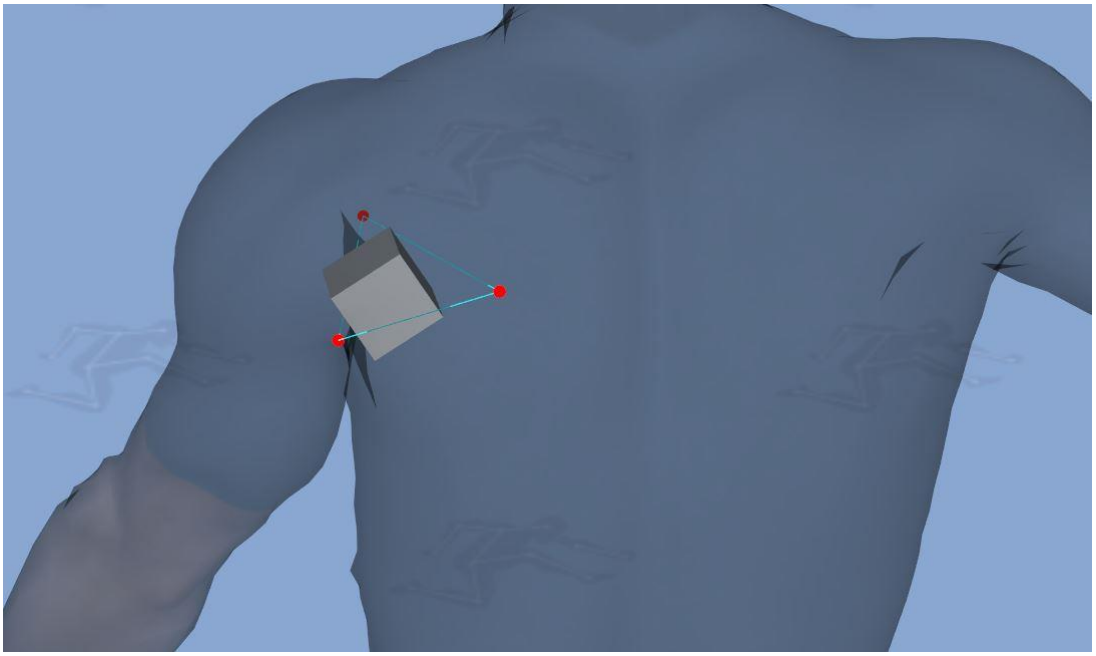

Figure 15: Test Case 1: A cube using skin-based parenting is initially positioned within the torso region near the shoulder

Furthermore, the only required knowledge for the internal viscera model was an initial position. Another example of skin-based parenting's superior performance over joint-based parenting can be seen in Figures 16-21 when a model is placed between the elbow and wrist of a DHM. Enforcing the correct orientation of the model is non-trivial as both the rotation of the wrist and elbow must be taken into account. This would normally require the user to have an understanding of the underlying skeletal structure of the DHM. However, with skin-based parenting, the user must simply place the model in the desired position and the algorithm enforces the desired orientation automatically. Thus, the skin-based parenting method results in a more desirable motion of the model and requires less initial time and knowledge from the user.

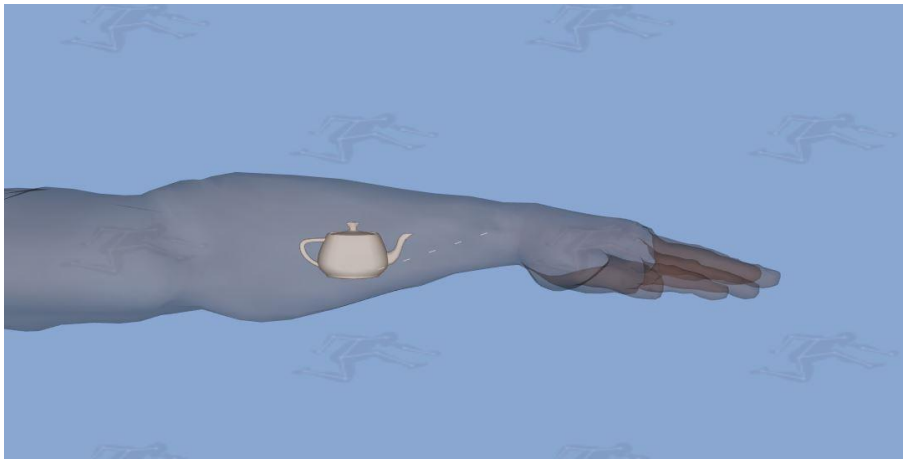

Figure 16: Test Case 2: A model placed within the forearm of a DHM has a joint-based parenting relationship established with the wrist

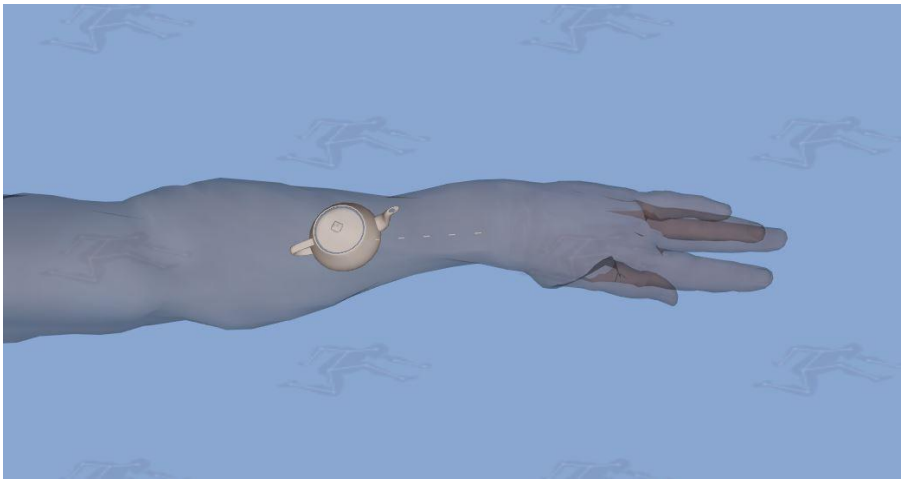

Figure 17: Test Case 2: The model mimics the exact rotation applied to the wrist, thus over-rotating skin-mesh

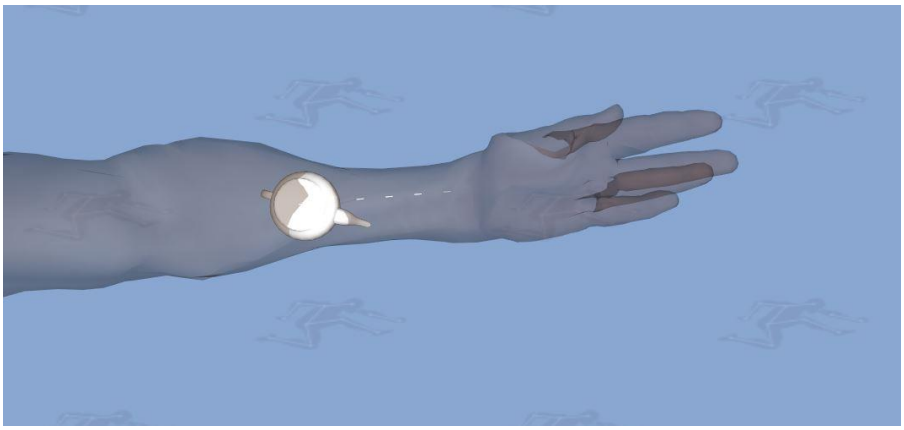

Figure 18: Test Case 2: The model rotates and moves out of the DHM's skin-mesh

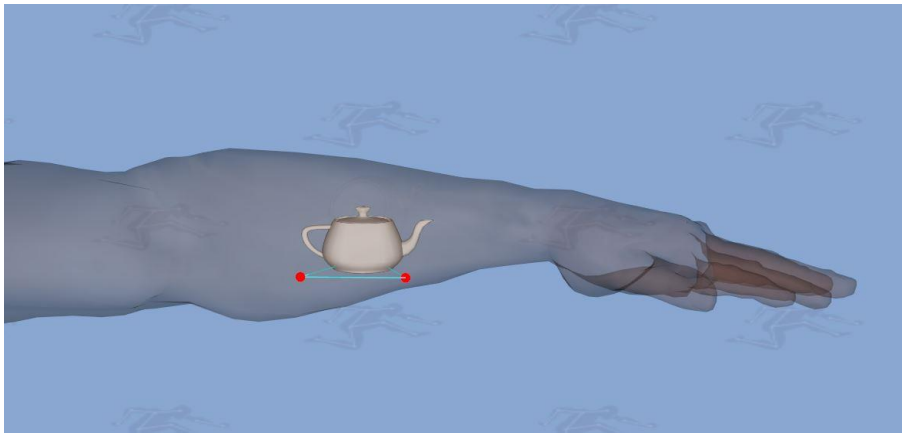

Figure 19: Test Case 2: A model is placed within the forearm of a DHM and has a skin-based parenting relationship established

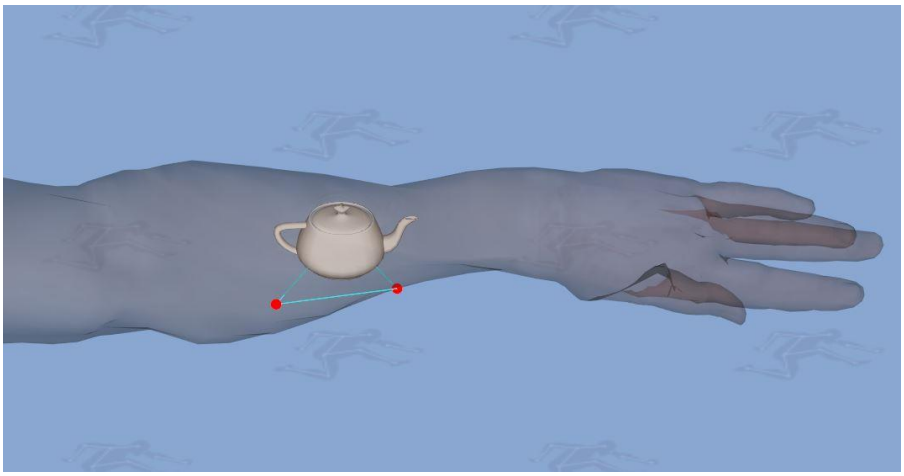

Figure 20: Test Case 2: The model rotates with the DHM's skin-mesh rather than a specific joint, thus maintaining a realistic motion

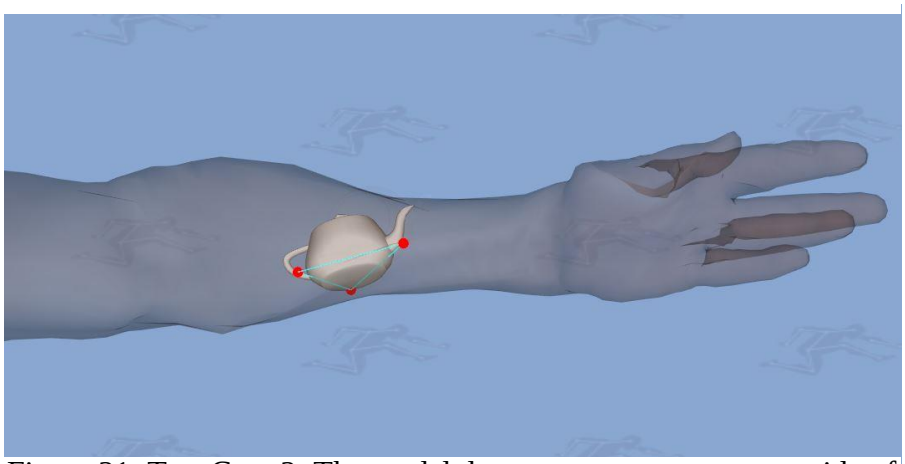

Figure 21: Test Case 2: The model does not rotate or move outside of the DHM's skin-mesh

A summary of comparisons between traditional joint-based parenting and the presented skin-based parenting method is displayed in Table 1. With skin-based parenting, knowledge of the underlying joint structure that governs the DHM's posture and the specific joints is not needed to establish positional and rotational relationships. The skin-based method does remove the capacity for a user to define specific relationships manually, but this is only needed if the desired motion is not achieved by the skin-based parenting algorithm. Additionally, the skin-based parenting was able to produce an intuitively expected motion of a rigid, internally placed, model in both Test Case 1 and Test Case 2. Test Case 1 presented a scenario where the designation of the reference joints needed in joint-based parenting would be exceedingly challenging, but establishing a skin-based parenting relationship was trivial. Test Case 2 presented a situation where a single joint-based parent is unable to correctly emulate the desired motion, but skin-based parenting was able to generate the expected motion.

| Parenting Method | Knowledge of underlying structure required | Manual designation of reference joints required | Precise control of reference joint weights possible | Expected motion emulated in Test Case 1 | Expected motion emulated in Test Case 2 |
|---|---|---|---|---|---|
| Joint-based | | | x | | |
| Skin-based | x | x | | x | x |

Table 1: Comparative features and performances of joint-based parenting and skin-based parenting

# CHAPTER 3

# MODEL DEFORMATION

This chapter presents an implementation of a free-form deformation (FFD) algorithm for deforming internal viscera models as the DHM progresses through a variety of postures. The implemented FFD algorithm relies on the positioning of control-points to drive the deformation. The overall shape and quality of the deformation is dictated heavily by the configuration of these control-points. Thus, several iterations of control-point configurations are presented and discussed. Then a method for coupling this FFD algorithm with the skin-based parenting algorithm presented in Chapter 2 is provided. The advantages of pairing these two systems is discussed. Additionally, the resulting deformable representations of internal viscera are compared against their static counterparts.

## 3.1 Introduction

In computer graphics and simulation, 3D models are composed of vertices and faces. These models are either procedurally generated through an algorithm or they are created through the use of 3D modeling software. In either case, the result is a collection of polygons which can be rendered in 3D space. Inherently, these models are static and the vertices do not change relative position. However, in nearly all 3D rendering and simulation environments, simple global transformations can be applied to these vertices. A global transformation is any transformation that does not change the relative position

of any vertex, but instead changes where each vertex is rendered in the world space. Typical global transformations are scale, translation, and rotation. In brief, scale changes the global distance between each vertex, translation changes the global position of each vertex, and rotation changes the global orientation of each vertex. With all of these transformations, the relative positions of the model's vertices do not change. Models that only support these global transformations will be referred to as static models.

