

2009

Determination of critical factors for fast and accurate 2D medical image deformation

Catherine Elise Peloquin
Iowa State University

Follow this and additional works at: <http://lib.dr.iastate.edu/etd>

 Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Peloquin, Catherine Elise, "Determination of critical factors for fast and accurate 2D medical image deformation" (2009). *Graduate Theses and Dissertations*. 11085.
<http://lib.dr.iastate.edu/etd/11085>

This Thesis is brought to you for free and open access by the Graduate College at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Determination of critical factors for fast and accurate 2D medical image deformation

by

Catherine Elise Peloquin

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Co-majors: Mechanical Engineering, Human Computer Interaction

Program of Study Committee:
Eliot H. Winer, Major Professor
James H. Oliver
Matthew C. Frank

Iowa State University

Ames, Iowa

2009

Copyright © Catherine Elise Peloquin, 2009. All rights reserved.

TABLE OF CONTENTS

TABLE OF FIGURES	v
ABSTRACT.....	vii
INTRODUCTION	1
Medical imaging	1
X-ray imaging	1
Magnetic resonance imaging.....	3
Digital Imaging and Communications in Medicine (DICOM) format	4
Three-dimensional representations of medical image data	5
Volume models	6
Indirect volume rendering.....	8
Surface models.....	9
Usage of three-dimensional models in training and planning	10
Motivation	12
Thesis organization.....	15
LITERATURE REVIEW	16
Tissue deformation models.....	16
Physically based models.....	16

Linear elasticity.....	16
Finite element method (FEM).....	17
Mass spring method	19
Kinematically based models.....	21
Free form deformation lattice	21
Proxy meshes	21
Chainmail.....	22
Visualization.....	24
Current interfaces	24
User requirements	31
Research issues	31
METHODOLOGY	33
ChainMail variation selection.....	36
Tissue classification for ordered queue processing.....	41
Interactive tissue classification.....	42
Neighbor determination and addition to queue	44
Relaxation.....	44
Element data structure.....	45
RESULTS	46

Truth cube composition	46
Parameter selection	48
Single DICOM slice truth cube deformation	49
Deformation application and queue building	52
Ideal image generation.....	52
Displacement error results	53
Single slice processing time	58
CONCLUSIONS.....	60
Summary and conclusions	60
Future work.....	60
REFERENCES	62

TABLE OF FIGURES

Figure 1: The first X-ray image, taken in 1895 (http://prl.aps.org/50years/timeline/Alpha)	2
Figure 2: CT image of the Visible Human male head (a) and MRI image of the Visible Human male head (b) [5]	4
Figure 3: Example DICOM file with data elements: the header is shown in blue, the image data shown in red	5
Figure 4: Artificially colored volume render of chest study (left), and same volume render using "clipping planes" to look into the tissue (right)	7
Figure 5: Volumetric model of knee used in Mitsubishi knee arthroscopy simulator [9]	7
Figure 6: Indirect volume render of a MR data set in ROBO-SIM surgical simulator [10]	8
Figure 7: Surface render of the bones, muscles, and veins of the Visible Human Male [12]	10
Figure 8: Immersive Corporation's LapVR Surgical Simulator [14]	11
Figure 9: Abdominal inflation during laparoscopic procedure (http://thefirsthundred.files.wordpress.com/2009/04/laparoscopy.jpg)	13
Figure 10: Elements of software-assisted surgical planning algorithms	14
Figure 11: Finite element model of uterus with approximately 2000 elements [26]	18
Figure 12: Bending of a knee CT dataset	22
Figure 13: Deformation of a 2D chain mail when a selected link is moved [43]	23
Figure 14: Volume representation of medical data seen in Amira	25
Figure 15: 2D images and clipped volume representations seen in 3D-DOCTOR interface [48]	26
Figure 16: The Visible Human dataset shown in VolView	27

Figure 17: 2D image viewing in OsiriX	28
Figure 18: Volume model shown in BodyViz	29
Figure 19: Volume model with surgical tools in Isis software	30
Figure 20: Proposed deformation algorithm	35
Figure 21: Schematic of ChainMail constraints between a deformed point (shown in black) and its right and bottom neighbor [43]	37
Figure 22: The original positions of elements A and B (a), and the repositioning of element B after sponsor A is moved to A*[56]	39
Figure 23: Loaded DICOM image in tissue classification mode	43
Figure 24: Classified tissue	43
Figure 25: The sponsor point, shown in red, and its eight neighbors, shown in green	44
Figure 26: Schematic of truth cube uniaxial experiment, where L is the distance between the loading points to the center of the cube and m is the mass of the counter weights [59]	47
Figure 27: Truth cube undeformed center slice (left) and slice with 5% strain applied (right)	48
Figure 28: Center slice image default air classification (a), classified as soft tissue and bone (b), deformed with 5% strain using scheme B (c)	51
Figure 29: Element spacing 0.37 mm undeformed ideal slice (a), deformed with A constraints (b), deformed with B constraints (c)	57
Figure 30: Single slice processing times	58

ABSTRACT

The advent of medical imaging technology enabled physicians to study patient anatomy non-invasively and revolutionized the medical community. As medical images have become digitized and the resolution of these images has increased, software has been developed to allow physicians to explore their patients' image studies in an increasing number of ways by allowing viewing and exploration of reconstructed three-dimensional models. Although this has been a boon to radiologists, who specialize in interpreting medical images, few software packages exist that provide fast and intuitive interaction for other physicians. In addition, although the users of these applications can view their patient data at the time the scan was taken, the placement of the tissues during a surgical intervention is often different due to the position of the patient and methods used to provide a better view of the surgical field. None of the commonly available medical image packages allow users to predict the deformation of the patient's tissues under those surgical conditions.

This thesis analyzes the performance and accuracy of a less computationally intensive yet physically-based deformation algorithm- the extended ChainMail algorithm. The proposed method allows users to load DICOM images from medical image studies, interactively classify the tissues in those images according to their properties under deformation, deform the tissues in two dimensions, and visualize the result.

The method was evaluated using data provided by the Truth Cube experiment, where a phantom made of material with properties similar to liver under deformation was placed under varying amounts of uniaxial strain. CT scans were before and after the deformations. The deformation was performed on a single DICOM image from the study that had been

manually classified as well as on data sets generated from that original image. These generated data sets were ideally segmented versions of the phantom images that had been scaled to varying fidelities in order to evaluate the effect of image size on the algorithm's accuracy and execution time. Two variations of the extended ChainMail algorithm parameters were also implemented for each of the generated data sets in order to examine the effect of the parameters.

The resultant deformations were compared with the actual deformations as determined by the Truth Cube experimenters. For both variations of the algorithm parameters, the predicted deformations at 5% uniaxial strain had an RMS error of a similar order of magnitude to the errors in a finite element analysis performed by the truth cube experimenters for the deformations at 18.25% strain. The average error was able to be reduced by approximately between 10-20% for the lower fidelity data sets through the use of one of the parameter schemes, although the benefit decreased as the image size increased. When the algorithm was evaluated under 18.25% strain, the average errors were more than 8 ½ times that of the errors in the finite element analysis. Qualitative analysis of the deformed images indicated differing degrees of accuracy across the ideal image set, with the largest displacements estimated closer to the initial point of deformation. This is hypothesized to be a result of the order in which deformation was processed for points in the image.

The algorithm execution time was examined for the varying generated image fidelities. For a generated image that was approximately 18.5% of the size of the tissue in the original image, the execution time was less than 15 seconds. In comparison, the algorithm processing time for the full-scale image was over 3 ½ hours.

The analysis of the extended ChainMail algorithm for use in medical image deformation emphasizes the importance of the choice of algorithm parameters on the accuracy of the deformations and of data set size on the processing time.

INTRODUCTION

Medical imaging

Medical imaging is a primary tool used by physicians to diagnose, monitor, and treat patients. Medical imaging has evolved through the years to include the following major modalities: x-ray imaging, magnetic resonance imaging (MRI), nuclear medicine imaging modalities, and ultrasound imaging [1].

X-ray imaging

X-rays were first utilized for medical imaging in 1895 (Figure 1), and this oldest form of medical imaging is still widely used today. In this type of image acquisition, a cathode is heated in a vacuum tube also containing an anode. Electrons flow from the cathode to the anode. When these electrons strike the nuclei of atoms in the anode, X-ray radiation is released. It is this radiation that is detected by a film plate or modern detector. Attenuation of the X-ray depends on the density of the tissues through which it passes; denser tissues result in lighter areas on the final image [2].



Figure 1: The first X-ray image, taken in 1895 (<http://prl.aps.org/50years/timeline/Alpha>)

X-ray imaging has developed from the single 2D projection to computed tomography, or CT, which provides 3D information about the subject being studied. CT scanners utilize multiple detectors and an x-ray source which rotates about the study object to collect 3D physiological data. When the first CT scanner was made in the 1970's, data acquisition for a single image took several hours. Now, state of the art CT machines can collect up to 4 slices of data in 350 ms and reconstruct a 515x512 image in less than a second [3].

Magnetic resonance imaging

Like CT imaging technology, magnetic resonance imaging provides 3D information about the subject, but unlike x-ray based technology, the scanning technology does not transmit external radiation through the subject but instead utilizes an electromagnet to cause the protons in the subject to align. This causes the body tissues produce magnetic fields which are detected by the scanner and are used to construct an image [4].

The images taken from CT and MRI scans share some common verbiage. Images taken of the same subject are called a study. Images taken of that subject in different orientations or at different periods of time are called a series.

The resultant images from CT and MRI scans look very similar, as shown in the images from the Visible Human male dataset seen in Figure 2, but the modalities are generally used for differing types of medical image studies. MRI imaging is more often used for soft tissue studies, where contrast dyes may be injected into the patient to highlight certain tissue types in the final image. CT imaging is more often used for a wider range of tissue types including bone.

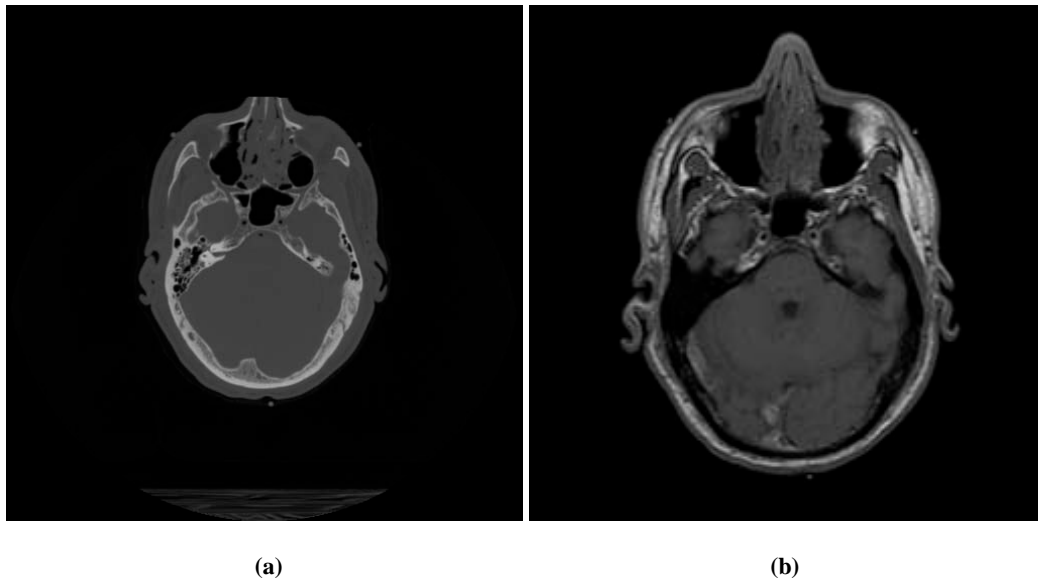


Figure 2: CT image of the Visible Human male head (a) and MRI image of the Visible Human male head (b) [5]

Digital Imaging and Communications in Medicine (DICOM) format

The advent of digital medical imaging technologies such as computed tomography led to the formation of a committee comprised of members of the American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA) to create a standard format for the storage and transmission of digital medical images. In 1985, this committee released the standard, now called the Digital Imaging and Communications in Medicine (DICOM) format. DICOM stores information such as the patient's name, age, and doctor, along with technical specifications of the image acquisition method, and the density information of each image [5].

DICOM files are composed of many different types of data elements. These elements are defined in the DICOM standard's data dictionary, where each element is assigned a tag, a name, value characteristics, and semantics for that element. Examples of different elements

are: the date the study was performed, the modality, the manufacturer of the imaging machine, the imaging institution's name, the referring physician's name, the patient's name, the patient's age, and the image data. A DICOM file is composed of two major parts, the image data and non-image data referred to as the "header", although it is not necessarily at the beginning of each file. The header contains attributes that give information about the patient, the procedure, and other information. It is where data elements such as the patient name, the number of different "slices" if the file contains multiple compressed images, the number of rows and columns if image data, the modality, the sampling spacing of the scanner, and other similar information are stored. The image data contains the density of each pixel for the image or images stored in the file. A simplified illustration of this format is shown in Figure 3.

Patient name	Patient sex	Patient birth date	Scan date	Institution
Referring physician	Modality	Manufacturer	Rows	Columns
Image Data				

Figure 3: Example DICOM file with data elements: the header is shown in blue, the image data shown in red

Three-dimensional representations of medical image data

The choice of modeling a physical object as a volume or a surface depends greatly on the fidelity needed for the intended use and the computational power and time available for modeling. As shown in a blind study which had participants rate reconstructed volume and

surface models of colonic, gastric, and bronchial structures, the participants preferred the volume renders over surface renders [6]. Although the surface render was less computationally intensive, tissue density for each of the structures varied across patients, and the surface render was also more sensitive to noise and image artifacts, leading to “absence or exaggeration of detail”. In contrast, volume rendering resulted in greater detail and greater dimensionality of the data. Volumetric models also result in more accurate deformations over thin regions, through inhomogeneous tissue, and for cutting or suturing operations [7].