When representing rigid bodies, objects that do not consider deformation, static models are generally sufficient for use in simulations. However, when the required models reflect non-rigid bodies, static models are insufficient representations. In order to represent non-rigid bodies, the models must support local transformations of their vertices. The limitation of static models is especially apparent when attempting to model internal viscera within a DHM. A DHM's skin-mesh deforms as it changes posture, and thus non-deformable models placed close to the skin-mesh are likely to extrude through the skin-mesh. Additionally, the static internal viscera models are likely to intersect and create undesirable spaces between one another.

One method for obtaining the appearance of a non-rigid body is to create many static models, each of which is altered slightly from the original model. These models are then rendered in rapid succession, thus giving the appearance that one model is deforming as time progresses. This method is tedious, as a variation of the base model must be created for each desired frame, and requires predetermined knowledge of the deformations that

will occur. Therefore, a preferred method is to provide an algorithm that can modify the local vertex positions of a 3D model in real-time based on some form of input.

A method for achieving interactive deformation of 3D models was proposed by Sederberg and Parry (Sederberg and Parry 1986). This method involves constructing a set of control points, in a parallelepiped lattice around and through a 3D model, and then transforming the model's vertices as these control points are repositioned. In Sederberg and Parry's implementation, the deformation is mathematically defined by tensor product trivariate Bernstein polynomials (Sederberg and Parry 1986). Although this method results in smooth deformation, the deformed vertices are not guaranteed to stay within the bounds of the control points. Additionally, this method is primarily limited to the creation and modification of geometric models. Due to the large number of control points required for precise deformation of the internal viscera, it is cumbersome to manually position all control points. Some work has been completed to lessen the effort required to manually arrange all control points (Hsu, Hughes and Kaufgamn 1992), however input from the user about the deformation process is still required.

In order to achieve precise deformation with minimal user input, a new method for automatically establishing a set of FFD control points around a specified model was created. Furthermore, these control points are then automatically parented to the DHM's skin-mesh through the skin-based parenting technique described in Section 2.3.2. This allows for the positions of the control points to be driven directly by the DHM's skin-mesh and results in a dynamic 3D model that deforms as the DHM changes in posture.

## 3.2 Approach: Overview

One of the primary differences between the FFD method implemented here and the method implemented by Sederberg and Parry is that trilinear interpolation is used in place of trivariate polynomial interpolation. The motivating factor for using linear interpolation is to constrain the mesh vertices within the hull formed by the control points. Keeping vertices within the control points is necessary, since it generally precludes results that violate skin and adjacent organ boundaries. By using the skin-based parenting method in combination with a linearly interpolated FFD algorithm, the likelihood of an internal viscera model protruding through the DHM's skin-mesh or the mesh of the other internal viscera models is greatly reduced.

The first step in setting up a deformable model is to establish a parallelepiped lattice of control points for the 3D model. This lattice will surround the 3D model, and each of its vertices will fall within one of the lattice's cells, as seen in Figure 22. Several methods for determining the initial configuration of this lattice will be discussed in Section 3.3.

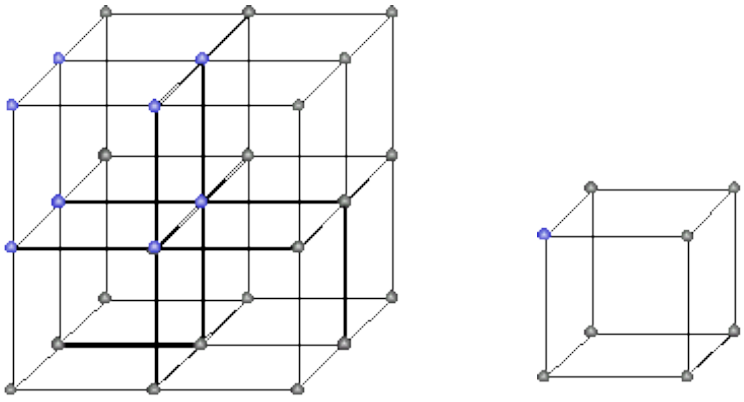

Figure 22: A parallelepiped lattice of control points on the left and an individual cell on the right

A single cell of the lattice can be used to explain how the manipulation of this control point lattice is used to deform the vertices in the whole model. A control point is capable of influencing the position of a vertex within the model if and only if it falls within a cell of which the control point is a member. Each cell has exactly eight control points and therefore each vertex is affected by exactly eight control points. When one or more of the eight control points governing a vertex move, as seen in Figure 23, the resulting movement of that vertex is calculated. To calculate the new position of the vertex, an offset $V_n$ between the initial $V_i$ and current $V_c$ position is calculated for each control point (n=0, 1, 2, … 7):

$$\boldsymbol{V_n = V_{nc} - V_{ni}}$$

Equation 10: Calculating the offset vector for each control point

Trilinear interpolation is then performed using the distances between each control point and the vertex as well as the offset vectors:

$$x_d = (x - x_0)/(x_1 - x_0)$$
$$y_d = (y - y_0)/(y_1 - y_0)$$

$$z_d = (z - z_0)/(z_1 - z_0)$$

Equation 11: Determining x, y, and z distances used in trilinear interpolation

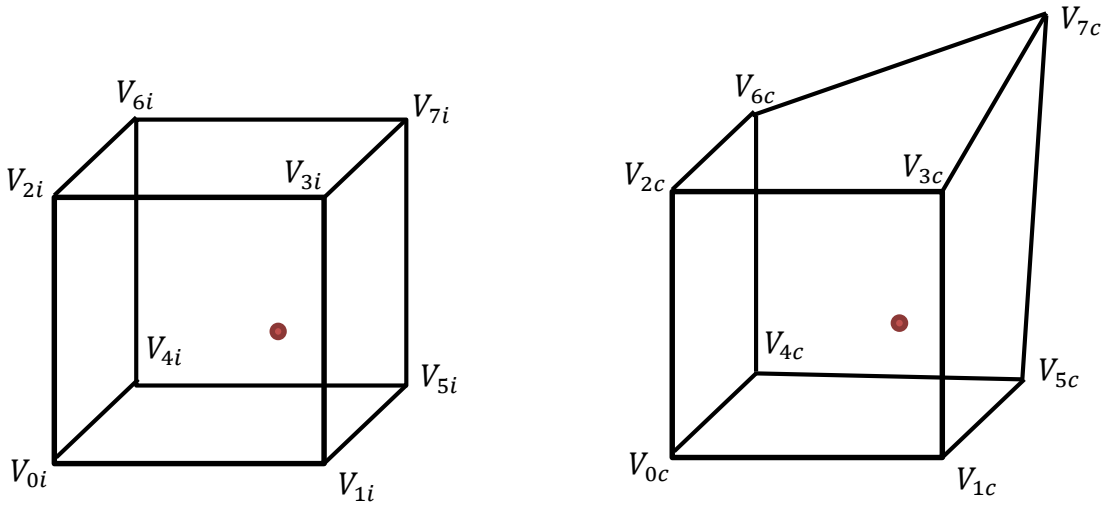


Figure 23: Initial control points shown on the left, and a modified control point is shown on the right

In Equation 11, $x_0$ corresponds to the lattice point below $x$ and $x_1$ represents the lattice point above $x$ and in the same manner for $y_0, y_1, z_0$, and $z_1$. First the offset vectors of each control point $\boldsymbol{V_0}, \boldsymbol{V_1}, \ldots, \boldsymbol{V_6}$ are interpolated along the $x$-axis:

$$c_{00} = \boldsymbol{V_0} * (1 - x_d) + \boldsymbol{V_1} * x_d$$

$$c_{01} = \boldsymbol{V_2} * (1 - x_d) + \boldsymbol{V_3} * x_d$$

$$c_{10} = \boldsymbol{V_4} * (1 - x_d) + \boldsymbol{V_5} * x_d$$

$$c_{11} = \boldsymbol{V_6} * (1 - x_d) + \boldsymbol{V_7} * x_d$$

Equation 12: Calculating interpolation values through $x$-axis

Then, values are interpolated along the y-axis:

$$c_0 = c_{00} * (1 - y_d) + c_{10} * y_d$$

$$c_1 = c_{01} * (1 - y_d) + c_{11} * y_d$$

Equation 13: Calculating interpolation values through $y$-axis

Finally, values are interpolated along the z-axis:

$$c = c_0 * (1 - z_d) + c_1 * z_d$$

Equation 14: Calculating interpolation value for $z$-axis

The resulting offset vector $c$ is added to the original vertex position to generate the current position of the vertex, depicted in Figure 24. This interpolation process is performed for each vertex within the cell. Each vertex is then repositioned, and the mesh is updated, thus modifying the vertices in the mesh in a non-global way, as seen in Figure 25. This method is repeated for every cell in the parallelepiped lattice. At the end of the deformation process, the total movement of control points is reflected in a series of local deformations within each cell which contribute to the overall appearance of the deformation. This more complex deformation can be seen in Figure 26.

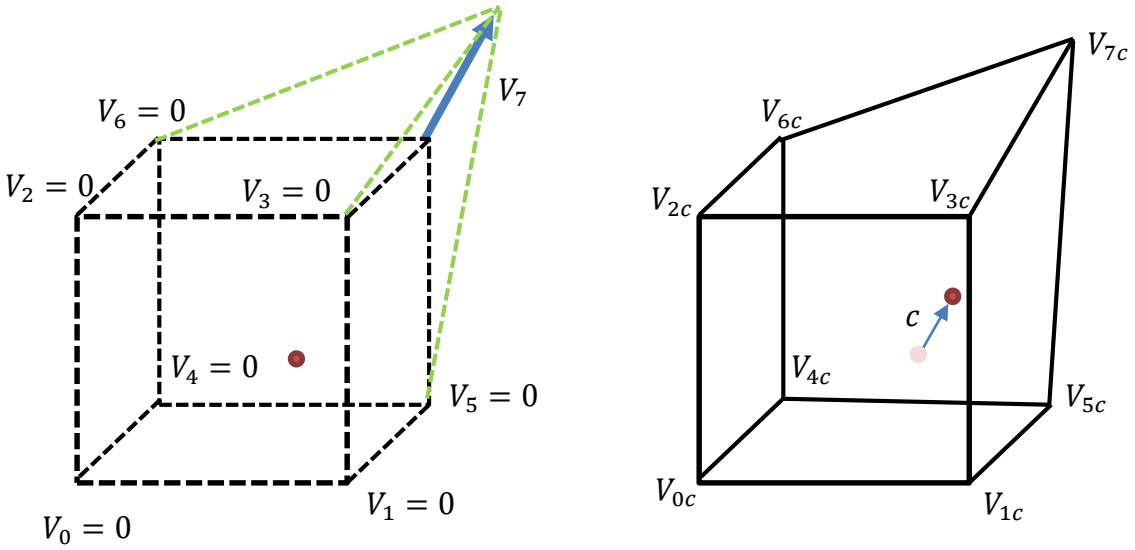


Figure 24: Offset vectors are calculated and the resulting interpolated vector $c$ is applied to the vertex position

## 3.3 Approach: Control Point Implementations

The initial configuration of the parallelepiped lattice of control points greatly impacts the quality of deformation that can be achieved. Too few control points will result in insufficient resolution of deformation, and with too many control points, the deformations will appear isolated. Additionally, the computational requirements of the algorithm increase with the number of control points and ideally must be kept reasonably low to ensure acceptable real-time performance. Next, several iterations of initial control point are considered in order to find a balance between quality of deformation and real-time performance.

### 3.3.1 Bounding-box

The minimal number of control points needed for the FFD algorithm is eight. This creates one cell around all vertices in the model's mesh as seen in Figure 25. This configuration of control points only allows for linear deformations of the model such as scale, shear, rotate, and translate. The limited deformation does not allow for models to bend or wrap. A bounding box configuration is ultimately insufficient in representing all internal viscera, as there are internal viscera models that need to bend as the DHM changes posture.

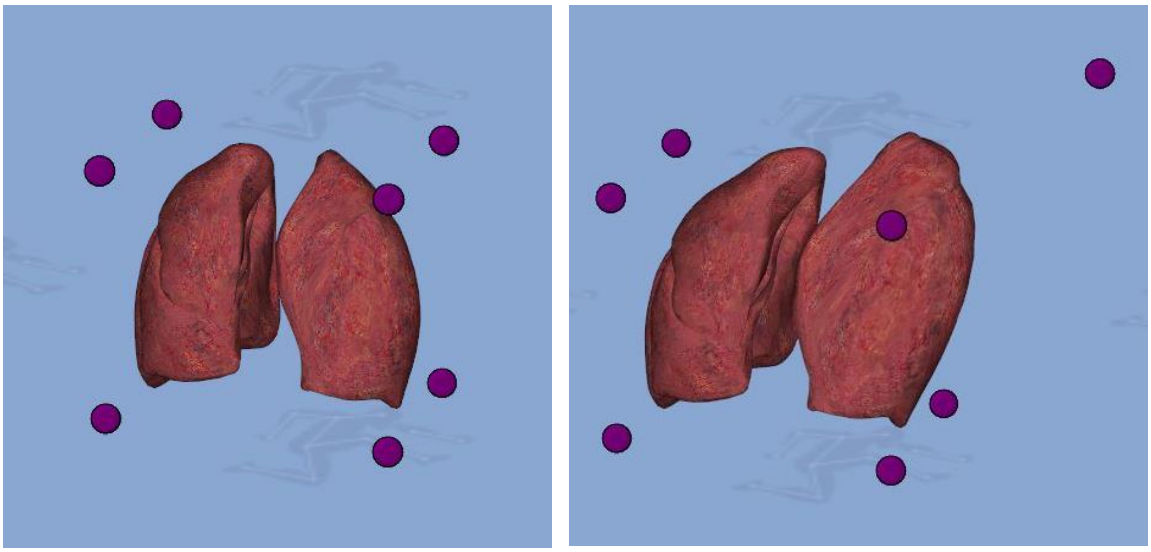

Figure 25: A single cell of the FFD algorithm being applied to a model of the lungs

### 3.3.2 Uniform Distribution

To increase the quality of deformations achievable by the FFD algorithm, additional control points are needed. These control points are distributed evenly through each dimension as seen in Figure 26. The number of control points in each dimension can be specified for each model by a user. With a large number of control points per dimension, a higher quality deformation can be achieved. This configuration results in many small cells distributed throughout the model. Each cell in the lattice performs a linear deformation on the vertices within itself. The composite motion of these cells working together provides the overall appearance of a non-linear, smooth deformation. The improvement caused by adding more control points can be seen in Figure 27. With the bounding-box configuration the organs cannot bend with the DHM and thus penetrate the DHM's skin-mesh. However, the increased control point configuration allows the internal viscera models to bend and not penetrate the DHM's skin-mesh.