Volume models

Volumetric models are completely solid models, with data points located in the interior and on the exterior of the object being modeled. In medical imaging, these models can be created from raw data from scans, where each element in the three-dimensional array of data is a sampled point from the original scan image. This use of the original data makes the volume render more accurate than other modeling methods in terms of representing the original data and not losing fidelity [8]. A volume render of a CT chest study is seen in Figure 4. The volume model may also be easily “cut” into through the use of clipping planes to show the model’s internal data.

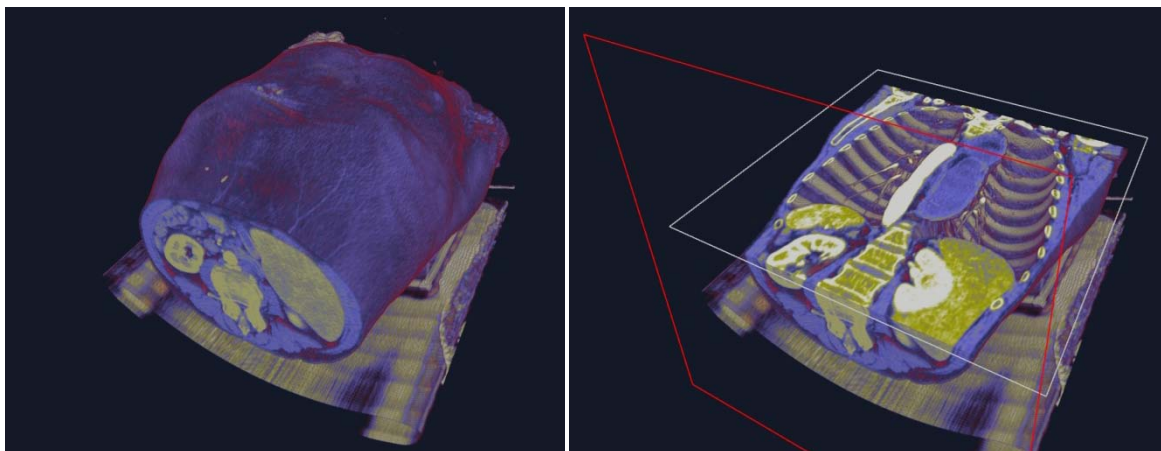


Figure 4: Artificially colored volume render of chest study (left), and same volume render using "clipping planes" to look into the tissue (right)

An example of using volume representations can be seen in a knee arthroscopy simulation by Mitsubishi as shown in Figure 5 [9]. The models used in this simulation were created by hand-segmenting the femur, fibula, and tibia from a 512x512x124 dataset by orthopedic surgeons, then generating volumes from the segmented images.

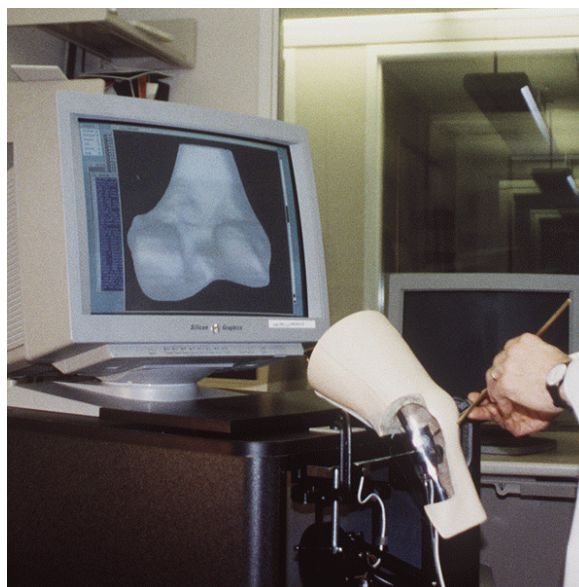


Figure 5: Volumetric model of knee used in Mitsubishi knee arthroscopy simulator [9]

Although volume modeling maintains the level of detail in the original scans, there are drawbacks. Volume models have a greater amount of data than surface models, which requires more computational storage space and power to compute changes to the model.

Indirect volume rendering

Indirect volume rendering is a method that maintains a greater level of data than surface modeling but requires less computational power than volume rendering. In indirect volume rendering, partial volumes or surfaces of interest are extracted from the original volume using segmentation techniques. This decreases the portion of the volume's data that must be maintained and altered in processing. Textures made from the original volume can be applied to the resultant surfaces for greater realism. Radezky et al. have used indirect volume rendering alone [10] and in combination standard volume modeling [11] in their work with surgical simulators like that shown in Figure 6.

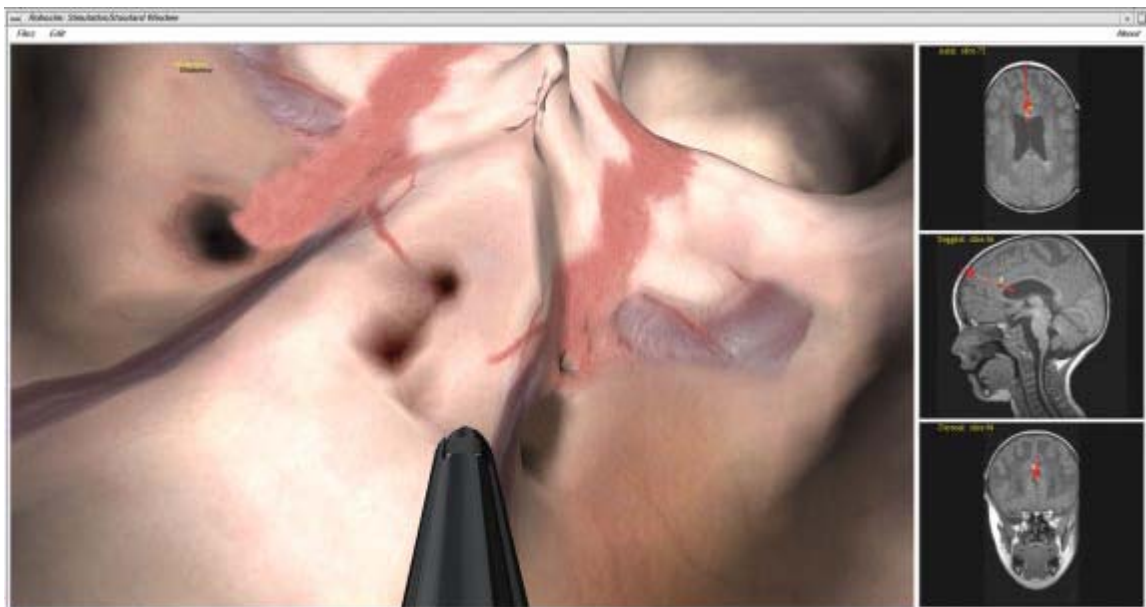


Figure 6: Indirect volume render of a MR data set in ROBO-SIM surgical simulator [10]

Surface models

Surface models are created from a volume dataset by isolating the parts of the dataset that compose a surface and only displaying those polygons. The decreased dataset size results in computational performance increases over volume datasets in exchange for decreased accuracy. Surface models can also result in unrealistic deformations for some tissue types, although they have been shown to function well for cavernous tissues such as blood vessels and the gallbladder [7].

Surface rendering has been used with success at the University of Colorado's Center for Human Simulation in a haptic surgical simulator [12] which utilized the Visible Human Dataset [13]. A render from this application is shown in Figure 7. This surgical simulator includes a PHANToM-controlled virtual scalpel which users can use to make incisions on the volume dataset.

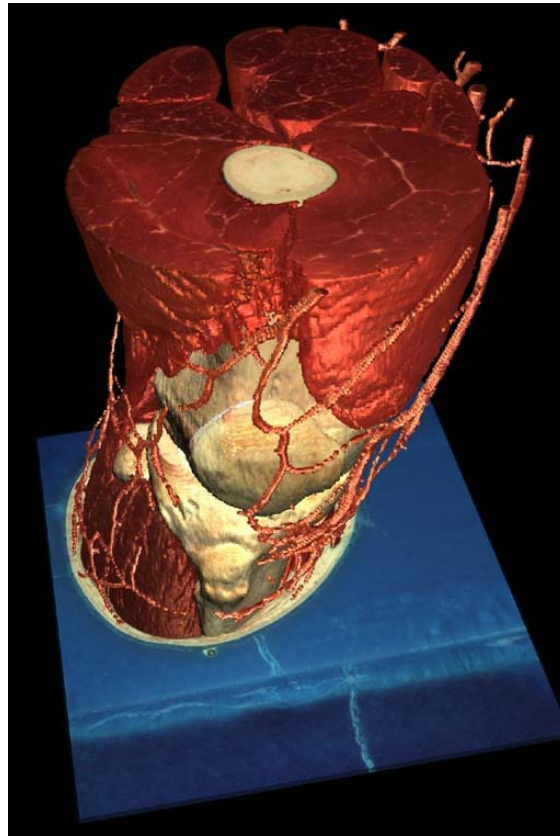


Figure 7: Surface render of the bones, muscles, and veins of the Visible Human Male [12]

Usage of three-dimensional models in training and planning

Surface and volume models have been used to assist in medical diagnosis, surgical planning, and physician training in surgical simulators. One of the areas of medicine where these models are of assistance is in surgical training simulators for laparoscopic surgery. In laparoscopic surgery, surgeons access the patient's body through small ports, rather than through a large incision. Laparoscopic surgeons must rely on a camera inserted through those ports, which removes the surgeons' tridimensional perception, limits tactile feedback, and also requires different mapping between the surgeons' movements and the movement of the instrument than in traditional, "open" surgeries.

Simulators have been developed to assist surgeons in the development of key skills for laparoscopic surgeries. These simulators include those created by Immersion Corporation [14]. The LapVR Surgical Simulator, a laparoscopic simulator, is shown in Figure 8. This simulator allows surgeons to refine their motor skills in preparation for surgery, but the data used in these simulators is artist-generated. Surgeons cannot train on the actual patient data. Immersion Corporation also provides endoscopic, endovascular, arthroscopy, and vascular access simulators, but these, too, use artist-generated images.



Figure 8: Immersive Corporation's LapVR Surgical Simulator [14]

Generated data has also been used for endoscopic surgery training simulation [15], open surgery simulation [16], and suturing simulation [17].

Training simulations with authoring systems for new data input do exist. CathSim [18], a commercial intravascular catheterization simulator, is one such product. This system utilized the occupancy-map algorithm and the ChainMail algorithm to model volumetric deformations. Users of this system could create case studies by creating a deformation mesh on top of a user-provided picture. This system was available for PC. CT and MRI data is not input into the system, however.

Although the use of deformable, three-dimensional models have provided to be a great asset to the field of medicine, the data used in simulation is rarely obtained directly from medical image data from a specific patient. Although these simulators are useful in refining technical skills, these simulators cannot be used to broaden surgeons' understanding of a specific patient's medical data. This limits the usefulness of both the CT and MRI images collected and the simulators themselves.

Motivation

The advancement of medical imaging technology and the increased resolution of scans such as CTs and MRIs, paired with the increased computing capacity of general performance hardware, makes it possible to perform high-quality volume renders for use in pre-surgical planning. Unfortunately, the orientation of the patient during surgery is rarely the same as when the scan was taken, and the patient's tissues are often manipulated and deformed. For example, in minimally invasive abdominal laparoscopic procedures like those in Figure 9, the patient's abdomen is inflated with pressurized gasses in order to improve the surgeon's view of the field and allow her more room for manipulation. This inflation causes

significant distension of the abdominal wall and also affects the position and shape of the organs within the cavity.

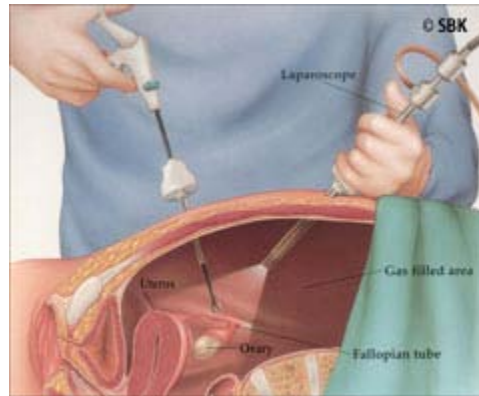


Figure 9: Abdominal inflation during laparoscopic procedure

(<http://thefirsthundred.files.wordpress.com/2009/04/laparoscopy.jpg>)

The difference in tissue positioning between the imaging study and the actual procedure necessitates that surgeons using medical images in their planning must exploit their understanding of anatomy to create a mental model of the deformed patient tissue. This mental model is then verified or proven faulty during the execution of the surgical procedure. If the model is shown to be inaccurate, several actions may be taken depending on the severity of the conflict. If the mental model was only slightly different from the actual patient condition, the surgery may continue as planned. For gross miscalculations, surgeons may have to create a new surgical plan while the patient is sedated or the surgery may have to be rescheduled. This increases both the risk to the patient and the costs incurred.

Discussions with surgeons revealed that creating an accurate mental model required amassing a large collection of previous cases from which to refer. Less experienced surgeons and their patients are then at a disadvantage in pre-surgical planning. Some

conditions are also so rare it is unlikely even an experienced surgeon can create a highly accurate model.

The required elements for surgical planning software are shown in Figure 10. Once a study has been acquired, the software will need to identify the various types of tissue present in the study, then use the properties of that tissue and other inputs (such as forces applied) to set the parameters that will describe the deformation. These parameters are then input into the deformation algorithm to create a 3D model of the deformed tissue.

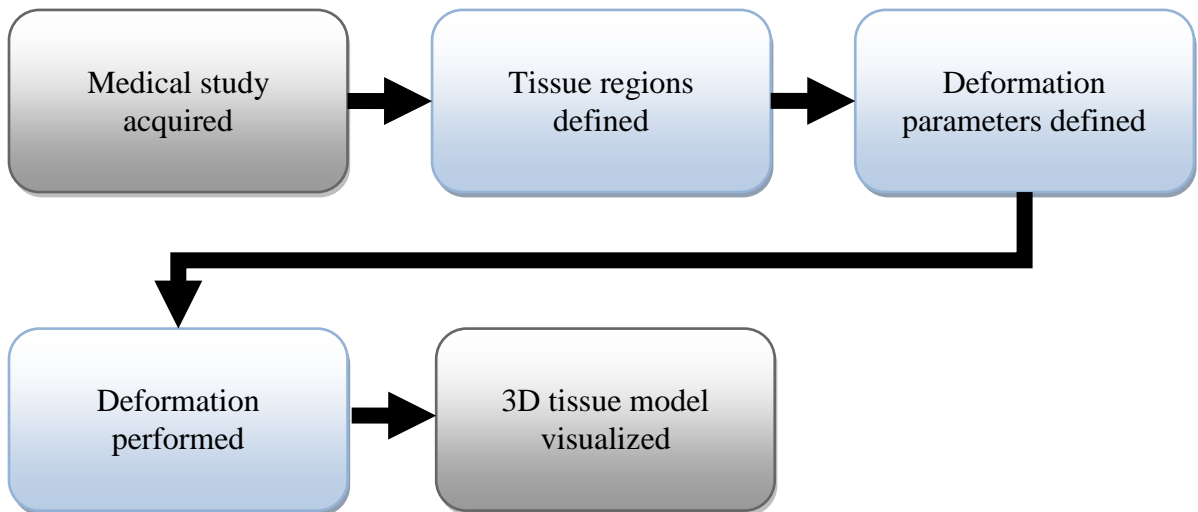


Figure 10: Elements of software-assisted surgical planning algorithms

Some software attempts to provide surgeons with an accurate visualization of the patient data, but this software is often unintuitive, uses deformation algorithms that are too intensive for commodity hardware, or uses algorithms that are not tied to the actual physical model of the tissues. The goal of this thesis is to focus on the elements of the surgical planning software highlighted in blue in Figure 10 in order to create the foundation for a

robust and effective tissue deformation algorithm using existing deformation techniques, combined with basic HCI knowledge to design a usable interface.

Thesis organization

The organization of the remainder of the thesis is as follows: the second chapter will be devoted to an in-depth review of the current literature on tissue deformation and intuitive interfaces in medical imaging, followed by the third chapter's explanation of the methodology used to develop deformations for this thesis. Test cases will be run and analyzed in fourth chapter's results section, while conclusions will be discussed in the last chapter.

LITERATURE REVIEW

Tissue deformation models

As is required for deciding between surface and volume models of tissue, choosing a deformation algorithm for the tissue model is a tradeoff between responsiveness and fidelity [8]. As the deformation algorithm becomes more accurate, it becomes more complex and computationally expensive, leading researchers to choose between several methods to suit their individual demands. As described by Liu et al. [19], deformation models can be broken into two large categories: physically based models and kinematically based models.

Kinematic models do not account for the object's mass, forces, or other physical properties during the deformation, while physics-based models do incorporate those material properties.

Physically based models

Linear elasticity

One algorithm for calculating the displacement of tissues under stress is the linear elastic model based on Hooke's law seen in Equation 1. For this representation of the equation, \mathbf{K} is the stiffness matrix, which is symmetric, positive, definite, and sparse, \mathbf{u} is the unknown displacement field, and \mathbf{f} is the external force.

$$[\mathbf{K}]\mathbf{u} = \mathbf{f} \quad \text{Equation 1}$$

This algorithm has been used in medical modeling [20, 21]. A modified algorithm, which combines modeling of elastic tissues with an accounting for muscular and rigid tissues, has also been explored [22].

A shortcoming of the linear elastic model is its inability to accurately model large (greater than 10% of the total mesh size) or nonlinear displacements. Modified, nonlinear models have been created to more accurately model under those conditions, including the St. Venant-Kirchoff [20,23] and Mooney-Rivlin models [24].

Finite element method (FEM)

Finite element analysis has been used with success in engineering applications. In this type of analysis, the volume is decomposed into very small elements of triangular, quadrilateral, tetrahedral, or similar shapes. The deformation of the tissues is then determined based on a shape function and global parameters for the volume or surface [25]. The finite element problem is continuous, but solves for each element individually by solving the system of equations shown in Equation 2, where \mathbf{K} is the symmetric, sparse stiffness matrix, \mathbf{U} is the nodal displacement vector, and \mathbf{F} is the force vector.

$$\mathbf{KU} = \mathbf{F}$$

Equation 2

Székely et al. used the FEM to model the deformation of soft tissues for use in a laparoscopic surgery simulation (Figure 11) [26]. The computational cost for this simulation was high, however. A parallel cluster of high-performance computers were needed to computer the deformations in real time.

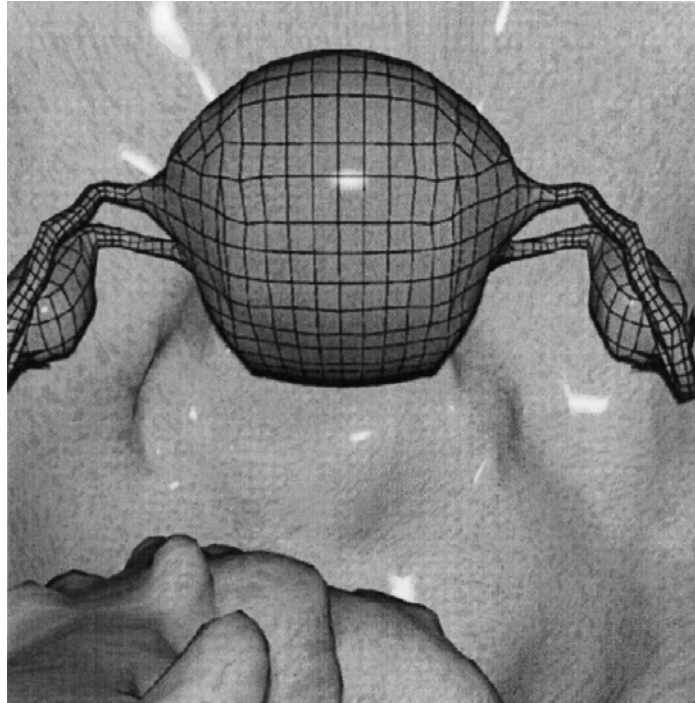


Figure 11: Finite element model of uterus with approximately 2000 elements [26]

Finite element modeling has also been used to successfully model the deformation of brain tissue with errors on the range of millimeters to sub-millimeters depending on the particular case [27, 28].

The decomposition of the volume into elements results in a computationally intensive problem, but the solutions obtained through the use of this method are physically realistic in comparison to other methods [8]. Depending on the problem size and the computational resources available, some volumes must have large elements in order to be solvable on current hardware. As the size of the elements in the volume increases, the solution accuracy can decrease.

Attempts have been made to increase the computational speed for finite element models, including fast finite elements (FFE) [29], which sacrifices time in pre-computation in

order to provide some real-time simulation [30]. This method does not accurately compute large deformations or the behaviors of viscous volumes, however [31]. Off-loading of calculations onto the GPU has also been examined [32, 33]. Although this method is notably faster than FEA, the failure to compute large deformations makes this method less viable for simulation of laparoscopic surgeries, where the abdominal walls are displaced large amounts when the patient is inflated with pressurized gas in order to provide a large field of view and workspace for the surgeon.

Mass spring method

A simpler, and more commonly used, method for modeling tissue deformation is the mass spring method. In this method, masses are attached to vertices and springs act as connectors between each mass. These springs may have different stiffness values to represent different tissue types [34]. For the motion of a point i , the displacement can be described by the Equation 3, where μ is the mass, γ is the damping factor, K is the stiffness of the spring connecting point i to points in the neighborhood of i, j .

$$\mu \frac{d^2 r_i}{dt^2} = -\gamma \frac{dr_i}{dt} + \sum_{j \in N(i)} K_{i,j} \frac{(l_{i,j}^0 - \|r_i r_j\|) r_i r_j}{\|r_i r_j\|} \quad \text{Equation 3}$$

Although the deformations resultant from a spring model have different physical mechanism than actual tissue deformation, spring models have been shown to approximate a linear finite element model as long as the deformations are small [35]. The mass spring problem can also be computed on standard hardware [36].

The mass-spring method has been used with success in real-time surgical simulators to model deformation based on collision with surgical instruments with soft tissues. Brown et al. [37] were able to minimize computation time for the mass spring system by taking advantage of the damping effect of human tissues to only compute deformations in the local region where they were the greatest. Their methods were tested at 30 Hz with up to 8000 masses and 66120 springs [38]. More recently, experimentation has been done with offloading spring compression and extension calculations to the GPU; a mesh with 512^2 vertices was able to be computed 480 times per second [39].

The mass spring method has several drawbacks. First, inhomogeneous tissues or tissues with nonlinear elasticity cannot be correctly modeled with the basic mass spring algorithm. Because the model is not based on the physical characteristics of real tissues, the spring constants for the system components may be difficult to determine appropriately, which results in incorrect deformation. Spring constants that are too high may also result in “stiffness”, where the system is unstable and is slow in simulation [36]. These models are also prone to oscillation during real time deformation and do not maintain constant volume during deformation [31]. The small deformations for which the mass-spring system is valid when compared to a linear finite element model also mean that larger deformations, such as when tissues are manipulated in a surgical setting, may not be able to be accurately modeled with this method.

Kinematically based models

Free form deformation lattice

Sederberg and Parry [39] were able to model free-form deformation of a surface by utilizing control points on a free form deformation lattice. As the position of the control points changes, the surface is deformed using a trivariate tensor product Bernstein polynomial to calculate the displacement of each point in the surface. This algorithm has been used as the basis for other surface deformation algorithms [40].

Proxy meshes

Rezk-Salama and associates deformed volumes via proxy meshes, which utilized the original volume model but decreased the computation required to determine deformation [41]. In this method, rather than transforming the solid geometry directly, an intermediate model of the geometry is created using 3D textures. This intermediate model is composed of a set of sub-cubes, known as “patches”, rather than maintaining the fidelity of the entire original geometry. Then, rather than calculating deformation on the original model whenever deformation is performed, the deformation is calculated as translation of the texture coordinates for the model. Deformations within the patches are computed using a weighted trilinear interpolation. Constraints for the edges and vertices of the model can be created to prevent gaps in the model when it is deformed.

A similar method was implemented by Correa et al. [42]. In that method, a 2D displacement map containing control points was generated. Displacements on this map were then interpolated to the 3D volume as a whole. By offloading computations to the GPU, real-

time deformation and cutting were able to be performed. Resultant deformations for a knee CT dataset are seen in Figure 12.

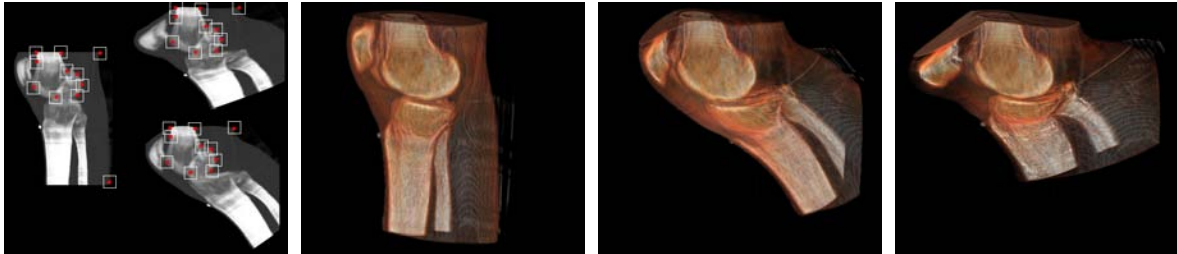


Figure 12: Bending of a knee CT dataset

Chainmail

In the Chainmail algorithm, developed by Gibson [43], deformation is transmitted through a series of “link” elements, shown in Figure 13. Collision detection and an energy minimizing elastic relaxation algorithm are used to propagate deformation when a single link is moved. As outlined by Gibson, elements in the volumetric object all have constraints that describe the maximum stretch and compression that can be performed on that element before that movement causes its neighbors to also become displaced. At that point, the element’s top, left, bottom, and right neighbors are then added to a queue of candidate points to be processed. Each of those neighbors is processed in first-in-first-out (FIFO) order in similar fashion as the first point until the queue is emptied. Elastic relaxation is then performed to relax the shape into a minimum energy configuration. This method has been used to simulate the inflation of a balloon in the blood vessel lumen for a 2D angioplasty simulation [44].