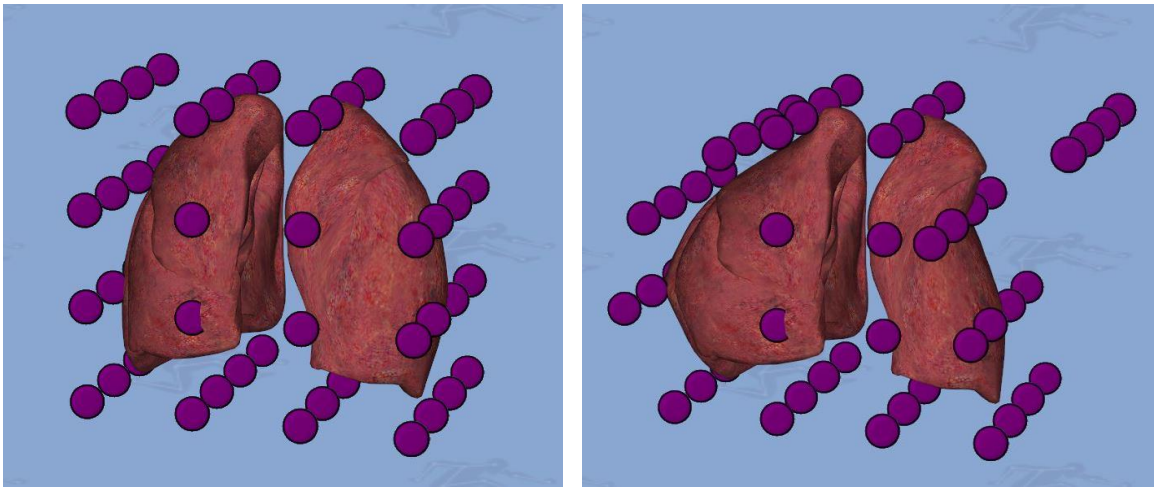

Figure 26: Multiple cells of the FFD algorithm working to create overall non-linear deformations

Adding more control points comes at a cost. Either the eight control point indices need to be stored for each vertex in the DHM's skin mesh, or they need to be determined each frame. As the number of vertices in the internal viscera models' meshes are not guaranteed to be a fixed number, the eight control points are determined for each vertex every frame. This results in reduced performance, in terms of frame rate, as the number of control points is increased. Because of the need to maintain acceptable performance, above 30 frames per second, additional control point configurations are considered.

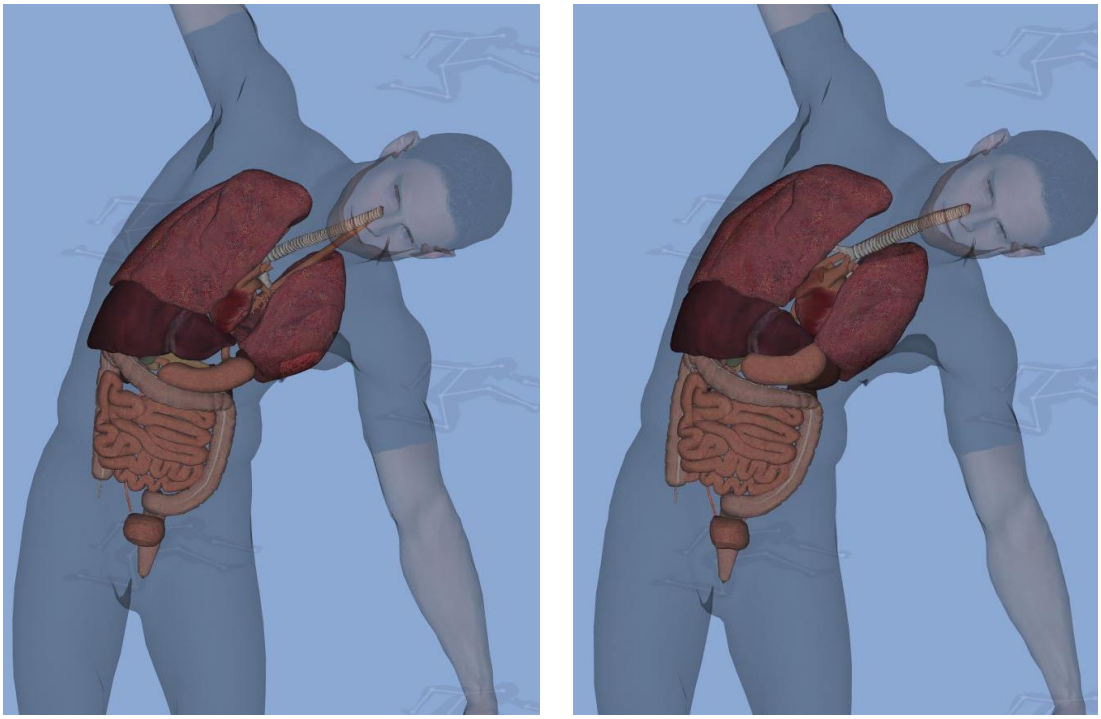

Figure 27: Deformable models using a bounding-box configurations, left, compared against models using user defined control points per dimension, right

### 3.3.3 Dimension Independent Distribution

Not all of the internal viscera models are the same shape. Furthermore, each model does not occupy the same amount of space in each dimension. This can be used to optimize the number of control points for an internal viscera model. For example, the small intestine is much thinner in the z-axis as compared to the x- and y-axis (Figure 28). The level of detail needed for deformation in the z-axis is significantly less than the level of detail needed in the x- and y-axis. Therefore, an additional configuration was created which allows the user to specify the number of control points per dimension. By allowing for the resolution of control points to vary not only by each model, but by each dimension in each model, a user is able to maintain high resolutions of deformation for a model where they are needed while minimizing the number of control points used.

### 3.3.4 Automatic Pruning

With the user possessing a high level of control over the distribution of control points through the model, there is the possibility for an excess of control points being created. An internal viscera model is not guaranteed to have vertices evenly distributed throughout its bounding box. Therefore, when a high number of control points are specified for a model, there can be cells created that have no mesh vertices within them. These cells essentially result in a waste of computation time. A method has been implemented which automatically identifies and eliminates control points of lattice cubes that contain no vertices within their volume. The method for pruning control points

consists of iterating through each control point and checking the space of each of the four cells to which this control point belongs. If there are no mesh vertices within any of this space, the control point is removed and not considered in any further calculations. This enables the user to be slightly more liberal in their assignment of control points per dimension, as excess control points are automatically pruned. The result of control point pruning can be seen in Figure 28.

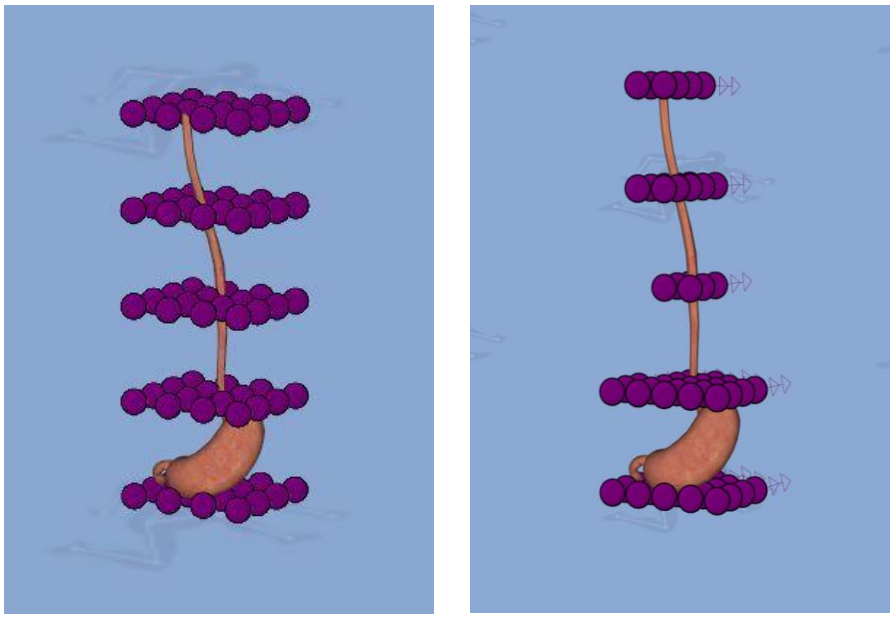

Figure 28: Deformable model with all user-specified control points on left, and pruned control points on right

## 3.4 Auto Control-Point Determination

The base FFD algorithm requires a method for manipulating the control points to deform the model. It is impractical to require manual manipulation, due to the large number of control points needed for high resolution deformations and the desire for minimal user input. In order for the internal viscera models to deform as the DHM moves through a

variety of postures, an automatic method for moving the control points is required. The skin-based parenting method can be coupled with the FFD algorithm to provide this control point motion.

Intuitively, the control points for a model should move in a manner that reflects the motion of the DHM's skin-mesh, and each skin-based parenting relationship (presented in Section 2.3.2) establishes a point which moves based on the position changes in the DHM's skin-mesh. Therefore, it is logical for each control point in the FFD algorithm to establish a relationship with the DHM's skin-mesh through the skin-based parenting method. Furthermore, the process for establishing a skin-based parenting relationship for each control point is trivial.

There are cases where some control points for an internal viscera model will not be positioned within the skin-mesh of the DHM. In these situations, the control point's initial position does not satisfy the constraint for skin-based parenting. Additionally, when two internal viscera models have vertices that are close together, it is logical for these vertices to move together in a manner that maintains this proximity. To accommodate these two conditions, skin-based parenting relationships are not established directly at the positions of each control point. Instead, a collection of positions within the DHM's skin-mesh is determined. These positions are evenly distributed throughout the interior of the skin-mesh (Figure 29). Once the collection of internal positions has been determined, the closest position within the collection is determined for each control point. After the closest internal position has been calculated, a skin-based parenting relationship

is established for that internal position. Finally, the offset of the control point from this internal position is stored.

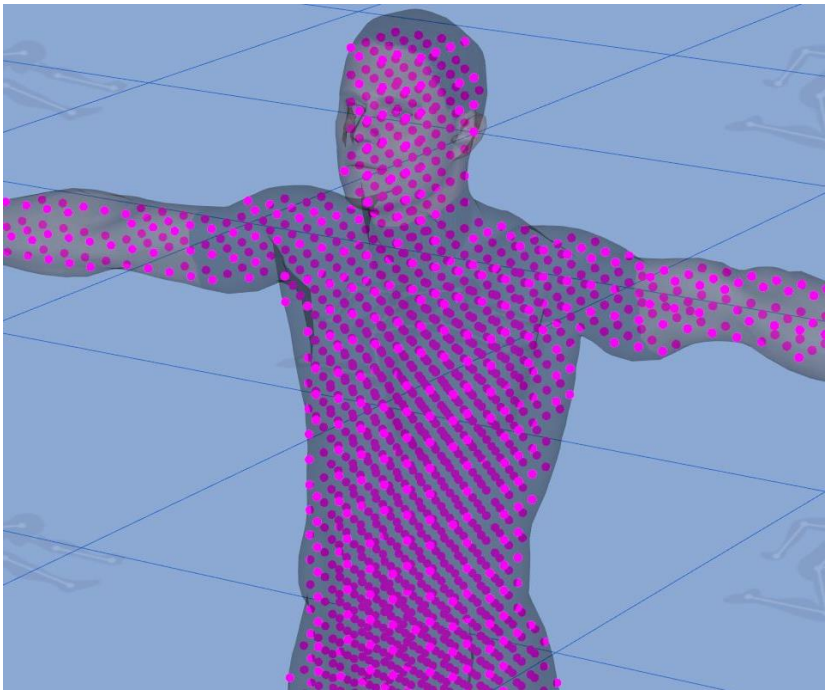

Figure 29: A collection of internal positions that could potentially be referenced by a control point

Each control point now has an associated skin-based parenting relationship defined. As the posture of the DHM is changed, its skin-mesh deforms which causes the skin-based parenting positions to move. The movement of these skin-based parenting positions is directly being reflected in the movement of the control points which govern the FFD algorithm and thus deforms the internal viscera meshes. The deformation of the internal viscera models is driven indirectly from the motion of the DHM's skin mesh.

## 3.5 Results

The FFD algorithm coupled with the skin-based parenting algorithm substantially reduced the manual input required to maintain the position and deformation of internal viscera models. One case where this reduction can be seen is when attempting to represent arteries and veins within the DHM. As seen in Figure 30, the main vein and artery models span much of the DHM. If static models are used to represent these internal viscera, a large amount of effort would be needed to segment these models into pieces so they could move with changes in posture. In some places, such as the elbow and shoulder, it would not be possible, due to the lack of control at points where two models should join, to represent a continuous model using static representations. When the static

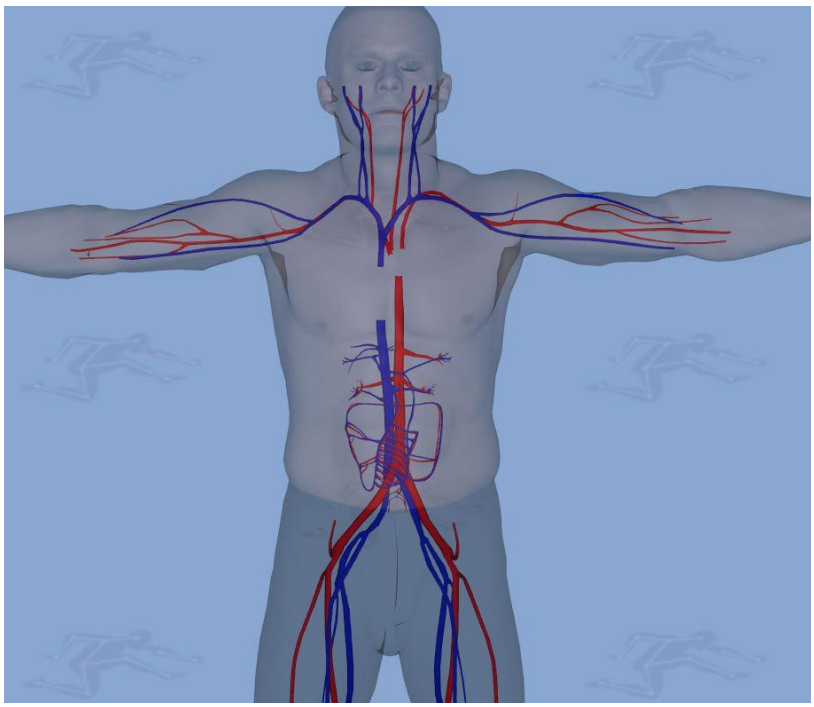

Figure 30: Vein and artery models are placed within a DHM in a default position

models are not manually segmented and used with traditional parenting methods, they penetrate the DHM's skin-mesh and behave unrealistically, as seen in Figure 31.