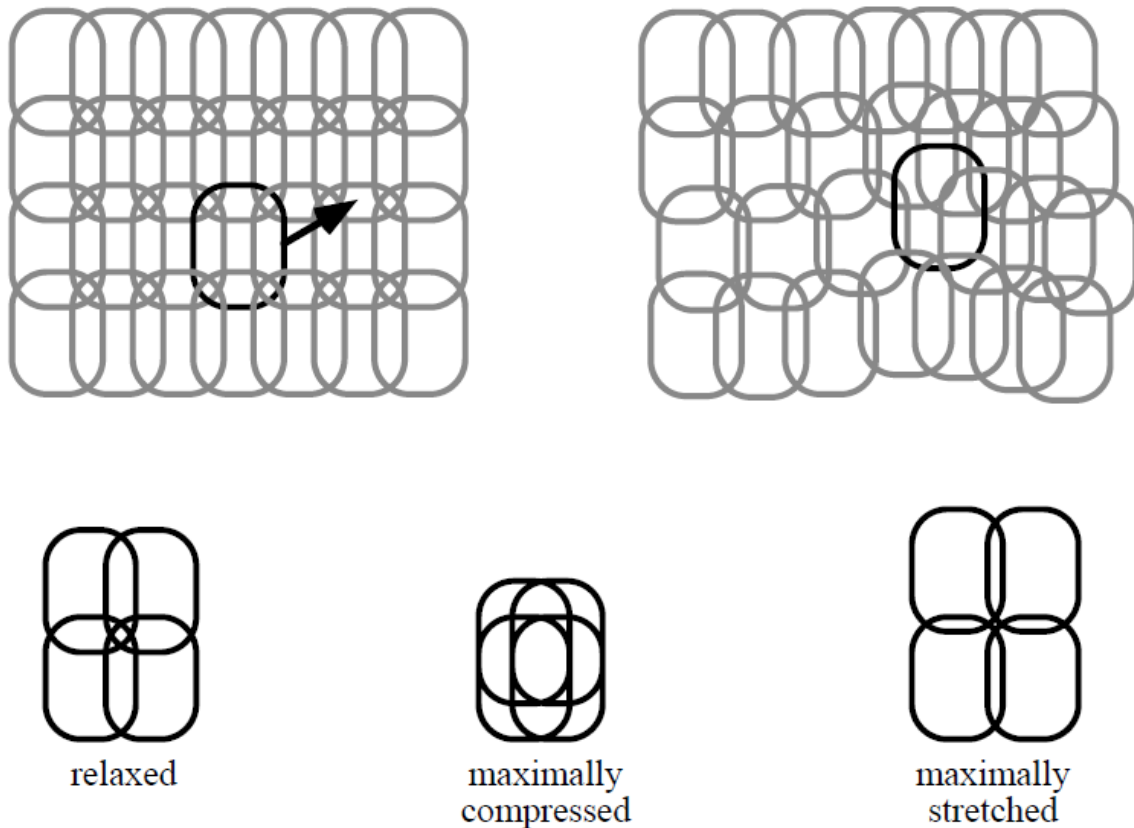


Figure 13: Deformation of a 2D chain mail when a selected link is moved [43]

The basic ChainMail algorithm has also been modified to correctly process inhomogeneous materials in the enhanced ChainMail algorithm [45]. In this modification, the candidate queue is not processed in FIFO, but is instead prioritizes elements with stiffer properties. Although this increases the processing time, it enables rigid materials to maintain their correct positions relative to each other.

The ChainMail algorithm has been used for tissue deformation calculation for a variety of reasons. The parameters used the enhanced ChainMail algorithm can be modified to approximate the physical behavior of a variety of tissue types. Also, although these

parameters used in the algorithm are tied to the physical properties of the tissue, the deformation calculations are far less computationally intensive than those used in finite element analysis. Although computation time will suffer for large data sets no matter what tissue deformation method is used, the ChainMail algorithm's relatively low overhead allows the method to use high-resolution data from CT and MRI images.

Fast, relatively accurate deformations with real patient tissue are necessary in order for the proposed surgical planning technology to be of use. The data sets provided in typical medical scans are generally 512x512 pixels for each image, with a study containing between 40 and several hundred images. The sheer quantity of data provided suggests the use of a non-physically-based model in order to increase computation speed. The ChainMail algorithm provides an efficient mode of processing while still maintaining a tie to the tissue properties through its parameters and has been used for other medical applications [44-46]. For this reason, the ChainMail algorithm and its variations are the most promising for the application proposed.

Visualization

Current interfaces

Many different medical imaging programs exist in the market today, including Amira, 3D-DOCTOR, VolView, OsiriX, BodyViz, and Isis.

The Amira [47] software is available on Windows, Mac OS, and Linux as a base package with a number of add-on functional modules available for purchase. The base package includes the ability to import and export to and from a number of file formats (not

including DICOM). It also includes surface and volume generation, material statistics, analysis tools, clipping and cutting tools, and presentation tools for the generation of animations, demos, annotation, and stereoscopic tools. The functional modules add advanced options for the following: meshes, microscopy, skeletonization, molecular models, very large data, quantification, immersive virtual reality, specialized readers for other image formats such as DICOM, and a developer option for the creation of new components. Amira is seen in Figure 14.

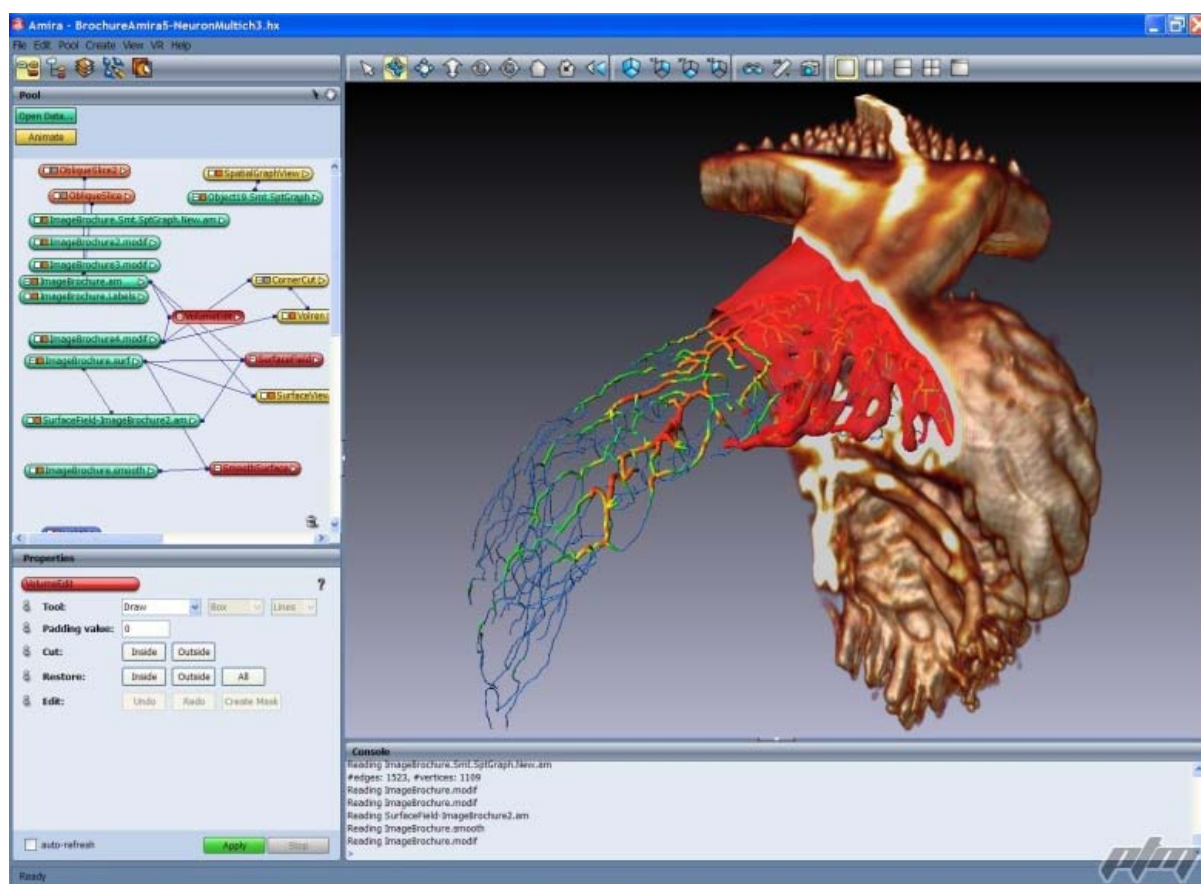


Figure 14: Volume representation of medical data seen in Amira

3D-DOCTOR [48] is an imaging program that has been approved by the United States Food and Drug Administration for medical imaging applications. This application can

read and display a number of 2D image formats, some in color, including the following: DICOM, TIFF, BMP, JPEG, Interfile, PNG, PGM, GIF, raw image data, and uncompressed file formats. In addition to being able to save images and 3D objects to a number of file formats, 3D-DOCTOR also allows users to view images as a surface and volume, re-slice data along an arbitrary analysis, and perform segmentation and volume measurement. It is available on Windows platforms only. 3D-DOCTOR is shown in Figure 15.

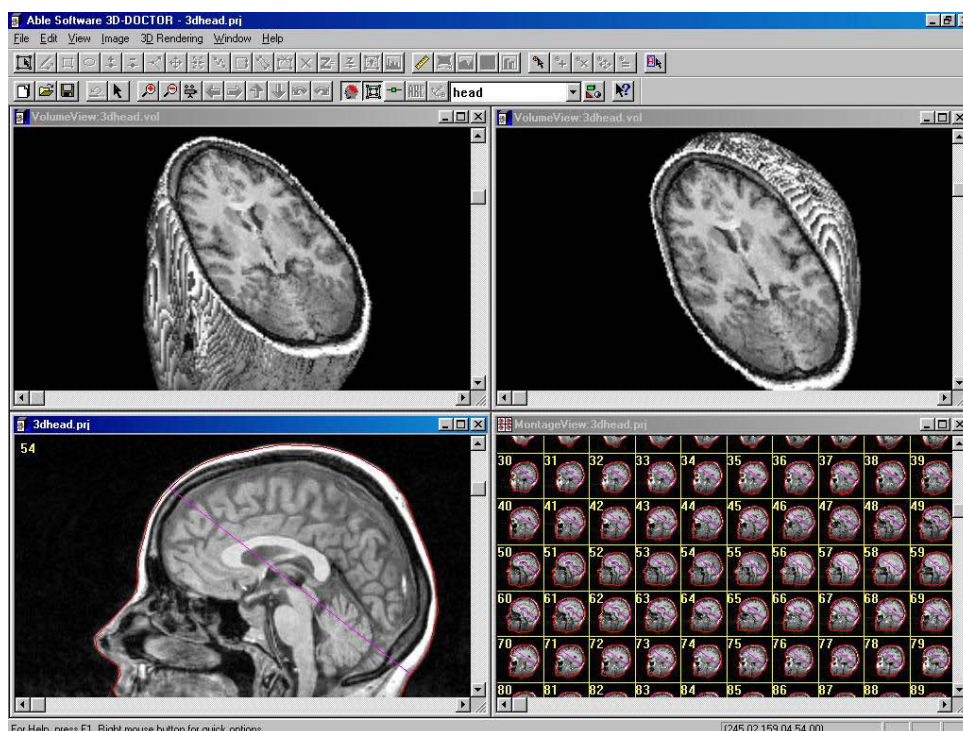


Figure 15: 2D images and clipped volume representations seen in 3D-DOCTOR interface [48]

VolView [49] is available for Mac or PC and can load image data types including DICOM, TIFF, PNG, and JPG, then can display the data a volume render, a maximum intensity projection, 2D orthogonal images, or as a movie. Tools included in VolView

include image manipulation such as window/level adjustment, panning, and zooming as well as cropping, and light box views. The VolView interface is shown in Figure 16.

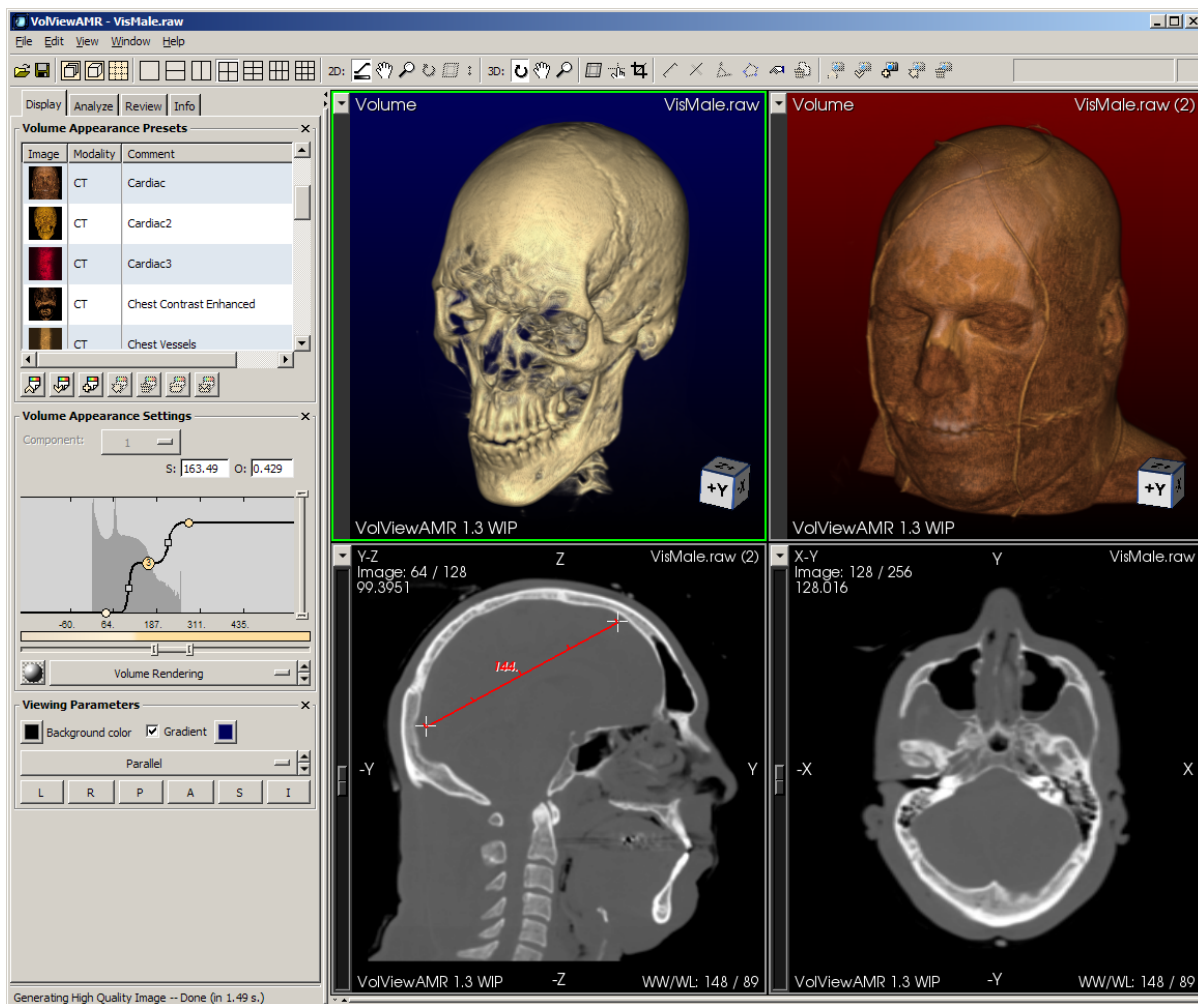


Figure 16: The Visible Human dataset shown in VolView

OsiriX [50] is available only on the Mac OS and on the iPhone. Like VolView, it can load multiple file types such as DICOM, TIFF, JPG, PDF, AVI, MPEG, and QuickTime. As well as displaying orthogonal views, surface rendering, and volume rendering, OsiriX can view 4D data, or 3D data with time lapse, and 3D data, or 3D data with time lapse and functional dimensions. OsiriX is shown in Figure 17.