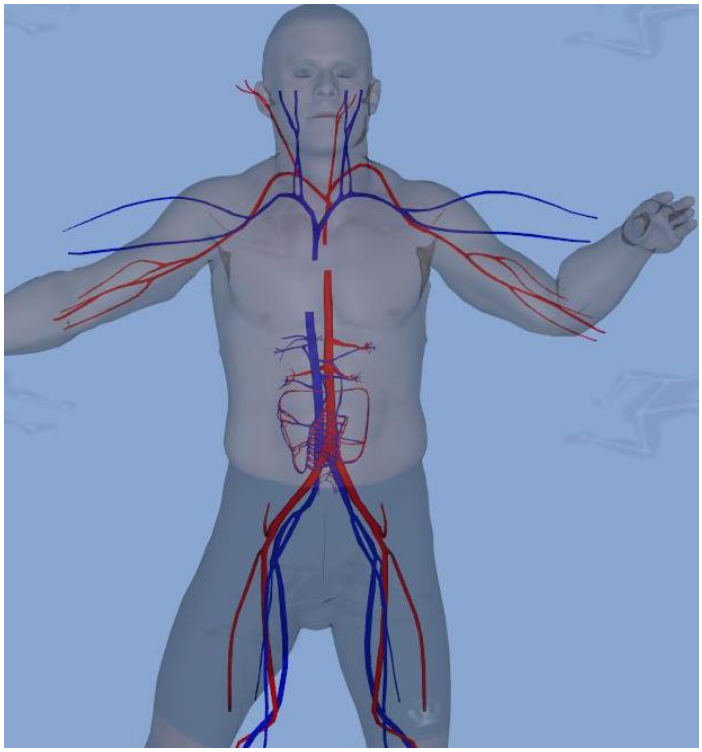

Figure 31: Static vein and artery models do not work well with changes in posture

However, using the deformation algorithm presented in the previous section, these models can be used in a variety of postures (Figure 32). Furthermore, the way the models are constructed is irrelevant, and no manual input is needed other than specifying the models' initial positions and orientations within the DHM. While the deformation of the veins and arteries has limited fidelity, the algorithm generates results for various postures without the need for a large amount of manual labor.

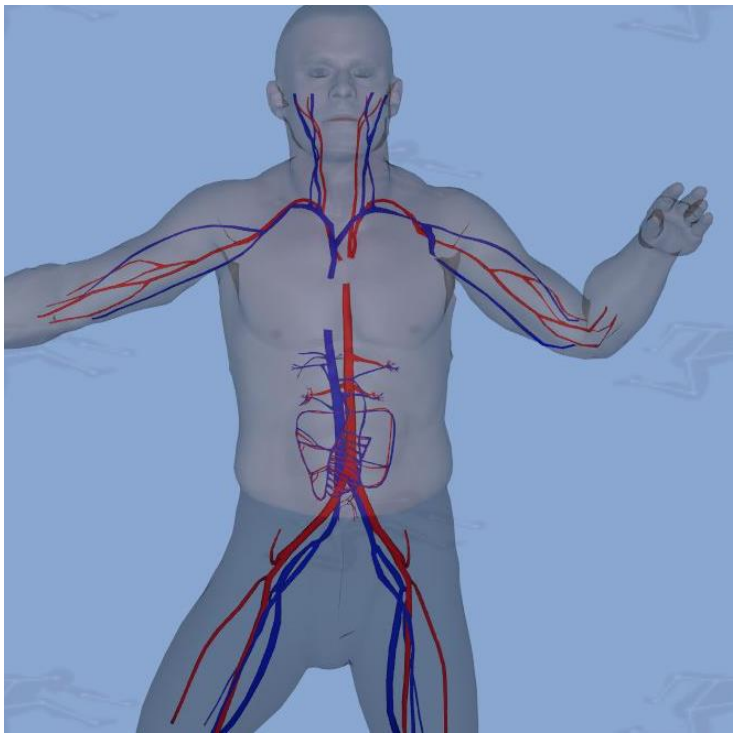

Figure 32: Deformable vein and artery models bend and move with changes in posture

The quality of the positions and deformations produced is difficult to validate. Current methods of image capture that pertain to internal viscera are not conducive for capturing internal viscera images while the subject is in motion. Some work has been done to recreate internal viscera motion through medical imaging data (Hostettler, et al. 2006), but this work generally pertains to the motion of internal viscera during the inspiration/respiration cycle. Validating the created motion and deformation of the internal viscera through the use of real-world experimentation is difficult but several obvious issues with the use of static models are solved through the proposed methods. Three of these issues are: internal viscera models intersecting with the DHM's skin-mesh, internal viscera models intersecting with each other, and internal viscera models creating

undesirable gaps between each other. These issues are illustrated through comparisons of implementations of static models (Figure 33, Figure 35, and Figure 37) with implementations of the presented FFD algorithm (Figure 34, Figure 36, and Figure 38).

In Figure 34, the DHM has simply twisted at the torso and the lung model has clearly penetrated through the skin-mesh. This is a common artifact of static models whenever they are placed close to the surface of the DHM's skin-mesh. The resolution of this error is shown in Figure 33, where the deformation algorithm has caused the lung models to twist with the DHM and not penetrate through the skin-mesh.

Figure 35 demonstrates an undesirable intersection between two internal viscera models. As the DHM again assumes a twisted posture, the lung model intersects the middle of the liver model. With the static representation of the lungs there is no way for it to maintain its seal with the liver model as the DHM changes posture. However, this is possible when using dynamic models as seen in Figure 36. Instead of intersecting the liver model, the lung model deforms to twist as the DHM twists.

Based on anecdotal information provided by medical experts as well as knowledge of the physical properties and attachment points for internal viscera within the chest cavity, it can be assumed there is generally not a significant amount of space between these internal viscera models. However, when using static models to represent these internal visceras a significant amount of space is created between the models as the DHM changes in posture (Figure 38). Due to the rigid behavoir of the models, inaccurate

separation occurs between the lungs and liver as well as between the liver and the large intestine.

Figure 37 shows this separation does not occur when dynamic models are used to represent the internal viscera models. As the DHM changes in posture, the models deform to maintain seals with the models that surround them. While the extact extent to which each organ will expand or contract to maintain these seals needs to be researched further, the general behavor of the dynamic models anecdotally results in a better representation.

While current medical imaging techniques and the limited scope of this research do not allow for thorough validation of the motions and deformations produced by this algorithm, three common errors that occur when using static models to represent internal viscera have been addressed and the overall motions and deformations are acceptable. Furthermore, it has been coupled with the skin-based parenting method to eliminate the need for manual input. The resulting system allows for a variety of internal viscera models to be integrated within a DHM in a simple and effective manner.

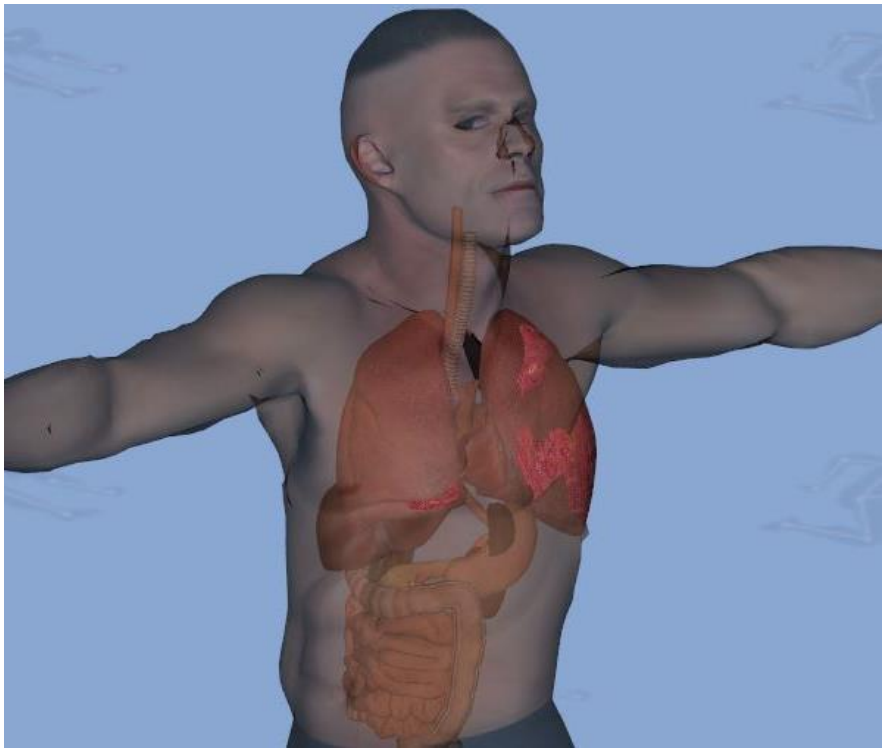

Figure 33: Static model of the lungs penetrates the DHM's skin-mesh in extreme posture

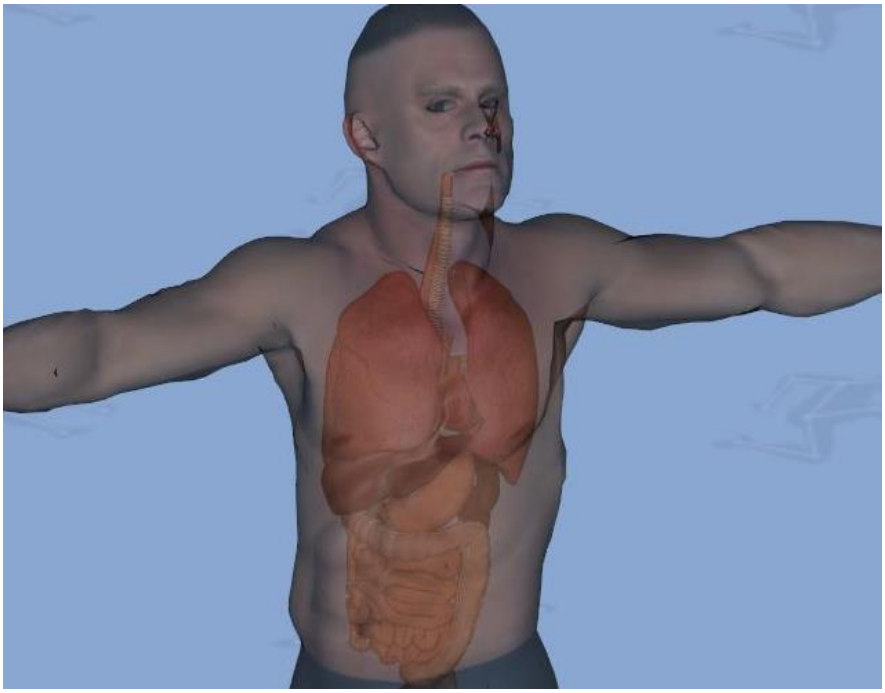

Figure 34: Dynamic model of the lungs deforms to stay within the DHM's skin-mesh in extreme posture

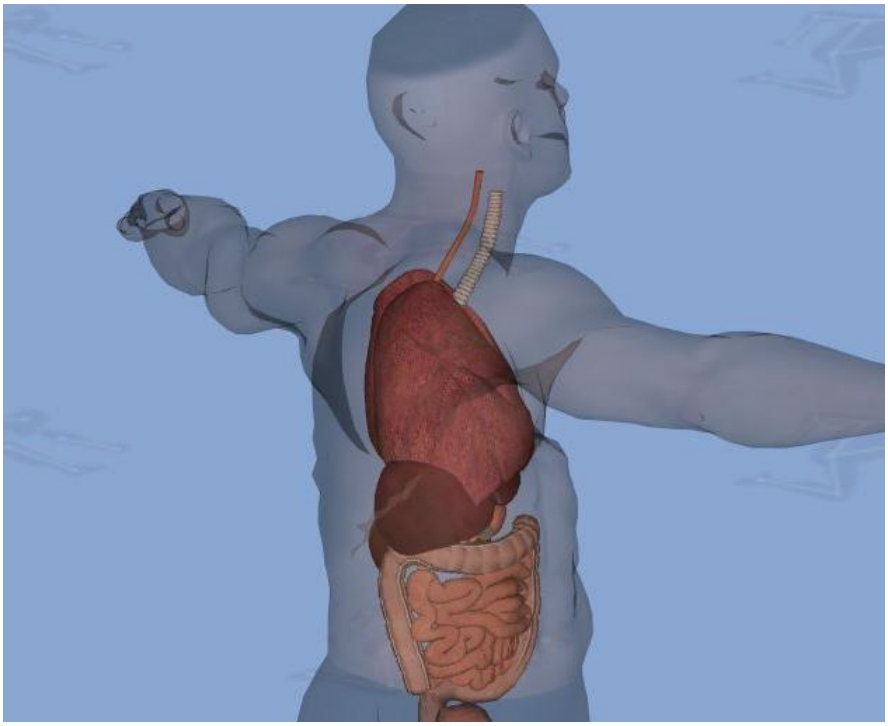

Figure 35: A static model of the lungs intersects the liver model as the DHM twists

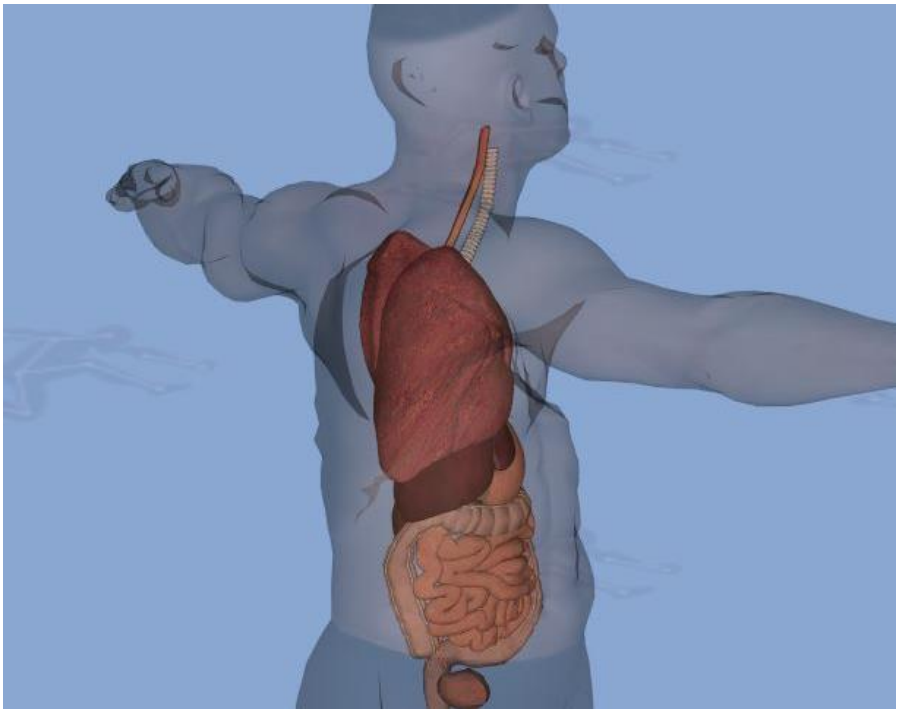

Figure 36: A Deformable model of the lungs deforms to twist around the liver as the DHM twists