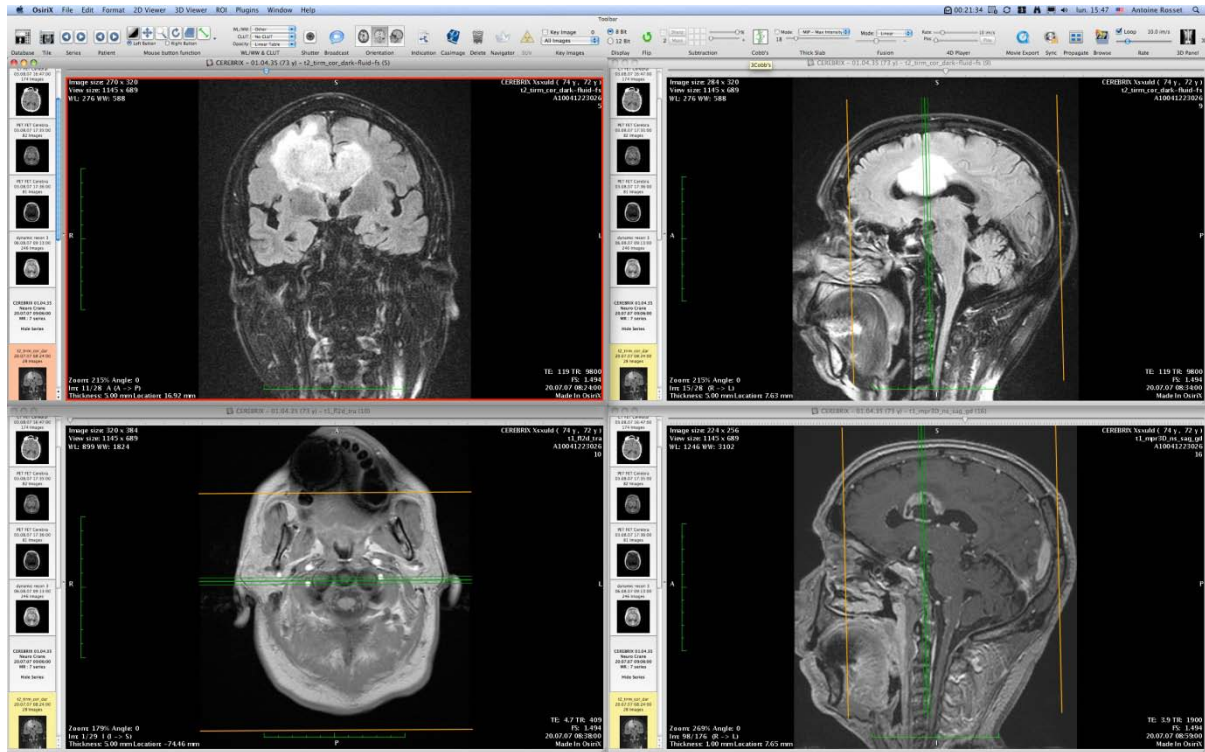


Figure 17: 2D image viewing in OsiriX

In contrast to the previous software packages, the BodyViz [51] software targets surgeons. The software, available for PC, enables users to navigate through volume representations of actual patient data through the use of an Xbox 360 controller. In addition, the software provides users with clipping planes, which are used to view the inside of the volume, virtual surgical tools that can be moved through the volume, and a variety of volume color schemes to highlight different tissue types.

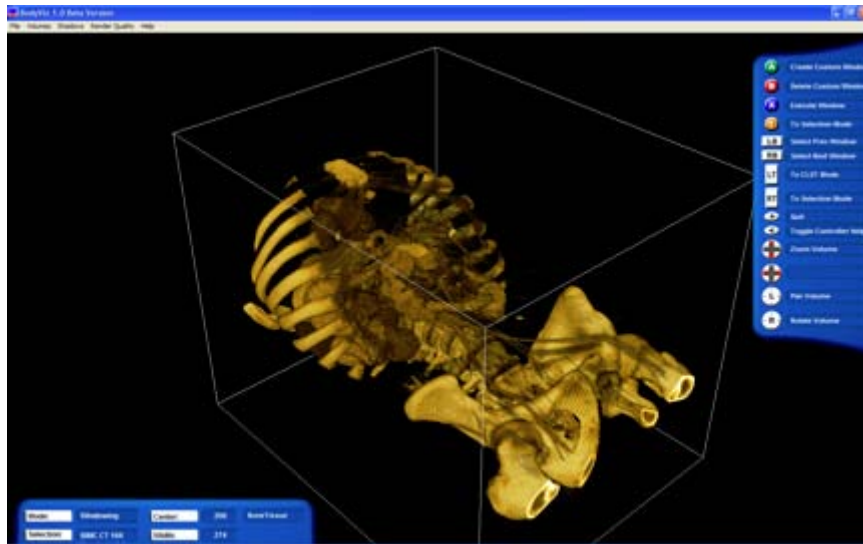


Figure 18: Volume model shown in BodyViz

The Isis software package [52], created at Iowa State University, also limits the number of software features available in favor of providing simplified applications for non-radiologists. Isis is available in both a desktop and a virtual reality version and allows for syncing between different platforms. Like BodyViz, the virtual reality version allows interactive viewing of volumes with gamepad navigation, different color schemes, clipping planes, and surgical tools. A volume model and surgical tools visualized in the virtual reality version of Isis is shown in Figure 19. The desktop version has viewing of 2D and volume models with various color schemes, surface model generation and viewing. The desktop version of the software uses mouse navigation.

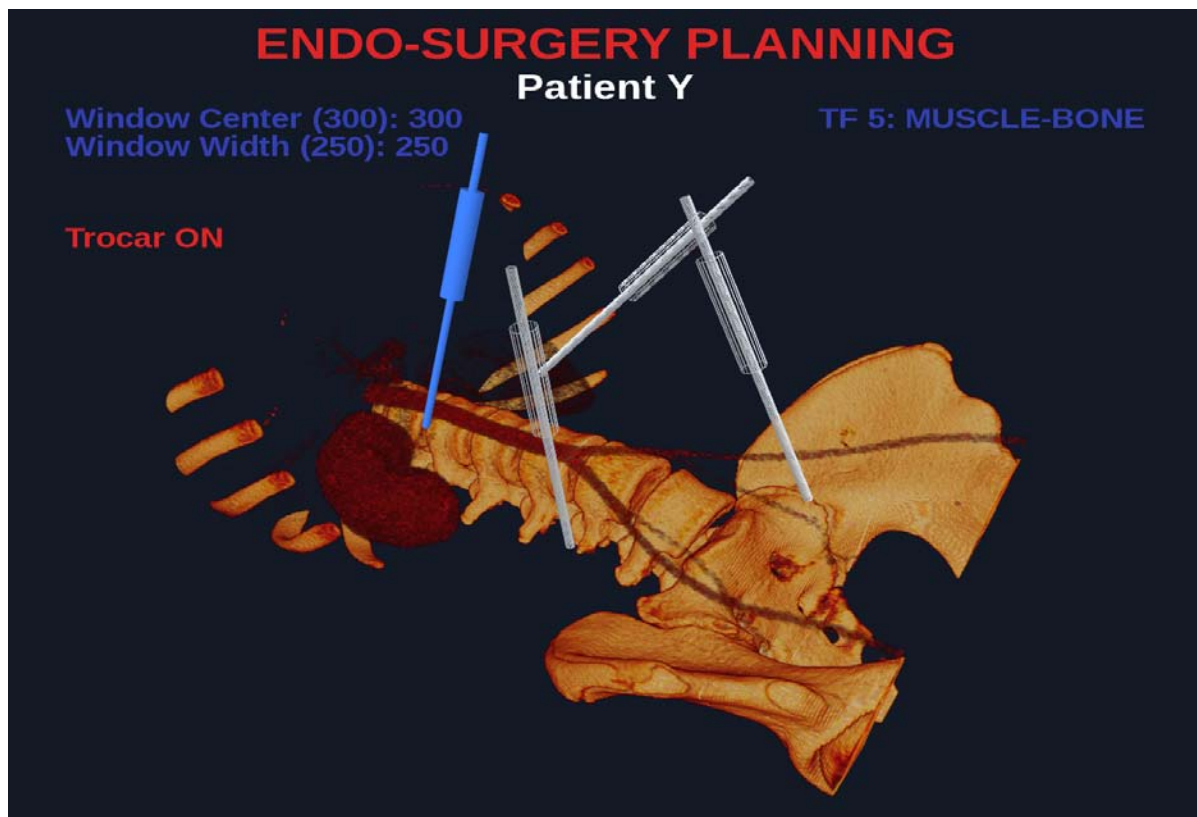


Figure 19: Volume model with surgical tools in Isis software

Although Amira, 3D-DOCTOR, VolView, and OsiriX desktop software have enabled a greater range of physicians to use patient data in their diagnostics and planning, these interfaces do have drawbacks. These programs are targeted to the radiologist, and so use visualization methods, like 2D grayscale images, that are more familiar to the radiologist than to the surgeon. These software packages also have an incredibly wide range of functionalities and allow for extreme tuning of settings within each functionality. Although this may be of use to radiologists, the surgeons interviewed desired a software package that was intuitive, simpler to use, allowed easy manipulations, and offered attractive visualizations. BodyViz and Isis attempt to tailor their software to surgeons by emphasizing

fast interactions with default settings, but neither they nor Amira, 3D-DOCTOR, VolView, and OsiriX include the ability to deform the image data.

User requirements

Interviews were conducted with two surgeons to assess their current methods, their priorities, and what they viewed as areas for additional technological assistance. The information gathered was then used to develop requirements for the ideal interface for pre-surgical planning. This ideal interface for pre-surgical planning would allow the physicians to quickly and easily explore datasets without needing to set parameters, use familiar language in the interface, and be able to run on their current machines.

Research issues

Several research issues were identified from the literature review. These issues are as follows:

1. Determine the factors critical for fast and accurate tissue deformation analysis of digital medical image software using ubiquitous computing hardware

Most tissue deformation methods either use time-consuming finite-element analysis or models that do not account for the true physical behavior of tissue under forces. Simplified deformation algorithms that still utilize the known properties of tissue would allow for an accurate deformation while minimizing computing overhead, but parameter setting for these algorithms is currently experimental. An analysis of the influence of parameters on the speed and accuracy of deformations would improve the resultant deformations while decreasing computation time.

2. To design a usable and intuitive user interface for examination and deformation of digital medical image data for surgical planning

Existing medical image software has a highly complicated interface, which creates a barrier to the physicians' ability to easily obtain accurate data for consideration in their treatment plan. A streamlined software program would allow the physicians to quickly adjust parameters and create a greater understanding of their patient's situation with a minimum of confusion.

METHODOLOGY

As mentioned, discussions with surgeons emphasized the need for the utilization of real patient data, rather than artistic models, in surgical planning assistance software. This requires utilizing a method that minimizes computational expense in order to efficiently operate on the vast amounts of data from an actual DICOM image. The proposed method allows users to interactively classify tissues from a DICOM image according to their general material properties. The classifications are then used in an implementation of the extended ChainMail algorithm.

A comparison of the proposed method and other ChainMail variations is shown in Table 1. Further details on the ChainMail variation selection are provided in the ChainMail variation selection section.

Table 1: ChainMail method comparison

Variation	Multiple classifications	Queue processing order	Implemented for tissue deformation	Uses DICOM data
Original ChainMail	No	FIFO	Yes	Yes
Enhanced ChainMail	Yes	Ordered queue based on magnitude of constraint violation	Yes	No
Generalized ChainMail	No	FIFO	Yes	No
Extended ChainMail	Yes	Priority queue based on classification	No	No
Proposed method	Yes	Priority queue based on classification	Yes	Yes

Inexpensive and readily accessible hardware was also important for the surgeons interviewed. For this reason, the target hardware configuration was the commodity desktop PC with mouse and keyboard. This limited the computational power and required the use of an interface easily manipulated through mouse and keyboard operations.

The final desktop software utilized several libraries. wxWidgets [53] was used to build the graphical interface. Raw DICOM image data was loaded using DCMTK [54], and visualization of the image data was accomplished using OpenGL [55]. The software operated using the algorithm shown in Figure 20.