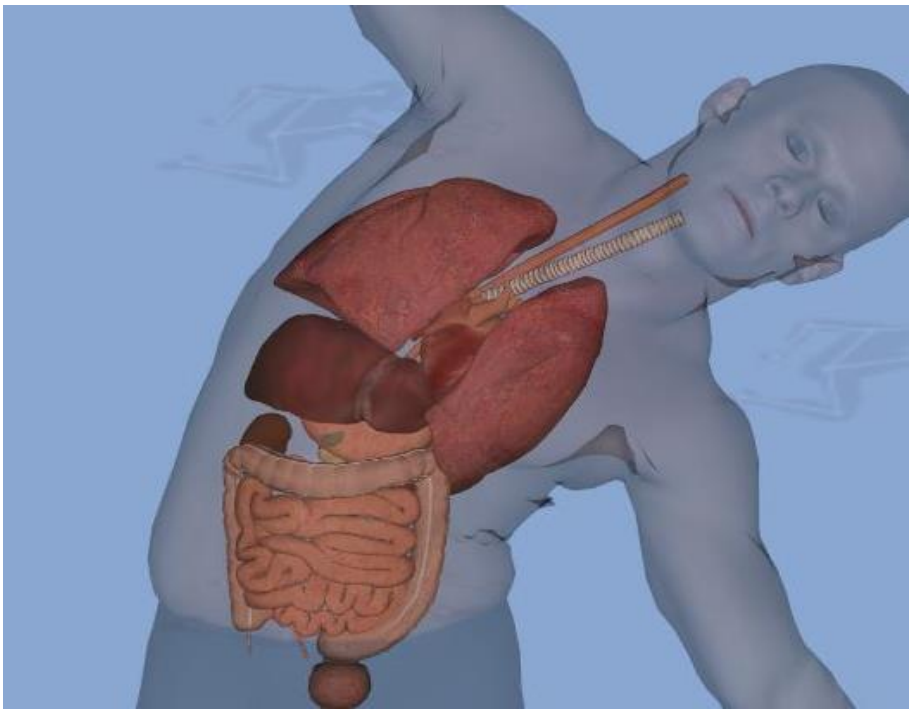

Figure 37: Separation occurs between static models of internal viscera

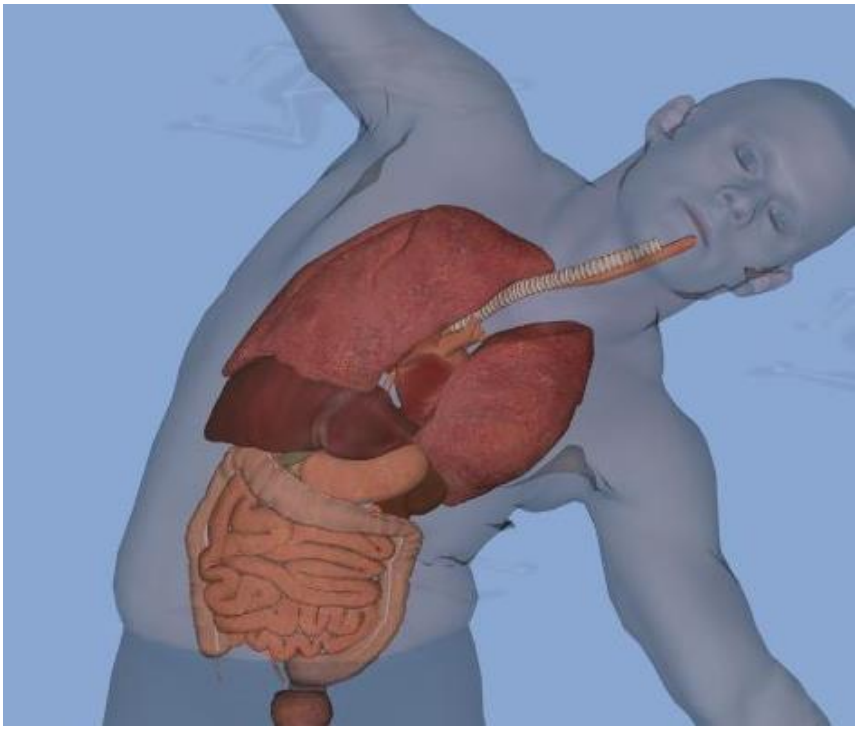

Figure 38: Dynamic internal viscera organs deform to maintain filled space

Examples of internal viscera models being represented in a DHM as the DHM progresses through a variety of postures, and the resulting performance of the internal viscera models being governed under the methods presented here (Figure 39, Figure 40. Figure 41). These images of mobility tasks showcase the reduction and/or elimination of the typical inaccurate artifacts brought out when simulating dynamic motions with static models of internal viscera.

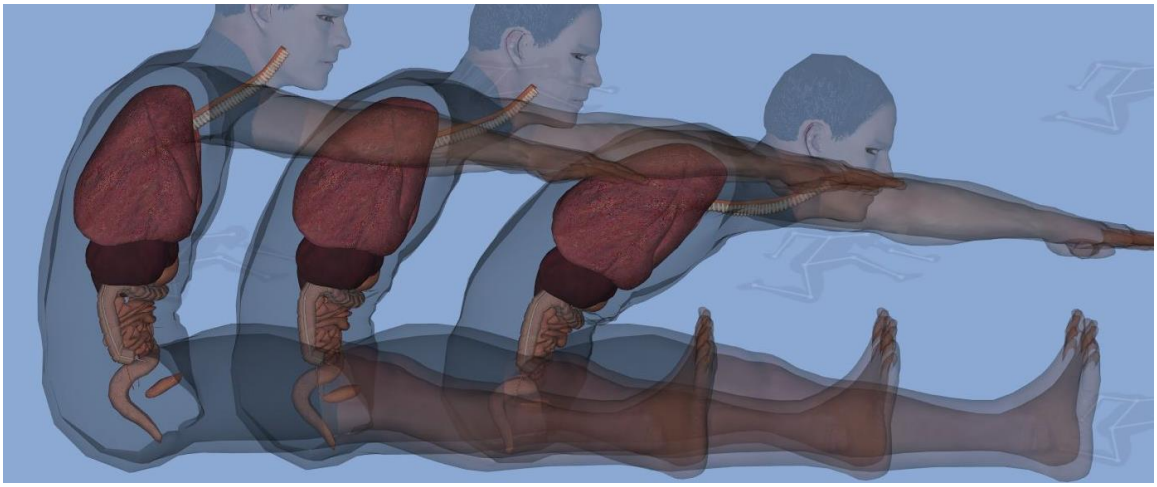

Figure 39: DHM with internal viscera performing a toe-touch

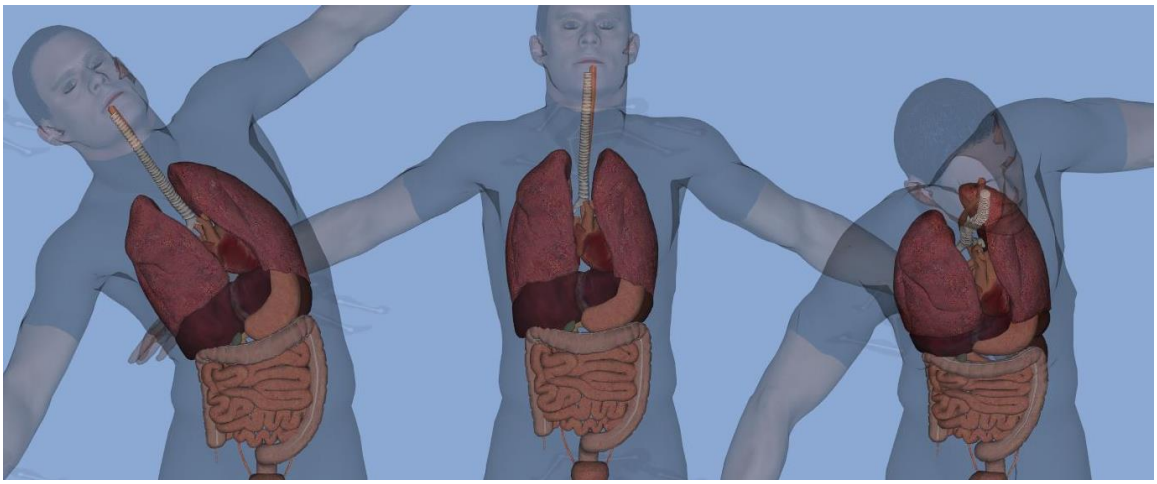

Figure 40: DHM with internal viscera performing a twist at the belly

Figure 41: DHM with internal viscera represented performing a lateral twist

# CHAPTER 4 POTENTIAL APPLICATION: SHOTLINE ANALYSIS OF PERSONAL PROTECTIVE EQUIPMENT

## 4.1 Introduction

One area of digital human modeling that requires accurate representations of internal viscera is in the design of personal protective equipment (PPE). With accurate internal viscera models a PPE system can be assessed to determine the overall protection it provides to the DHM. Traditional methods for assessing PPE coverage involve the use of manikins and manufactured prototypes. Furthermore, these manikins are generally constrained to a single posture, and multiple manikins are required to assess a variety of postures. These methods for assessing PPE designs are costly and time consuming. Digital modeling and simulation can enable PPE to be digitally assessed and iterated on through a variety of postures without the need to manufacture a prototype. Thus, PPE assessment is a desirable component in the task-based survivability platform.

In order to assess the coverage of a PPE system, a threat must be simulated. One method for simulating a threat is to perform shot line analysis (Yang, et al. 2009). 3D shot line analysis can be used to create coverage reports for a PPE system. Digitally modeled PPE pieces can be placed on a DHM while a series of randomized rays are drawn from a threat source towards the bounding area of a DHM as seen in Figure 42. It can be determined if a given ray intersects the DHM's skin-mesh, a piece of PPE, or both. Additionally, if the DHM is equipped with internal viscera models, it can be determined

if a given ray intersects any of these model, and if so, which model(s) it intersects. Statistical information regarding how many rays hit or missed the DHM or each of the internal viscera models can be utilized to assess the coverage of a PPE system. Furthermore, with dynamic viscera models, the PPE can be accurately assessed as the DHM progresses through a variety of postures. The remaining sections in this chapter will describe the methodology of shot line assessment and showcase a comparison of shot line assessment between static and deformable internal viscera models.

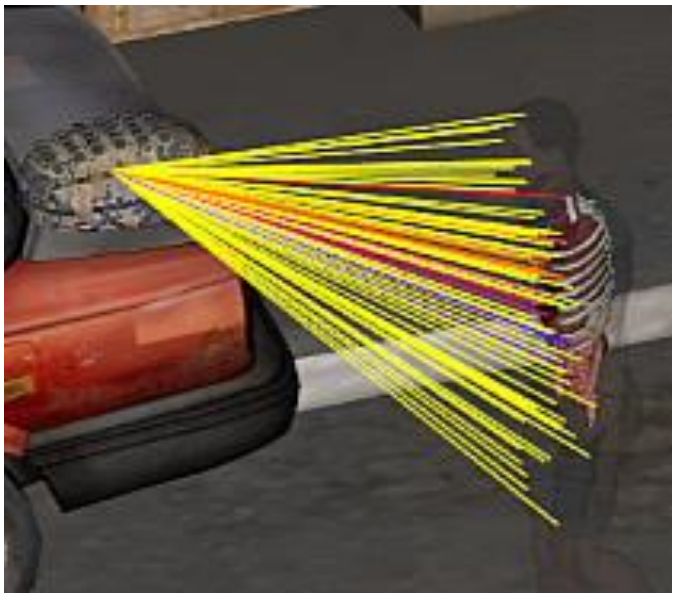

Figure 42: Shot line hit detection performed on DHM with internal viscera in city scene

## 4.2 Approach

The shot line hit detection process, as presented by Marler, et al (2016), begins with the specification of the source position, $v_s$. This point will serve as the origin for each ray used in the algorithm. The next step is to indicate the target DHM. Internal viscera

models are setup for the selected DHM using the methods presented in Chapter 2 and Chapter 3. Then, a user-defined number of rays, *n*, are generated. Each ray consisted of two positions within the 3D scene space. The first point for each ray is the source position. The second point is calculated for each of the *n* rays by determining a pseudo-random derivation $v_x$ of the vector between $v_s$ and $v_t$, the center of the DHM. This is shown in the following equation:

$$v_d = v_t - v_s$$

$$v_x = v_d * v_r$$

Equation 15: calculating a pseudo-random ray endpoint around the target DHM

where $v_r$ is a vector with each *x*, *y*, and *z* being a pseudo-random number between -10 and 10.

For each of the *n* rays generated, a *Virtools*® proprietary ray-intersection algorithm is applied to the DHM's skin-mesh, all PPE models, and all internal viscera models. If multiple models are intersected by a ray, the algorithm tracks the order in which models are intersected from source to target. This ordering can be used to determine whether or not a PPE piece was the initial point of contact for the shot line.

With each shot line calculated and analyzed to determine which, if any, of the objects in the scene it intersects, the acquired data is processed. Values that are calculated include: percentage of shot lines that intersect the DHM's skin-mesh, percentage of shot lines that do not intersect any model, percentage of shot lines that intersect PPE models, and

percentage of shot lines that intersect each of the internal viscera models. This values are then displayed to the user as seen in Figure 43.

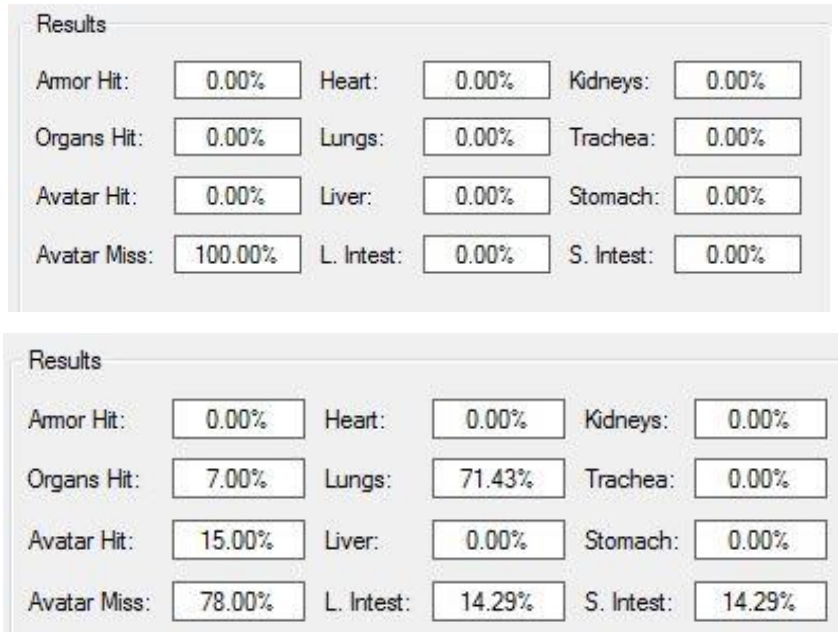


Figure 43: graphical user interface for shot line analysis before (top) and after (bottom) analysis

This method of shot line analysis can quickly be repeated for a variety of scenarios which include changes in the PPE system, DHM posture, and source position. Furthermore, these changes can be performed in real-time with both statistical values and 3D visualizations available to the user. Visualization of the shot lines relative to the DHM are rendered. The shot lines are colored based on the model they intersect and filters for which rays are visible can be toggled. The functionality of this program enables for quick superficial analysis of a PPE system with respect to the coverage it provides to a DHM and its internal viscera. The comparison of two PPE systems with this program can be seen in Figure 44 and Figure 45. It can be seen that more shot lines are intercepted by

PPE pieces in Figure 44 than in Figure 45 where red lines indicate a body hit and white lines indicate an armor hit.

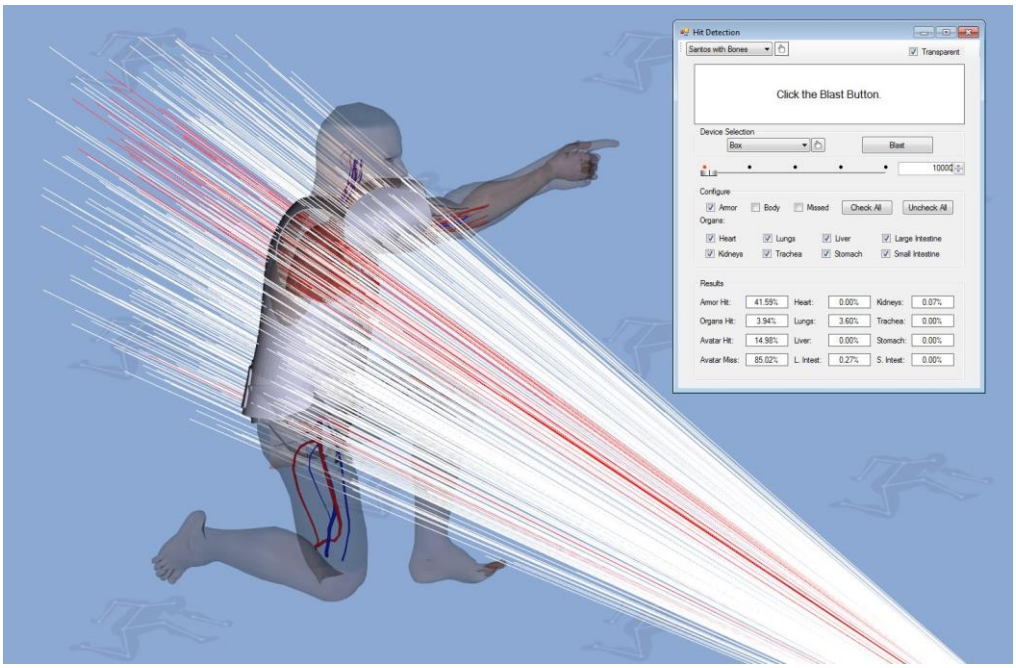


Figure 44: shot line analysis of a PPE system with full-torso coverage


Figure 45: shot line analysis of a PPE system with partial torso coverage

## 4.3 Results

Two instances of shot line analysis are presented to highlight the differences between the use of static and dynamic internal viscera models (Figure 47). Both instances contain the same DHM in the same position, with the same PPE system equipped (Figure 46). The sole difference between the two scenarios is that one uses static internal viscera models and the other uses dynamic internal viscera models. The percentage of hits on seven internal viscera models is determined for each shot line assessment. Due to the pseudo-random nature of the shot line creation, an average of five trials of shot line analysis is calculated for both static and dynamic scenarios. The results of shot line assessment with static models is presented in Table 2, and the results of shot line assessment with dynamic models is presented in Table 3.

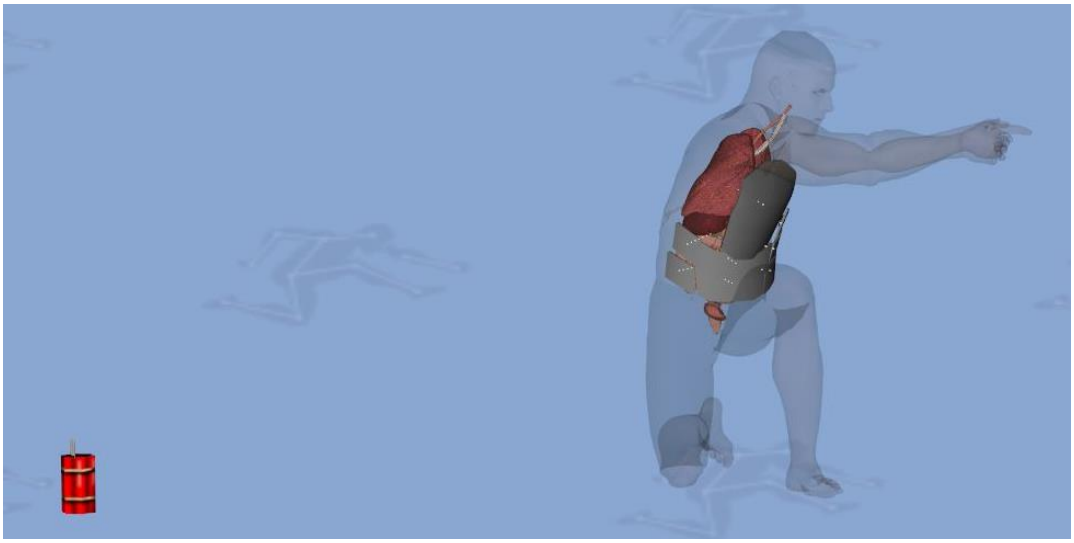

Figure 46: Shot line assessment scene with dynamic internal viscera models

| **Static Models** | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Body Hit | 3.85 | 3.84 | 3.85 | 3.83 | 3.82 |
| Body Miss | 96.15 | 96.16 | 96.15 | 96.17 | 96.18 |
| Armor | 11.03 | 11.05 | 10.73 | 10.99 | 10.5 |
| Organs | 10.57 | 11.01 | 10.92 | 10.86 | 11.27 |
| Heart | 0 | 0 | 0 | 0 | 0 |
| Lungs | 6.86 | 7.39 | 7.14 | 7.02 | 7.2 |
| Liver | 1.44 | 1.17 | 1.39 | 1.45 | 1.41 |
| Large Intestine | 1.35 | 1.37 | 1.36 | 1.37 | 1.58 |
| Kidneys | 0.06 | 0.02 | 0.06 | 0.04 | 0.07 |
| Trachea | 0.34 | 0.29 | 0.29 | 0.37 | 0.39 |
| Stomach | 0.47 | 0.63 | 0.56 | 0.51 | 0.52 |
| Small Intestines | 0.06 | 0.15 | 0.12 | 0.1 | 0.1 |

Table 2: Shot line results for five trials with static models

| **Dynamic Models** | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Body Hit | 3.84 | 3.87 | 3.82 | 3.84 | 3.87 |
| Body Miss | 96.16 | 96.13 | 96.18 | 96.16 | 96.13 |
| Armor | 11.12 | 10.42 | 10.61 | 10.57 | 11.1 |
| Organs | 10.87 | 10.93 | 10.98 | 11.44 | 10.55 |
| Heart | 0 | 0 | 0 | 0 | 0 |
| Lungs | 5.83 | 5.98 | 6.07 | 6.3 | 5.67 |
| Liver | 1.98 | 1.62 | 1.94 | 2.06 | 1.92 |
| Large Intestine | 1.69 | 1.65 | 1.72 | 1.66 | 1.63 |
| Kidneys | 0.29 | 0.19 | 0.2 | 0.18 | 0.24 |
| Trachea | 0.44 | 0.45 | 0.49 | 0.61 | 0.48 |
| Stomach | 0.35 | 0.36 | 0.3 | 0.34 | 0.37 |
| Small Intestines | 0.29 | 0.34 | 0.26 | 0.29 | 0.24 |

Table 3: Shot line results for five trials with dynamic models

| Static Models | Mean | STD |
|---|---|---|
| Body Hit | 3.838 | 0.0130384 |
| Body Miss | 96.162 | 0.0130384 |
| Armor | 10.86 | 0.23895606 |
| Organs | 10.926 | 0.25323902 |
| Heart | 0 | 0 |
| Lungs | 7.122 | 0.19829271 |
| Liver | 1.372 | 0.1154123 |
| Large Intestine | 1.406 | 0.09762172 |
| Kidneys | 0.05 | 0.02 |
| Trachea | 0.336 | 0.04560702 |
| Stomach | 0.538 | 0.06058052 |
| Small Intestines | 0.106 | 0.03286335 |

| Dynamic Models | Mean | STD |
|---|---|---|
| Body Hit | 3.848 | 0.02167948 |
| Body Miss | 96.152 | 0.02167948 |
| Armor | 10.764 | 0.32377461 |
| Organs | 10.954 | 0.31926478 |
| Heart | 0 | 0 |
| Lungs | 5.97 | 0.23906066 |
| Liver | 1.904 | 0.16757088 |
| Large Intestine | 1.67 | 0.03535534 |
| Kidneys | 0.22 | 0.04527693 |
| Trachea | 0.494 | 0.0680441 |
| Stomach | 0.344 | 0.02701851 |
| Small Intestines | 0.284 | 0.03781534 |

Table 4: Mean percentages and standard deviations of shot line percentages

This shot line comparison shows, as seen in the chart shown in Figure 48, that there is a significant difference in the resulting hit percentages between internal viscera models in this scenario between the use of static models and the use of dynamic models. The most notable of these differences is shown in the resulting values for the lung models. In the assessment with dynamic models the lungs, indicate a ~19% increase in hits over the scenario involving static models (Table 4).

A visual comparison of the two scenarios supplies an insight into a possible explanation for this discrepancy. In Figure 47, in the static case it can be seen that the lung model is inaccurately intersecting the liver model due to the static limitation of the models. Given the angle of the shot line source, this intersection results in more hits being recorded for the liver model. This is confirmed in the data with hit percentages for the liver in the static scenario being ~38% greater than the hit percentages for the liver in the dynamic

scenario (Table 4).

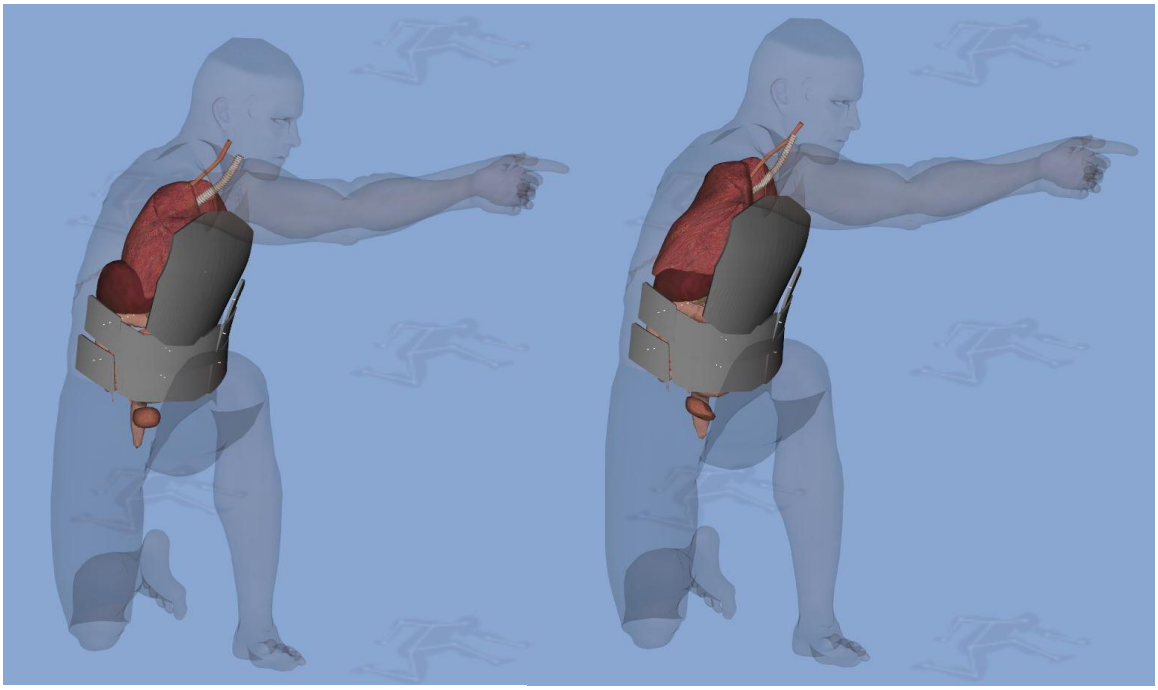

Figure 47: Static (left) versus dynamic (right) scenarios for shot line assessment

When coupled with the examples provided in Chapter 3 of their shortcomings, it can be seen that the use of static internal viscera representations is insufficient when attempting to leverage digital human modeling and simulation to assess PPE systems. This is especially true when attempting to assess the coverage provided by a PPE system when the DHM is progressing through a variety of postures.