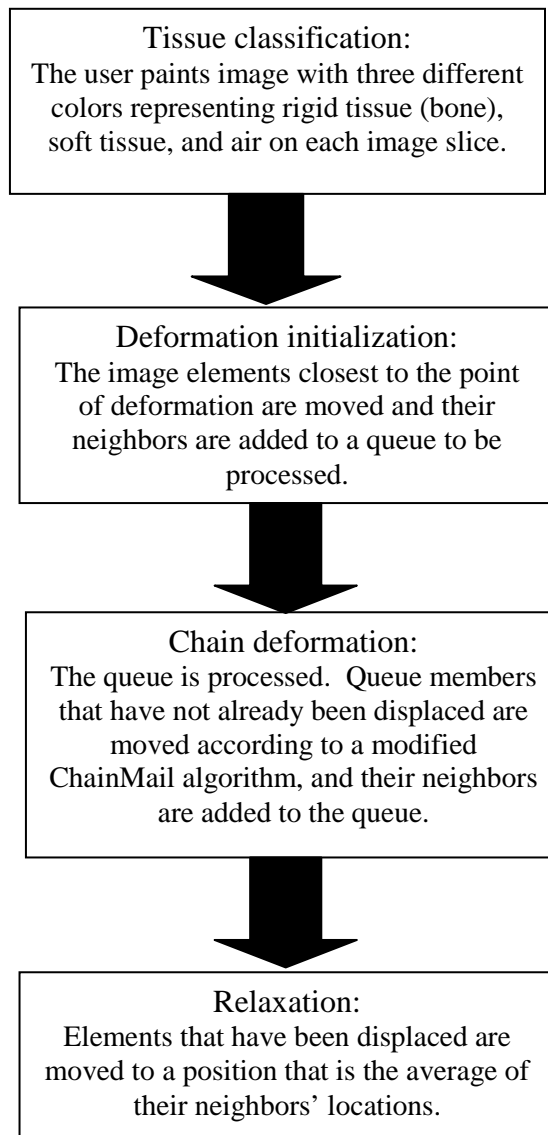


Figure 20: Proposed deformation algorithm

In this method, digital medical images are painted with classifications that represent the gross properties of the tissues. The user is then able to move individual elements of the 2D slice using mouse interactions in the GUI to create deformations at single points in the image, or by using a button to deform the entire top of the tissue in the image by a default

amount (representing uniaxial compression). As each element is moved, its neighbors are added to the priority queue, which is then processed as follows:

1. Check next point in queue for constraint violation. If there are no constraint violations, proceed to the next point in the queue.
2. If constraints have been violated, deform point according to the generalized ChainMail algorithm.
3. Add the current queue point's neighbors to a priority queue according to the extended ChainMail algorithm.

After queue processing, a relaxation step is performed to move processed elements to a position that is the average of their neighbors' positions.

ChainMail variation selection

Multiple ChainMail variations have been developed, and the variation chosen for analysis needed to be capable of approximating the physical behavior of heterogeneous tissues under stress. The original ChainMail algorithm was the first variation considered. This algorithm uses constraints seen in Figure 21. As discussed in the background, deformations are propagated through the mesh by analyzing constraint violations at various points in the mesh. MaxVertDx and maxHorizDy constrain the shear between the sponsor and its neighbors, while maxDx , minDx , maxDy , and minDy represent the maximum and minimum amount of stretch and compression. When these constraints are violated between a deformed point and its neighbors, the neighbors are moved until a violation no longer exists. In the original algorithm, these constraints were processed in FIFO order and the constraints were

assumed to be constant throughout the mesh. This allowed the algorithm to execute expeditiously, but did not accurately represent the behavior of inhomogeneous materials. The enhanced ChainMail method developed by Schill et al. was specifically designed to correct this issue [45].

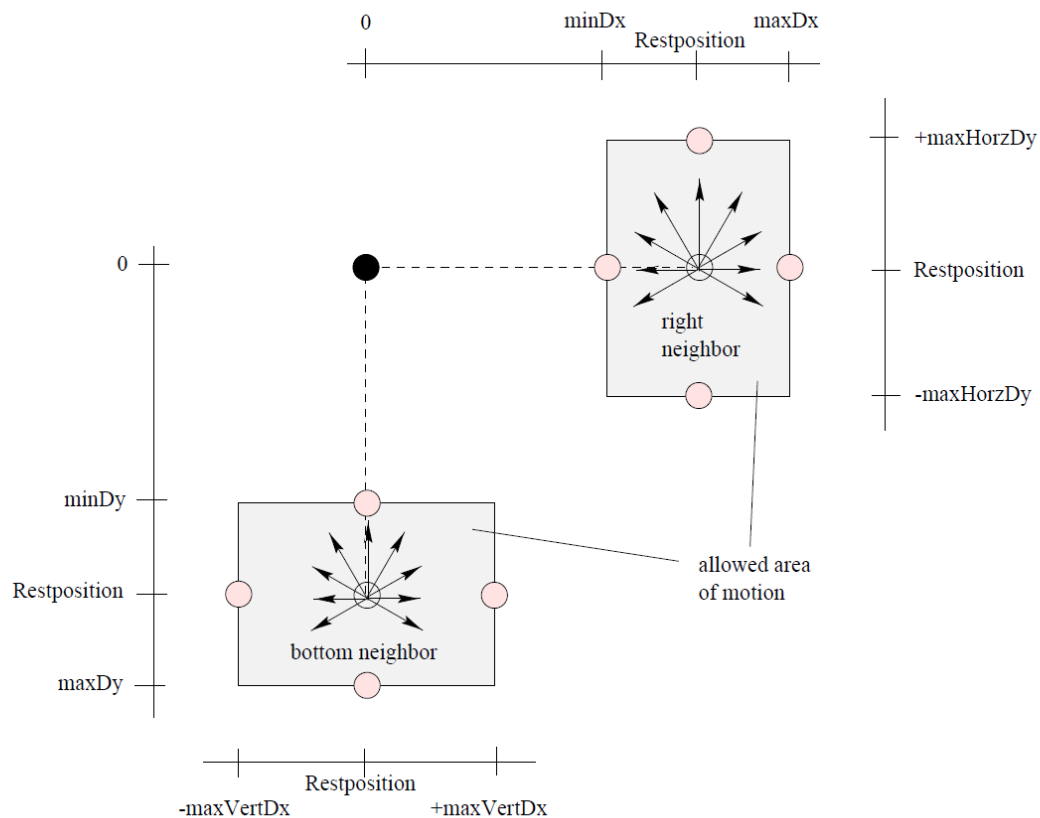


Figure 21: Schematic of ChainMail constraints between a deformed point (shown in black) and its right and bottom neighbor [43]

In contrast to the original Chainmail method's FIFO candidate point processing, the enhanced ChainMail method uses constraints of both the neighbor and the sponsor point when calculating new neighbor positions. Because the order in which candidate queue is processed influences the speed and direction of the deformation propagation, points with the

largest constraints are processed first. This method results in better representations of deformation of inhomogeneous materials, but the ordering of the candidate queue by magnitude of constraint violation and the storage of the different constraints for points represents a significant overhead.

The original and enhanced ChainMail methods both utilize the original six constraints to describe the “link” behavior of elements: $\max\text{VertDx}$ and $\max\text{HorizDy}$ constrain the shear between the sponsor and its neighbors, while $\max\text{Dx}$, $\min\text{Dx}$, $\max\text{Dy}$, and $\min\text{Dy}$ and require a rectilinear mesh. The generalized ChainMail method [56] proposed by Yi and Brodlie was designed for fast processing for web applications and extends the ChainMail method to an arbitrary mesh. For future extensibility of the proposed method, this method was chosen as the base for the algorithm development.

In the generalized ChainMail method, the parameters which describe the compression, stretch, and shearing are modified. The compression of each mesh element is parameterized by α_{min} and the stretch by α_{max} . The shearing is parameterized by β . These parameters are used to define the valid region for deformation candidate points to lie in after the movement of their sponsors. This generalized ChainMail element update is illustrated in Figure 22.

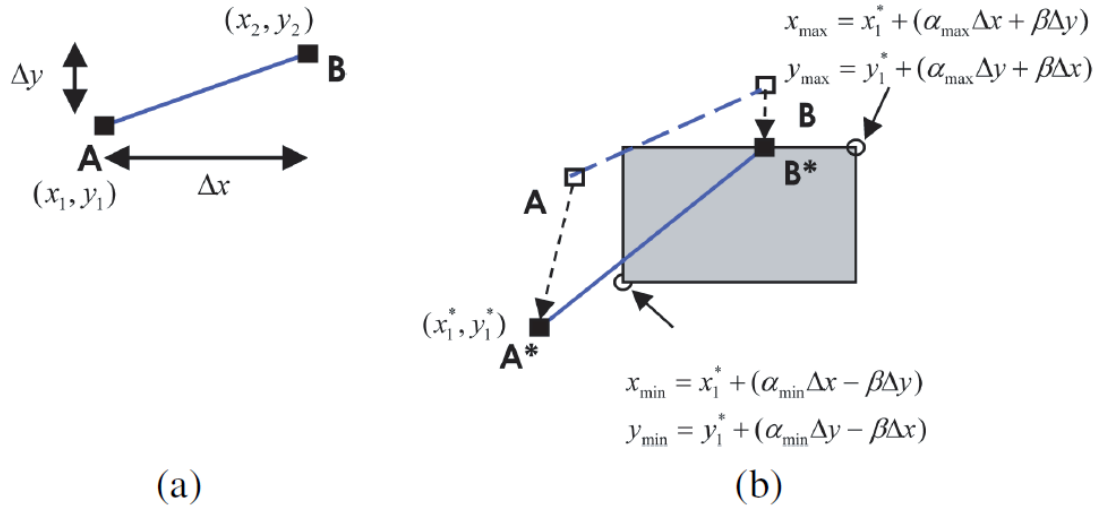


Figure 22: The original positions of elements A and B (a), and the repositioning of element B after sponsor A is moved to A*[56]

The points A and B are initially located at (x_1, y_1) and (x_2, y_2) , respectively. For the purposes of this exposition, $x_2 \geq x_1$ and $y_2 \geq y_1$. The original distance between the two points is represented as Δx and Δy in Equation 4 and Equation 5.

$$\Delta x = |x_2 - x_1| \quad \text{Equation 4}$$

$$\Delta y = |y_2 - y_1| \quad \text{Equation 5}$$

The compression, stretch, and shearing parameters and the original distance between the two points are used to define a valid region for the candidate point, B, to be located in after its sponsor, A, is moved. This region is represented in Equation 6

$$R = \{(x, y): x_{min} \leq x \leq x_{max}; y_{min} \leq y \leq y_{max}\} \quad \text{Equation 6}$$

where the region is defined by Equation 7 through Equation 10.

$$x_{min} = x_1^* + (\alpha_{min}\Delta x - \beta\Delta y) \quad \text{Equation 7}$$

$$x_{max} = x_1^* + (\alpha_{max}\Delta x + \beta\Delta y) \quad \text{Equation 8}$$

$$y_{min} = y_1^* + (\alpha_{min}\Delta y - \beta\Delta x) \quad \text{Equation 9}$$

$$y_{max} = y_1^* + (\alpha_{max}\Delta x + \beta\Delta x) \quad \text{Equation 10}$$

If the location of B lies outside of the valid region, B is moved to B*, the closest point in the valid region.

The generalized ChainMail algorithm is modified in a logical manner if the relative positions of A and B differ. For example, if $x_2 < x_1$ the region bounds x_{min} and x_{max} are as shown in Equation 11 and Equation 12.

$$x_{min} = x_1^* - (\alpha_{min}\Delta x - \beta\Delta y) \quad \text{Equation 11}$$

$$x_{max} = x_1^* - (\alpha_{max}\Delta x + \beta\Delta y) \quad \text{Equation 12}$$

The generalized ChainMail method is still not sufficient to maintain the relative positioning of elements within rigid regions of the mesh because its list of candidate elements is processed in FIFO order. A different method of ordering the deformation queue is needed.

The extended ChainMail algorithm proposed by Faeth [57] incorporates ordered processing of the candidate queue with a minimum amount of computation. In this method, used for haptic interaction with geologic surfaces, users interactively painted surfaces in order to represent the rigidity of the mesh points. Rather than comparing the degree of constraint violation or processing neighbors using FIFO, this method utilized a Standard Template Library, or STL, priority queue [58]. A STL priority queue which automatically

orders the elements from rigid to non-rigid is utilized. Although this order of processing adds complexity beyond a first-in-first-out style queue, it ensures the same relative positioning of connected regions of rigid tissues with greater processing speed than the Enhanced ChainMail algorithm. This priority queue processing style was utilized in the proposed method in order to maintain the relative position of rigid bone tissue through the use of a three-part classification system.

Tissue classification for ordered queue processing

Digital medical scans contain a variety of tissues of varying properties. To begin the deformation of each scan, a three-part tissue classification system was developed to represent the general properties. The classification is performed manually by the user, who “paints” colors representing various tissue designations on top of the original image: air, represented by a blue color, rigid tissue or bone, represented by a white color, and soft tissue, represented by red. These classifications are used to determine the rigidity of each element in ordering of the priority queue and in deformation calculations. This classification painting is performed on individual slices of original medical scans. The following conditions were used for each tissue classification:

1. Tissue classified as air does not propagate deformations to their neighbors. This is an idealized representation, but is user-definable.
2. The displacement of the rigid bone tissues is constant and fully transmitted to neighboring air and soft tissues.

3. For point classified as soft tissues, the magnitude of the deformation decreases as the deformation is propagated through the tissue.

Interactive tissue classification

All image elements are initially classified as air when images that have not previously been classified are brought into the deformation interface. Bone and muscle tissue classifications can be painted onto the images through the use of a paintbrush technique. Five different brush sizes are provided, as well as the three different tissue types. Different slices can be examined by use of the “previous” and “next” buttons. The mode can be switched from classification to active deformation by the use of another button. A DICOM image prior to tissue classification is seen in Figure 23. As seen in the image, the default classification of all image data is air, shown in blue. Painting the image with red and white classify the pixels as soft tissue and bone, respectively, as was done in Figure 24. These tissue classifications can be saved and loaded back into the interface at a later date.