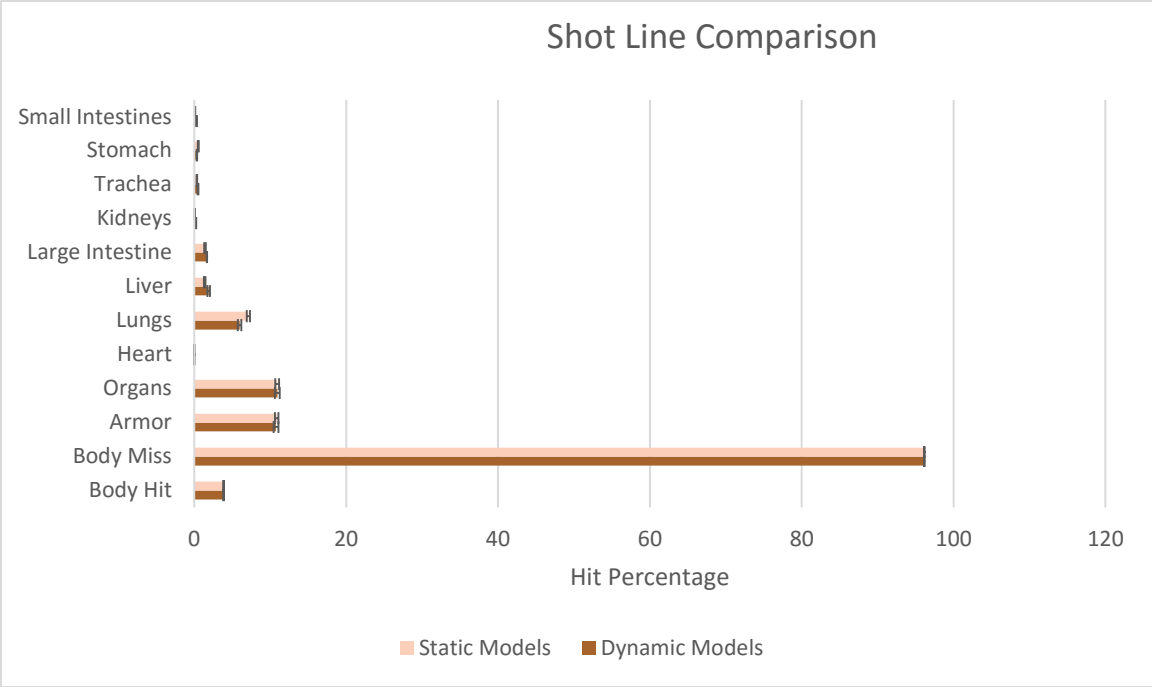


Figure 48: Chart comparing results of shot line assessment between static and dynamic internal viscera models

# CHAPTER 5 THREAT AND INJURY DATA

## 5.1 Introduction

Statistical threat assessment depicted in Chapter 4 is only one method for establishing a task-based survivability simulation. Another method is to take volumetric (3D) threat data and visualize it in a scene with a DHM. This threat data can be generated either through experimental measurements or simulation, and it can contain a variety of attributes, the common attributes being 3D positional coordinates. Additional attributes may take the form of pressure values for blast data, or binary values for data that depicts the path of fragments. Regardless of the form of the threat data, the underlying goal is to visualize the data relative to a DHM or other simulated model. Furthermore, the visualization must intuitively convey useful information to the user, and it must allow for real-time modification of the threat data. The real-time visualization of threat and injury data is another desirable component in the task-based survivability platform.

Given that threat data can come in many forms, it is appropriate to explore several methods of visualization. Three methods of data visualization are presented in this chapter along with their corresponding benefits and drawbacks. The methods, in order of discussion, are point cloud rendering, mesh coloring, and mesh generation. This list of visualization methods is not intended to be a comprehensive list of current visualization methods. However, the three methods presented are both widely used and suitable for the data considered for use in task-based survivability assessment.

## 5.2 Approach: Point Cloud

The core concept behind point cloud visualization is that traditional modeling methods, using polygonal meshes, are not well suited for rendering complex 3D shapes (Csuri, et al. 1979). The method utilizes a collection of primitive 2D shapes to represent a complex 3D volume. As volumetric data inherently comprises of attributes associated with x, y, and z coordinates, it is trivial to create a point cloud from such data. For each available data point, a sphere is rendered within the scene at the corresponding x, y, and z position.

The appearance of these spheres can be modified to convey the additional data attributes that correspond with this data point. Two appearance styles are used to convey additional information about the data points. The first appearance style involves changing the radius of the rendered sphere. The radius can reflect a variety of attributes where the magnitude of the attribute is useful. The second appearance style is the coloring of the rendered sphere. The color is used to visualize various ranges of values such as in Figure 49 where the spheres represent pressure values or Figure 50 where the spheres represent threat-level scores.

Point cloud rendering is a simple visualization method that requires minimal effort to show volumetric data. However, when attemping to render large numbers of data points the frame rate drops due to the large number of primitive shapes being rendered each frame. The reduction of frame rate ultimately inhibits the interactivity, and thus the

usefullness, of the simulation.

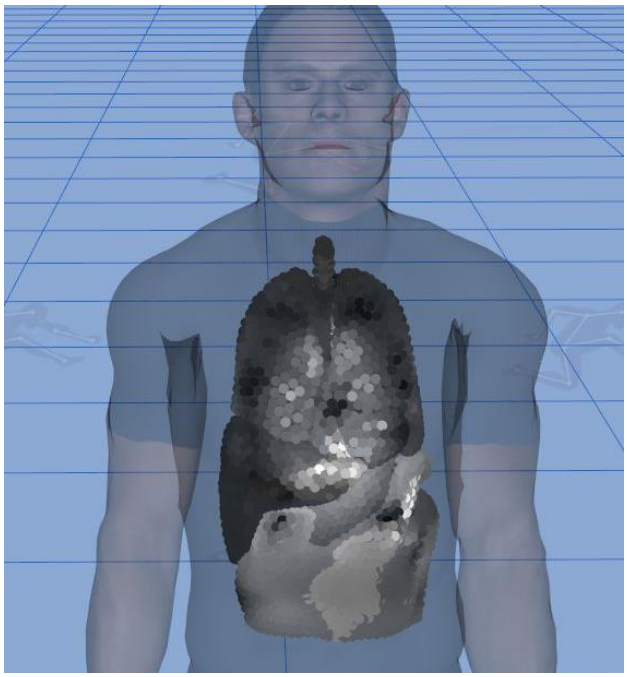

Figure 49: Point cloud rendering of pressure data

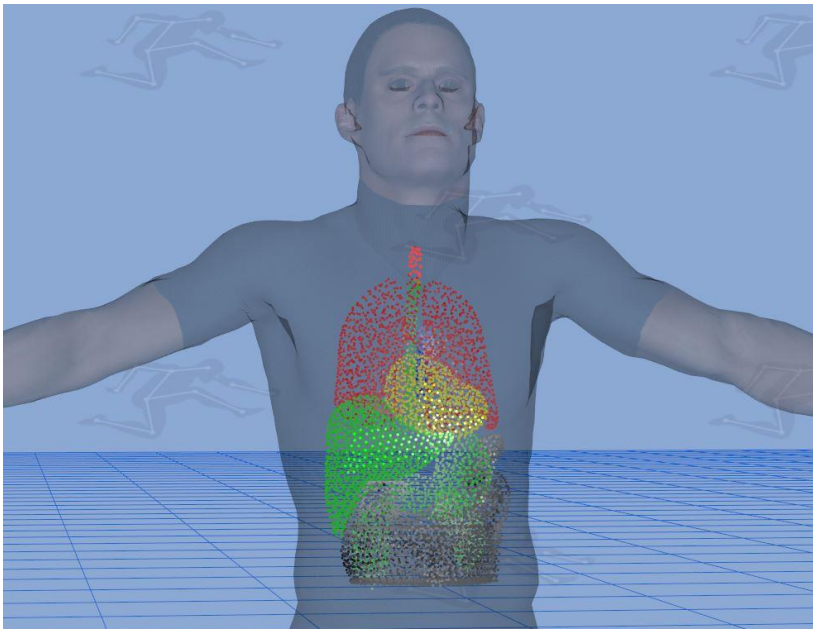

Figure 50: Point cloud of threat-score data

## 5.3 Approach: Mesh Coloring

A vertex shader is a common computer program, in the field of computer graphics, that is capable of changing the appearance of the geometry within a rendered scene. The program does this by processing each vertex in the scene and modifying its color based on values set in a 2D or 3D data structure, known as a texture. Given that a 3D texture parallels the structure of volumetric threat data, a 3D vertex shader is a suitable tool for visualizing threat data.

First, a 3D vertex shader program is constructed that processes each vertex in a scene and assigns the vertex a color based on its x, y, and z coordinates within the scene. The assigned color of the vertex will override the base color of the vertex. In order to use a 3D vertex shader to visualize threat data relative to a DHM, the threat data is converted into a 3D texture, which is simply a 3D array of integers. The 3D position of the data point is converted into x, y, and z indices in the array and the threat attribute is discretized into an integer. Once the data is converted into a 3D texture, the desired models are loaded into a scene. A scene consisting of a DHM and a vehicle with their default appearances is shown in Figure 51. The same scene is shown in Figure 52 with the 3D vertex shader actively visualizing a volumetric dataset.

The 3D texture used by the shader is able to be modified at runtime, thus enabling various data values to be shown over time. Furthermore, the positional mapping from the 3D texture into scene coordinates can be modified at run-time to move the data around

within the scene. These qualities allow for a high level of interactivity when visualizing volumetric data. Furthermore, due to the efficiencies of shader programs, mesh coloring does not suffer from the reduction in frame rate that is seen in point cloud rendering when dealing with large number of data points.

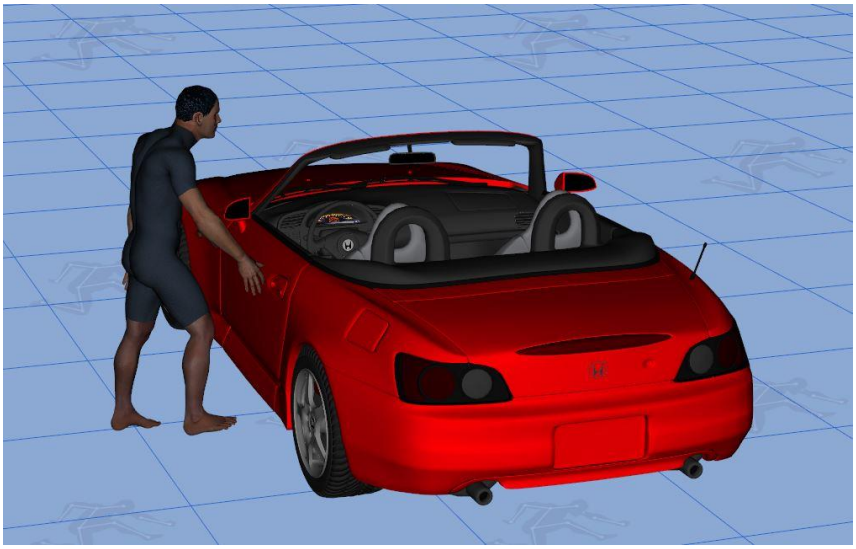

Figure 51: A scene of models with their default appearances

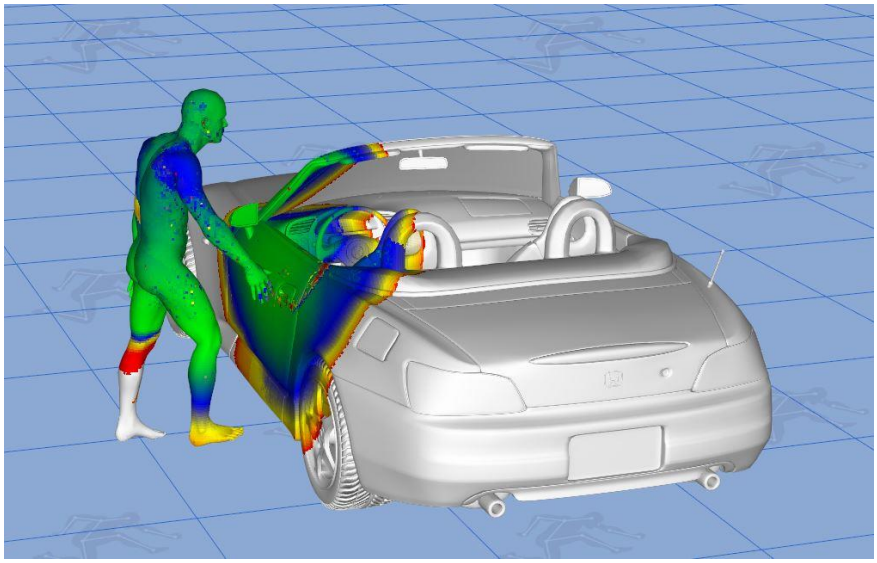

Figure 52: A scene of models with a 3D vertex shader visualizing volumetric data

Mesh coloring does suffer a drawback due to is inability to display data when there is no existing geometry within the scene. Since the shader is only able to show the data by modifying the appearances of models within the scene, data that does not overlap any models is not visible to the user. One common method for addressing this limitation is to place movable planes, essentially a thin cube, within the scene and allow the user to move these plane through the volumetric data-space. The progression of the thin geometry through the data enables the user to mentally visualize the entirety of the dataset through the viewing slices of the data in quick succession. While this method requires the user to mentally reconstruct the volume, it is commonly used in the viewing of medical imaging data such as magnetic resonance imaging (MRI) and computed tomography (CT) data.

## 5.4 Approach: Mesh Generation

One common method for visualizing 3D discretized data is to reconstruct a surface from the data. The concept of surface reconstruction, at a high level, takes 3D data points and creates a 3D model comprised of polygonal surfaces. These surfaces enable a user to visualize a solid 3D version of the data. This proves to be useful when attempting to visualize the overall shape or path of threat data. In addition to being useful in its own right, mesh generation is capable of providing geometry for mesh coloring. By enabling the automatic creation of geometry from data that then uses a mesh coloring shader, the requirement for preexisting models is removed from the mesh coloring visualization technique.

Substantial work has been put into surface reconstruction and there are many techniques with varying specific intents (Berger, et al. 2014). As this work targeted several visualization techniques in a low-depth manner, one of the most well-known and industry used methods was chosen, the marching cubes algorithm. The marching cubes algorithm, as presented by Lorensen and Cline (Lorensen and Cline 1987), is a surface reconstruction method that uses a collection of 15 predefined mesh configurations to reconstruct a surface from discretized 3D data.

Figure 53 shows a dataset visualized by point cloud rendering on the right and a corresponding surface reconstruction created through the marching cubes algorithm on the left. Sharp contours are visible in the surface reconstruction. These contours are due to the scale and resolution of the data provided to the marching cube algorithm. Several methods can be implemented to reduce the sharpness in the surface reconstruction. One method is to process the surface reconstruction with a smoothing algorithm. A method for smoothing without shrinkage has been proposed (Taubin 1995) and is widely used. The results of smoothing the surface geometry is shown in Figure 54.