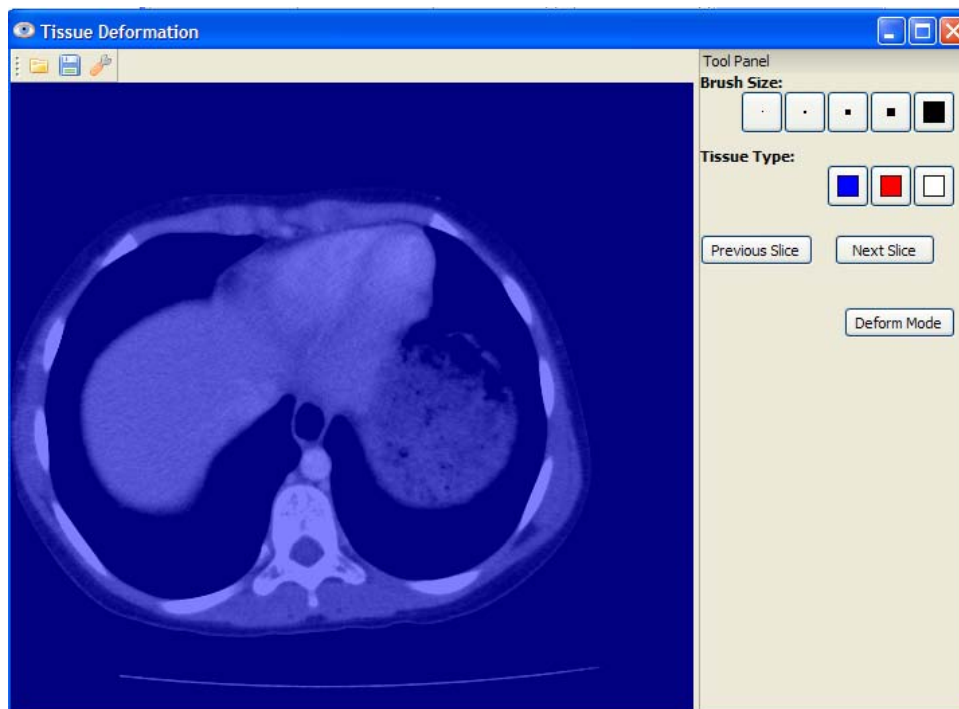


Figure 23: Loaded DICOM image in tissue classification mode

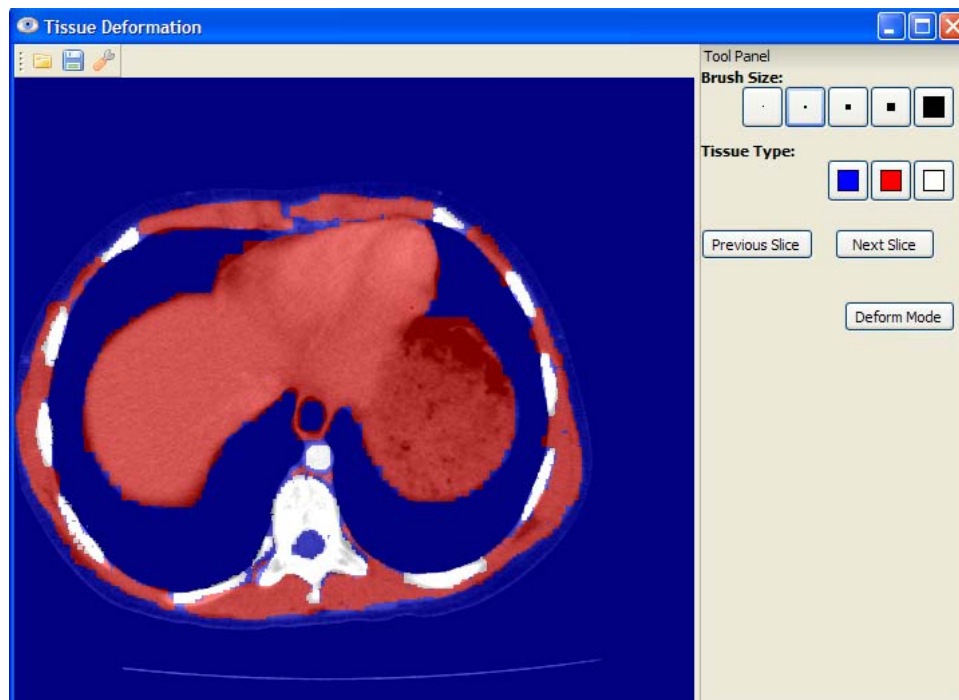


Figure 24: Classified tissue

Neighbor determination and addition to queue

The choice of neighbors influences the propagation of deformations through a mesh. The traditional ChainMail algorithm named the four closest mesh points as neighbors: top, bottom, left and right. The proposed methodology maintains an element placement that mirrors the placement of pixels in the original DICOM image. For this configuration, the closest neighbors are the eight surrounding the element, shown in Figure 25. These points are added to the processing queue from upper left to lower right.

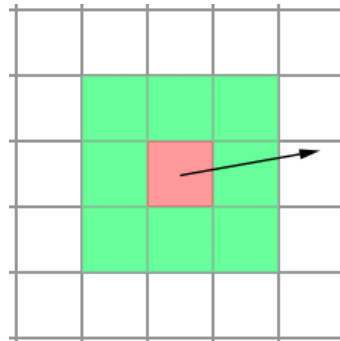


Figure 25: The sponsor point, shown in red, and its eight neighbors, shown in green

Before its addition to the queue, each neighbor is checked to ensure that it has not already been deformed by another sponsor. This is done by giving each element in the image a Boolean property that notes if it has been processed. Although this does not prevent the addition of duplicate unprocessed points to the queue, it does minimize unnecessary data storage.

Relaxation

A final relaxation step is preformed at the conclusion of queue processing. This relaxation moves the position of each of the deformed elements to the average of its

neighbors and is one of the relaxation algorithms suggested in the original ChainMail algorithm.

Element data structure

The final data structure for the each point in the deformation mesh is modified from the generalized ChainMail method. The stored parameters are as follows:

1. Element row, column, and slice- this is also the original position of the element
2. Current element position: the current x, y, and z location of the element
3. Processed state: a Boolean indicating if this point has already been deformed by a sponsor

RESULTS

The algorithm was evaluated using the uniaxial compression truth cube dataset [59] created as a basis for evaluating soft tissue simulation models. In that experiment, Kerdok et al. embedded Teflon spheres in a silicone rubber phantom which was then subjected to compression. Pre- and post-deformation CT scans were taken of the cube.

Truth cube composition

The truth cube was created with a two-part silicone rubber, RTV6166 General Electric, which behaves similar to soft tissues when in the linear range (less than 30% strain). The material also had similar properties to a mammalian liver. The total size of the cube was 8x8x8 cm.

A 7x7 matrix of commercially available 1.58mm radius spheres were positioned in the cube at intervals of 1 cm. That size of sphere was chosen in order to allow the spheres to be visible on two successive scan planes without affecting the cube's strain behavior. The spheres were composed of Teflon, which provided a high-contrast image for the CT scanner yet did not drastically alter the material behavior of the cube during testing. This was validated by conducting indentation tests samples of the pure silicone rubber, a sample with spheres placed sparsely through it, and a sample with spheres placed at 1 cm increments. The stiffness of the sample with spheres spaced most closely differed from that of the plain silicone sample by less than 2%, which was less than 4.4% of the standard deviation of the measurements on the plain sample.

The assembled truth cube was tested under uniaxial compression using the mechanism in Figure 26. The cube was first placed on a rigid base. A 2.54cm thick acrylic plate was attached to a vertical support with a low-friction joint. Loading was performed by adding weights to the side of the plate farthest from the joint.

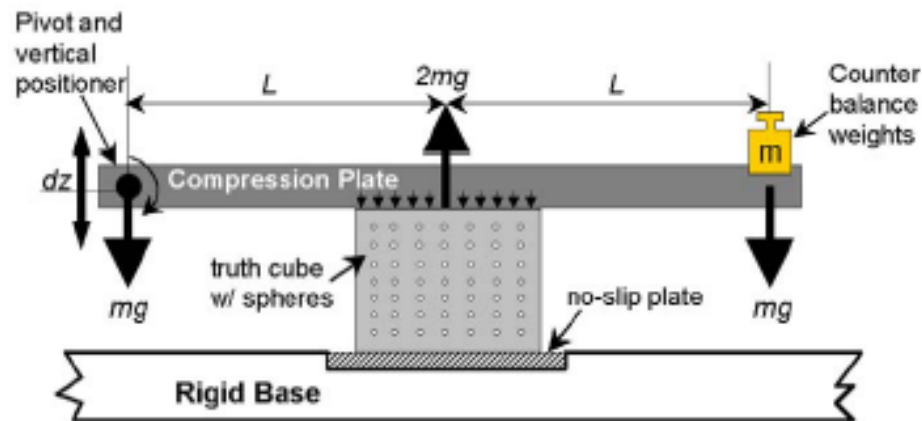


Figure 26: Schematic of truth cube uniaxial experiment, where L is the distance between the loading points to the center of the cube and m is the mass of the counter weights [59]

Three uniaxial loading conditions were performed: 4mm (5% strain), 10mm (12.5% strain), and 14.6mm (18.25% strain) displacement. CT scans were taken of the unloaded cube and of the three loading conditions using a General Electric LightSpeed CT scanner with a voxel size of 0.37x0.37x1.25mm, creating four datasets of 99 512x512 DICOM images. The 5% and 18.25% strain conditions were chosen to evaluate the proposed method. A single DICOM image was used for the evaluations. The image used is of the center vertical plane of the truth cube (the “center slice”). The original and deformed DICOM images of the center slice of the dataset under 5% strain are shown in Figure 27.

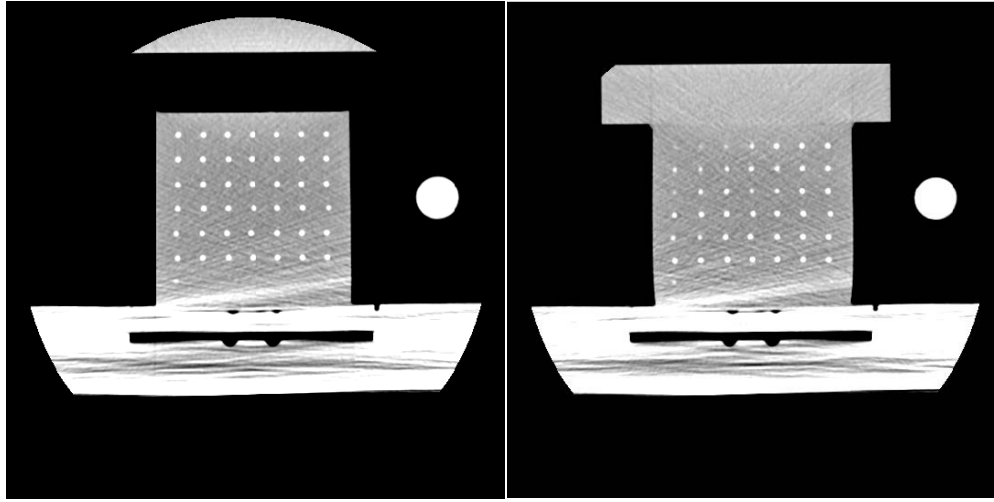


Figure 27: Truth cube undeformed center slice (left) and slice with 5% strain applied (right)

Parameter selection

Although the generalized ChainMail parameters have a foundation in material properties, experimentation is necessary in order to find accurate values. Two different sets of parameter values, shown in Table 2, were tested. Scheme A was selected after qualitative analysis of the resultant deformations for several different constraint values. A quantitative analysis comparing the predicted deformations to the measured deformations was performed. This analysis revealed excessive uniaxial displacement and not enough displacement perpendicular to the axis of strain. Constraint scheme B increased the strain constraint and utilized a basic boundary condition that did not allow the last row of elements or their neighbors in the classified tissue to be displaced. This was done to provide a more accurate representation of the tissue properties than scheme A.

Table 2: Constraint schemes

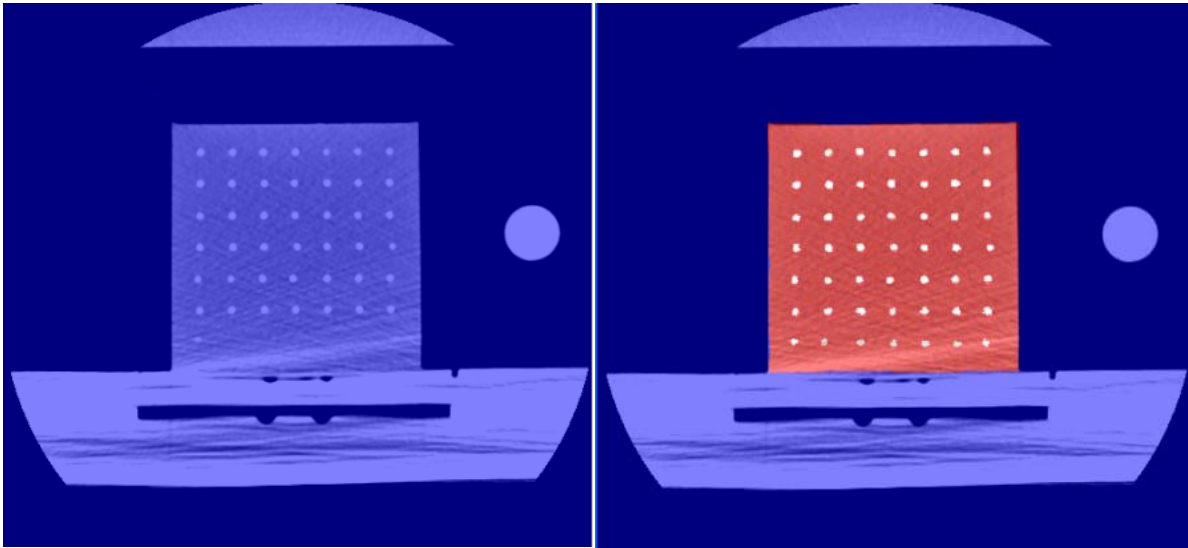
Tissue Classification	Constraint	Scheme	
		A	B
Soft tissue	α_{\min}	0.98	0.98
Soft tissue	α_{\max}	1.05	1.05
Soft tissue	β	0.01	0.03
Rigid tissue	α_{\min}	1.0	1.0
Rigid tissue	α_{\max}	1.0	1.0
Rigid tissue	β	0.0	0.0

Single DICOM slice truth cube deformation

The algorithm was partially evaluated using a manually classified DICOM image of the center slice and scheme B. The results of this are shown in Figure 28. The classifications for tissue and the deformations are drawn on top of the grayscale DICOM image. The original DICOM image remains visible through the tissue classifications for comparison with the deformed image.