Using surface reconstruction to visualize threat data is useful when attempting to understand the overall shape or path of the data. By itself it is unable to depict attributes

associated with the data positions. However, positional attributes can be displayed if mesh generation is combined with mesh coloring.

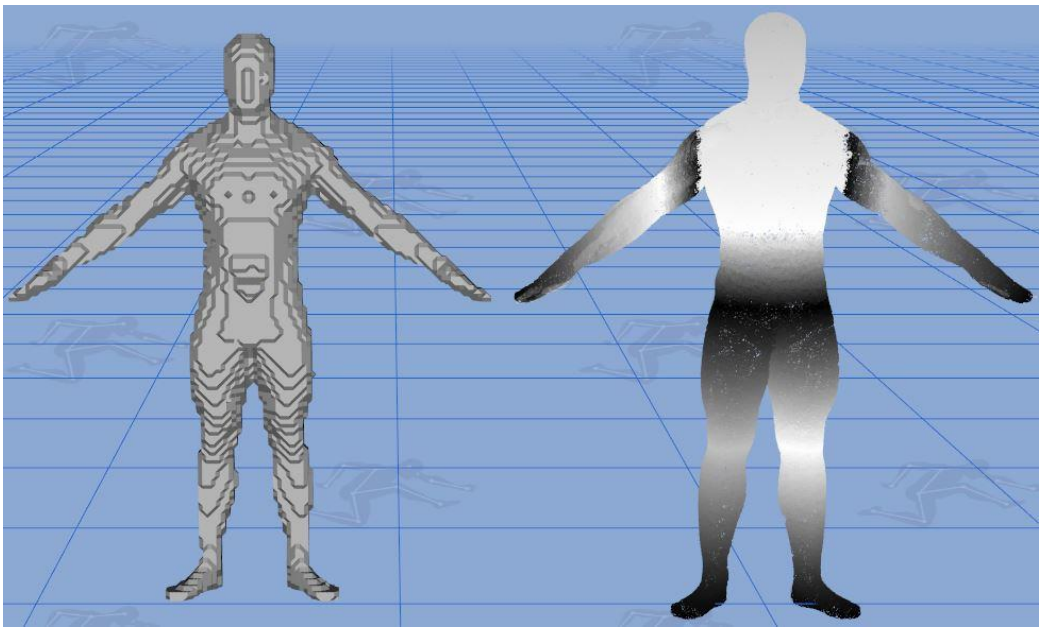

Figure 53: Marching cubes surface reconstruction (left) of point cloud data (right)

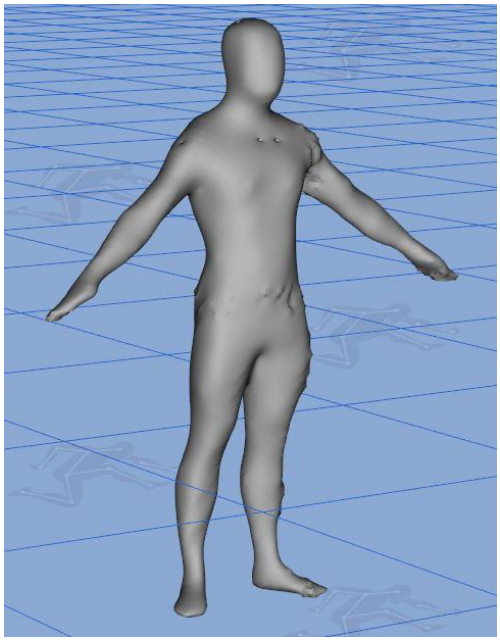

Figure 54: Smoothed surface reconstruction geometry

## 5.5 Discussion

The three visualization methods presented each have benefits and drawbacks. Point cloud rendering is by far the most easily implemented method, both in data preparation and in algorithm design. However, it suffers in performance as the number of rendered data points increases to large quantities. Mesh coloring is not afflicted by performance issues when encountering large numbers of data points. Additionally, it easily shows a large volume of data on existing models in a scene and allows for several methods of interaction such as real-time modification of the data values and their positional mappings. A drawback to mesh coloring is that it is unable to visualize data without access to preexisting model geometry. Mesh generation in the form of surface reconstruction is able to create geometry that represents the threat data. This is useful when the visualization goal is to understand the overall shape or path of the threat, but is unhelpful when there are relevant attributes association with the positional data. However, mesh generation can be used in combination with mesh coloring to provide both shape and attribute visualization of threat data.

# CHAPTER 6 CONCLUSION

## 6.1 Summary

The methods, algorithms, and techniques presented in this work attempt first to fill the gap that is present in the current state of the art for representing internal viscera within DHMs and second to help build a platform for task-based survivability assessment. Two primary objectives have been completed in order to achieve this. The first effort was to establish and maintain a spatial relationship between the internal viscera models and the DHM's skin-mesh. To fulfill this objective, a novel skin-based parenting method has been presented. The skin-based parenting method directly leveraged data from the DHM's skin-mesh in the form of vertex positions of polygonal faces. The method relies on establishing position and orientation relationships within the DHM's skin-mesh in a default posture. Once these relationships are defined, the desired internal position and orientation can be calculated from reference polygonal faces as the DHM's skin-mesh moves and deforms due to changes in posture and position.

The second effort to fill the described gap was to replace traditional static representations of internal viscera models with dynamic representations. The goal of using dynamic models was to reduce or eliminate several of the issues that arise when using static models to represent inherently dynamic objects. Three specific objectives were to minimize intersection with the DHM's skin-mesh, minimize intersection with other internal viscera models, and eliminate undesirable separation between internal viscera

models. Implementation of an FFD algorithm was presented to allow for real-time deformation of the internal viscera models. This algorithm allowed for user input to control the resolution of this deformation on a per model basis. Furthermore, the manual requirements generally associated with the FFD algorithm were removed through the coupling of this algorithm with the skin-based parenting method. Thus a system for creating dynamic representations of internal viscera within a DHM was established which only required an initial placement and orientation of the model within the DHM.

In addition to the primary objectives, two secondary components have been completed in order begin to construct a platform for task-based survivability assessment. First, a method for shot line assessment of PPE systems has been presented. Second, an investigation into methods for the visualization of threat and injury data has been presented.

## 6.2 Discussion

This work has documented the construction of a modeling and simulation platform for task-based survivability assessment. In doing so, it has also shown progress in furthering efforts to accurately represent internal viscera within a DHM. The initial hypothesis presenting in Chapter 1 was that increased accuracy in the modeling of internal viscera through the completion of the primary objective mentioned will increase the overall value and effectiveness of task-based survivability analysis. An experiment was presented in Chapter 4 which gave strong support to the notion that the dynamic internal viscera

models created in this work are better suited for use in survivability assessment than traditional static models. The methods for validating this claim are admittedly subjective. This is due to the nature of the problems this work aimed to solve. When considering positional relationships between rigid bodies with rigid connections, it is trivial to validate a method for simulating those relationships. However, when a spatial relationship is to be imposed between deformable models, such as the skin-mesh of a DHM and internal viscera models, the standard through which a comparison can be made is less clear.

Medical imaging data is not available that captures the exact motions of internal viscera as a subject moves through a variety of postures. Likewise, the exact deformations of internal viscera cannot be captured for use in validating the deformations produced by the proposed techniques. In light of these limitations for validation, the effectiveness of the proposed work has been demonstrated in two ways. The first is the reduction of manual user input for the creation and preservation of accurate representations of internal viscera through changes in DHM posture. The second is the removal of undesirable tendencies when attempting to represent internal viscera with current techniques.

For reduction in manual user input the skin-based parenting system eliminated the necessary step of manually choosing a parent reference which is present in the traditional parenting techniques. The coupling of the FFD algorithm with the skin-based parenting method removed the manual manipulation of control points typically required to perform a free-form deformation. The resulting system essentially parallels the “plug and play”

mentality found in the hardware field. Specifically, a 3D model consisting of polygonal faces can be placed within a DHM in the desired position and orientation and from that point on the system will automatically govern the position, orientation, and shape of the model as the DHM changes in posture.

Cases where internal viscera models, governed by traditional parenting techniques, within a DHM have moved in undesirable ways with changes in posture have been demonstrated. Whereas these same cases while using the skin-based parenting method have resulted in motions that are more representative of how an internal object would move. For representing internal viscera, three specific types of undesirable behaviors were identified when using static models. Each of these undesirable behaviors were shown to be lessened or eliminated when using the implemented FFD algorithm.

As for the general objective of constructing a platform for task-based survivability, the two components presented, shot line assessment and threat and injury data visualization, were shown as potentially effective tools. This work showcased the attainability of creating individual tools that offer use in a task-based survivability platform.

## 6.3 Future Work

The work presented does not fully fill the gap present in techniques for representing internal viscera within DHMs nor does it offer a comprehensive platform for task-based survivability assessment. Substantial improvements could be made to this work with the

availability of medical imaging data showing the motion and deformation of internal viscera as a subject changes posture. This data would not only allow further validation of this work, but it would allow for heuristic models to be created for each of the internal viscera.

Another area that could be expanded upon is the method through which the models are deformed. The FFD technique is generally used for the manual creation and deformation of models by an artist. Thus, there are no real physical properties pertaining to the models being deformed or the deformations themselves. The implementation of physics-based constraints for the internal viscera models would enhance the accuracy and realism achieved in the representations. The methods presented have no governance over the total volume of each internal viscera model. The addition of limits on the type and extent of deformation achievable by a specific model could be enforced based on the physical properties experimentally determined for the internal viscera object being represented.

In addition to limiting the achievable deformations, physical properties could be added to define interactions within the deformation algorithm. Physical interactions between control points could be modeled. However, these modifications soon begin to enter the realm of finite element analysis, which generally tends towards a higher processing cost and a loss of real-time interaction.

Much work can be done to build upon the methods and implementations presented in this paper. Such work can further enhance the realism and accuracy of representing internal

viscera within DHMs. This would expand the scope of human simulation and increase the usefulness of DHMs in any field that has a vested interest in assessing interactions between a humans and their environment.

# REFERENCES

Barr, Alan H. 1984. "Global and Local Deformations of Solid Primitives." *SIGGRAPH* 21-30.

Berger, Matthew, Andrea Tagliasacchi, Lee Seversky, Pierre Alliez, Joshua Levine, Andrei Sharf, and Claudio Silva. 2014. "State of the Art in Surface Reconstruction from Point Clouds." *Eurographics 2014 - State of the Art Reports.* Strasbourg, France: EUROGRAPHICS. 161-185.

Christy, M. 1980. *Mathematical Phantoms Representing Children of Various Ages for Use in Estimates of Internal Dose.* Oak Ridge, TN: Rep. ORNL/NUREG/TM-367.

Coquillart, Sabine. 1990. "Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling." *SIGGRAPH* 187-196.

Cowan, N. R. 1964. "The Heart-Lung Coefficient and the Transverse Diameter of the Heart." *British Heart Journal* 116-120.

Csuri, C., R. Hackathorn, R. Parent, W. Carlson, and M. Howard. 1979. "Towards an Interactive High Visual." *Proceedings of the 6th annual conference on Computer gaphics and interactive techniques.* New York, NY: ACM. 289-299.

D'Oronzio, U., O. Senn, P. Biaggi, C. Gruner, R. Jenni, F. Tanner, and M Greutmann. 2012. "Right Heart Assessment by Echocardiography: Gender and Body Size Matters." *Journal of the American Society of Exchocardiography* 1251-1258.

Hostettler, A., S.A. Nicolau, C. Forest, L. Soler, and Y. Remond. 2006. "Real time simulation of organ motions induced by breathing: first evaluation on patient

data." In *Biomedical Simulation: Third International Symposium, ISBMS 2006, Zurich, Switzerland, July 10-11, 2006. Proceedings*, edited by Matthias Harders, 9-18. Berlin, Heidelberg: Springer Berlin Heidelberg.

Hsu, William M, John F. Hughes, and Henry Kaufgamn. 1992. "Direct Manipulation of Free-Form Deformations." *SIGGRAPH* 177-184.

Hughes, JF, A Van Dam, JD Foley, and SK Feiner. 2013. *Computer Graphics: Principles and Practice.* Pearson Education.

Lewis, John P, Matt Cordner, and Nickson Fong. 2000. "Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation." In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 165-172. ACM Press/Addison-Wesley Publishing Co.

Lorensen, William E., and Harvey E. Cline. 1987. "Marching cubes: A high resolution 3D surface construction algorithm." *SIGGRAPH '87 Proceedings of the 14th annual conference on Computer graphics and interactive techniques.* New York, NY: ACM. 163-169.

Marler, T., N. Capdevila, J. Kersten, A. Taylor, and S. Wagner. 2016. "Human SImulation for Task-Based Survivability Analysis." *International Journal of Human Factors Modeling and Simulation.*

Meier, U., O. Lopez, C. Monserrat, M.C. Juan, and M. Alcaniz. 2005. "Real-time deformable models for surgery simulation: a survey." *Computer Methods and Programs in Biomedicine* 183-197.

Sederberg, Thomas W., and Scott R. Parry. 1986. "Free-Form Deformation of Solid Geometric Models." *SIGGRAPH* 151-160.

Segars, W. P., M. Mahesh, T. J. Beck, E. C. Frey, and B.M .W. Tsui. 2008. "Realistic CT Simulation using the 4D XCAT Phantom." *Medical Physics* 3800-3808.

Taubin, Gabriel. 1995. "Curve Surface Smoothing Without Shrinkage." *Fifth International Conference on Computer Vision.* Cambridge, MA: IEEE. 852-857.

Ungerleider, H. E., and C. P. Clark. 1939. "A Study of the Transverse Diameter of the Heart Silhouette with Prediction Table Based on the Teleroentgenogram." *American Heart Journal* 92-102.

Yang, Jingzhou, Salem Rahmatalla, Tim Marler, Karim Abdel-Malek, and Chad Harrison. 2007. "Validation of Predicted Posture for the Virtual Human Santos™." In *Digital Human Modeling, Lecture Notes in Computer Science, Vol.4561*, 500-510.

Yang, Pei, Song Bifeng, Han Qing, and Ou Baiyu. 2009. "A Direct Simulation Method for Calculating Multiple-hit Vunerability of Aircraft with Overlapping Components." *Chinese Journal of Aeronautics* 612-619.

Zaidi, Habib, and Xie George Xu. 2007. "Computational Anthropomorphic Models of the Human Anatomy: The Path to Realistic Monte Carlo Modeling in Radiological Sciences." *Annual Review of Biomedical Engineering* 471-500.

Zhu, Jiehua, Shiying Zhao, Yangboo Ye, and Ge Wang. 2005. "Computed Tomography Simulation with Superquadrics." *Medical Physics* 3136-3143.