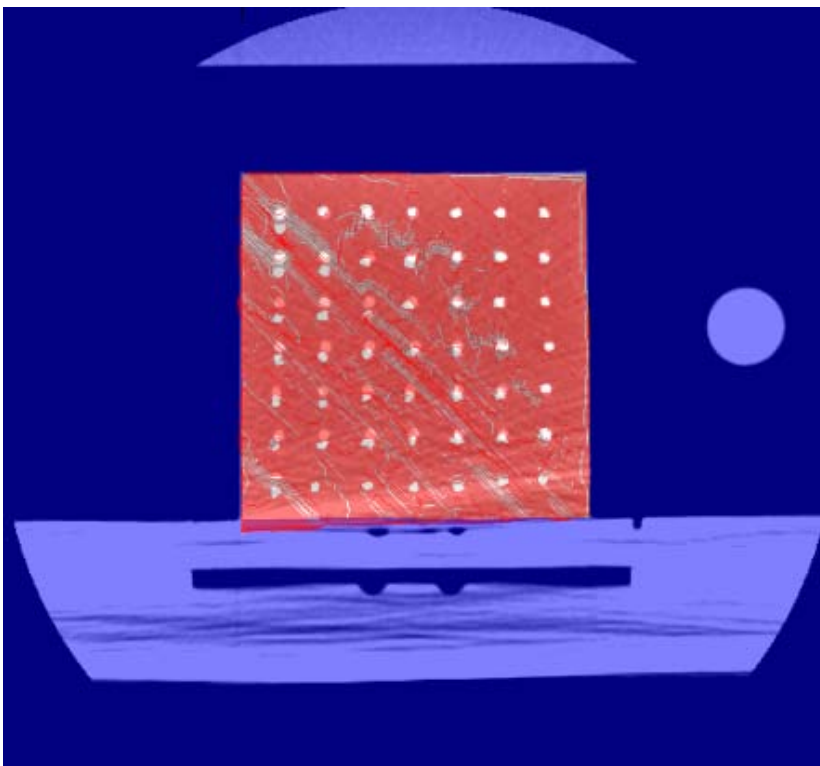
Figure 28 (a) shows the original classification for the image when it was first loaded. The default classification is automatically air, shown in blue over the grayscale image. The truth cube was then manually classified as soft tissue (red) and bone/rigid tissue (white) as shown in Figure 28 (b). This classification was performed using the interface discussed in the previous chapter in under five minutes. 5% strain was then applied to the classified

tissue. The resultant deformation is shown in Figure 28 (c). The regions of the image classified as air have not been deformed by the algorithm. The rigid and soft tissues have been displaced, and the rigid spheres have moved their relative position yet maintained their shape during the deformation, which is the expected result for priority queue processing of deformation points. The soft tissue, represented in red, displaces more relative to its neighbors and “cracks” appear where the gray cube of the original DICOM image in the background shows between the classified tissues. This apparent cracking is a result of the widening space between the soft tissue neighbors and the visualization of the continuous material using discrete elements. The large gaps between the soft tissue elements are a result of the ChainMail algorithm; each deformed neighbor is only analyzed for constraint violation with its sponsor point, not all surrounding points. This allows a classified tissue element to move far from its neighbors as long as it is still within the bounding box defined by its constraints and its sponsor point’s position.



(a)

(b)



(c)

Figure 28: Center slice image default air classification (a), classified as soft tissue and bone (b), deformed with 5% strain using scheme B (c)

Deformation application and queue building

As can be seen in the deformed image in Figure 28 (c), the solution predicts less deformation in the upper right corner as opposed to the upper left. It is hypothesized that this is a result of the order in which elements are processed during deformations. In the uniaxial deformations tested, deformation was applied from left to right on the top row of tissue elements, and each of those elements' neighbors were added to the priority queue from upper left to lower right. Although the priority queue standard does not require FIFO processing of elements with the same priority, the order of addition to the queue resulted in the sponsor points generally being those to the above and to the left of the candidate for deformation. It was the displacement of those sponsor points that then determined the valid region for the deformation candidate point.

Ideal image generation

An ideal segmentation scenario was created from the pre-deformed center slice information from the truth cube experiments. The ideal slice used the known dimensions of the cube, the radius of the spheres, and the center positions of the spheres before compression to create an image. The silicone cube was then classified as soft tissue and the spheres were classified as bone.

To evaluate the influence of image size on the accuracy and performance of the algorithm, datasets were created using seven different element spacing values. The element spacing from the original image was 0.37 mm. This original spacing was used to generate

coarser datasets that ranged from 2 mm to 0.44 mm element spacing. A summary of the generated datasets is shown in Table 3.

Table 3: Generated ideal segmentation data sets

Element Spacing (mm)	Elements per Row and Column	Total Elements
2.00	40	1600
1.33	60	3600
1.00	80	6400
0.80	100	10000
0.57	140	19600
0.44	180	32400
0.37	216	46656

Displacement error results

The predicted and actual displacements of the center of the spheres in the dataset's center slice were compared for the 5% and 18.25% uniaxial strains order to examine the performance of the method. The accuracy of the method was evaluated using the root mean squared error analysis performed by the truth cube experimenters [59]. The average, minimum, and maximum root mean squared error for intra-slice displacement was found for the sphere center locations. These are shown in Tables 4-6. For comparison, the root mean squared error resultant from a FEA analysis performed on the center slice data by the truth cube experimenters for 18.25% strain is shown in Table 7.

A comparison of the results for constraint schemes A and B in Table 4 and 5 reveals that, while constraint scheme B provided a decreased average and minimum root mean squared error for most element spacings, it also resulted in greater maximum errors.

The algorithm was also run with constraint scheme B for 18.25% deformation in order to compare its accuracy under larger strains to that of an FEA performed in the truth cube experiment. Examination of the ChainMail deformation results in Table 6 with those of the FEA in Table 7 reveals far greater errors for the ChainMail deformation, with average RMS errors greater than 8 ½ times that of the FEA average RMS error and minimum of more than 40 times larger than that of the FEA. This reveals that the algorithm still needs to be improved in order to model large displacements with comparable accuracy to FEA.

Table 4: Root mean square error for 5% strain, constraint scheme A

Number of Elements	Element Spacing (mm)	Strain (%)	Constraint Scheme	RMS _{ave} (mm)	RMS _{min} (mm)	RMS _{max} (mm)
1600	2	5	A	2.123	0.032	4.920
3600	1.3	5	A	1.807	0.194	3.543
6400	1	5	A	1.821	0.182	3.422
10000	0.8	5	A	1.919	0.342	3.806
19600	0.57	5	A	1.790	0.534	3.426
32400	0.44	5	A	1.678	0.180	3.429
46656	0.37	5	A	1.692	0.206	3.450

Table 5: Root mean square error for 5% strain, constraint scheme B

Number of Elements	Element Spacing (mm)	Strain (%)	Constraint Scheme	RMS _{ave} (mm)	RMS _{min} (mm)	RMS _{max} (mm)
1600	2	5	B	1.838	0.100	4.164
3600	1.3	5	B	1.589	0.116	4.075
6400	1	5	B	1.616	0.233	3.879
10000	0.8	5	B	1.460	0.143	3.641
19600	0.57	5	B	1.883	0.128	4.274
32400	0.44	5	B	1.666	0.104	4.182
46656	0.37	5	B	1.612	0.155	4.519

Table 6: Root mean square error for 18.25% strain, constraint scheme B

Number of Elements	Element Spacing (mm)	Strain (%)	Constraint Scheme	RMS _{ave} (mm)	RMS _{min} (mm)	RMS _{max} (mm)
1600	2	18.25	B	10.405	6.675	14.365
3600	1.3	18.25	B	10.659	6.747	14.074
6400	1	18.25	B	10.071	6.697	13.635

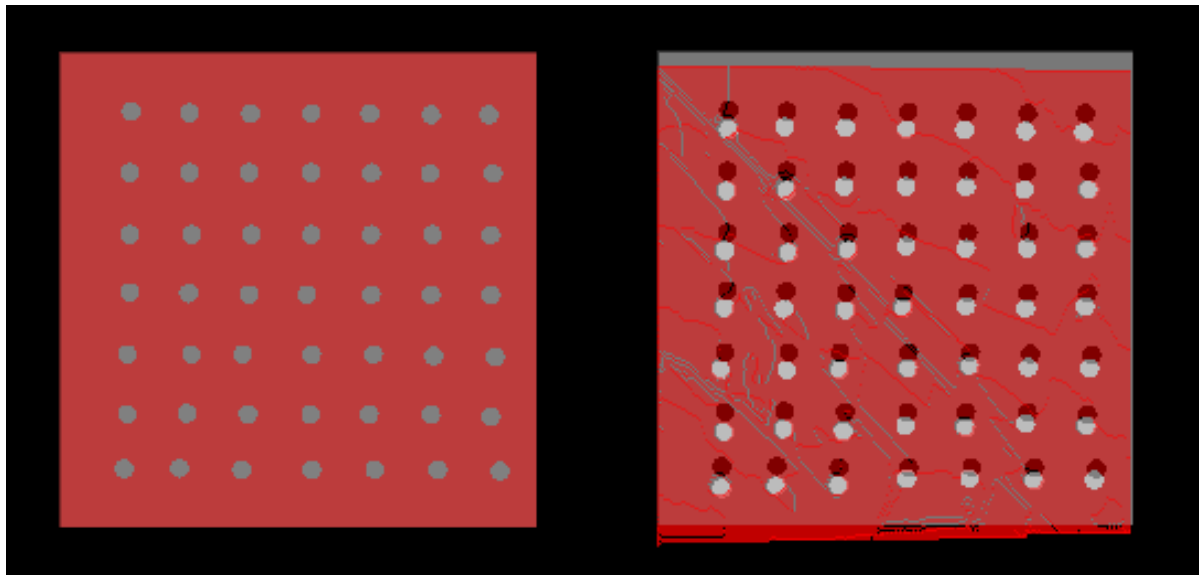
Table 7: Root mean square error for 18.25% strain for truth cube FEA analysis

Element Spacing (mm)	Strain (%)	RMS _{ave} (mm)	RMS _{min} (mm)	RMS _{max} (mm)
0.37	18.25	1.174	0.159	2.523

The undeformed ideal 0.37 mm element spacing image and the deformations resultant from 5% strain is shown in Figure 29. Deformations for other ideal image sizes were similar. As in the visualization of the truth cube slice in Figure 28, the original ideal slice is drawn behind the partially transparent tissue classifications. Once again, soft tissues classifications are shown in red and bone classifications are shown in white. The ideal images did not include any air classifications.

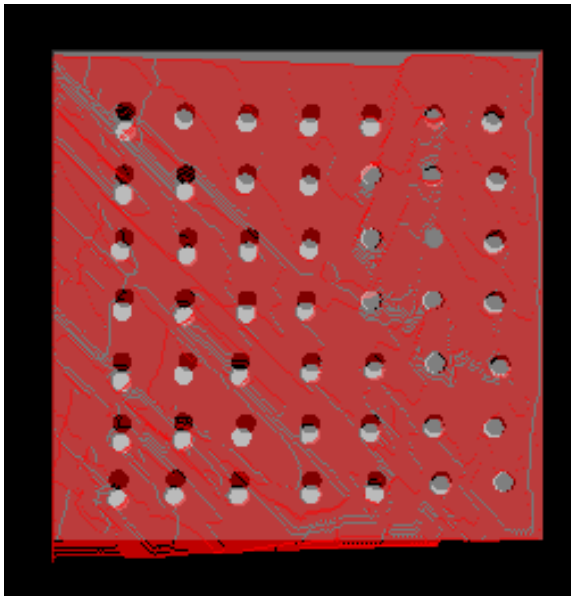
The general behavior of the classified tissues in the ideal slices was similar to that of the classified DICOM image in Figure 28. The elements classified as bone displaced while maintaining their same relative positioning to neighboring “bone” elements, while the soft tissues showed greater displacement relative to their neighbors as seen by the “cracked” appearance of the final deformation. Overlapping soft tissues appear as red lines bright red lines.

Comparing results for the A and B parameters show that although constraint scheme B results in a smaller average displacement error of the sphere centers when compared numerically, both schema result in a similar deformation when compared qualitatively. In addition to large displacement between some soft tissue elements, both schemes fail to accurately represent the boundary conditions and the deformed tissues move down to below the cube base rather than bow on the sides like the real cube.



(a)

(b)



(c)

Figure 29: Element spacing 0.37 mm undeformed ideal slice (a), deformed with A constraints (b), deformed with B constraints (c)

Single slice processing time

The time to deform a single slice was monitored for parameter set A when the algorithm was run on a on a Windows XP Pro desktop with 4GB of RAM and a Core2 X6800 CPU using ideal images and constraint scheme A for 5% strain. The results are shown in Figure 30. As the number of elements in the image increased, the computation time increased at a greater than linear pace. The processing time ranged from less than 15 seconds for the coarsest spacing (2mm) to over 3 ½ hours for the same element spacing as in the original DICOM image (0.37 mm).

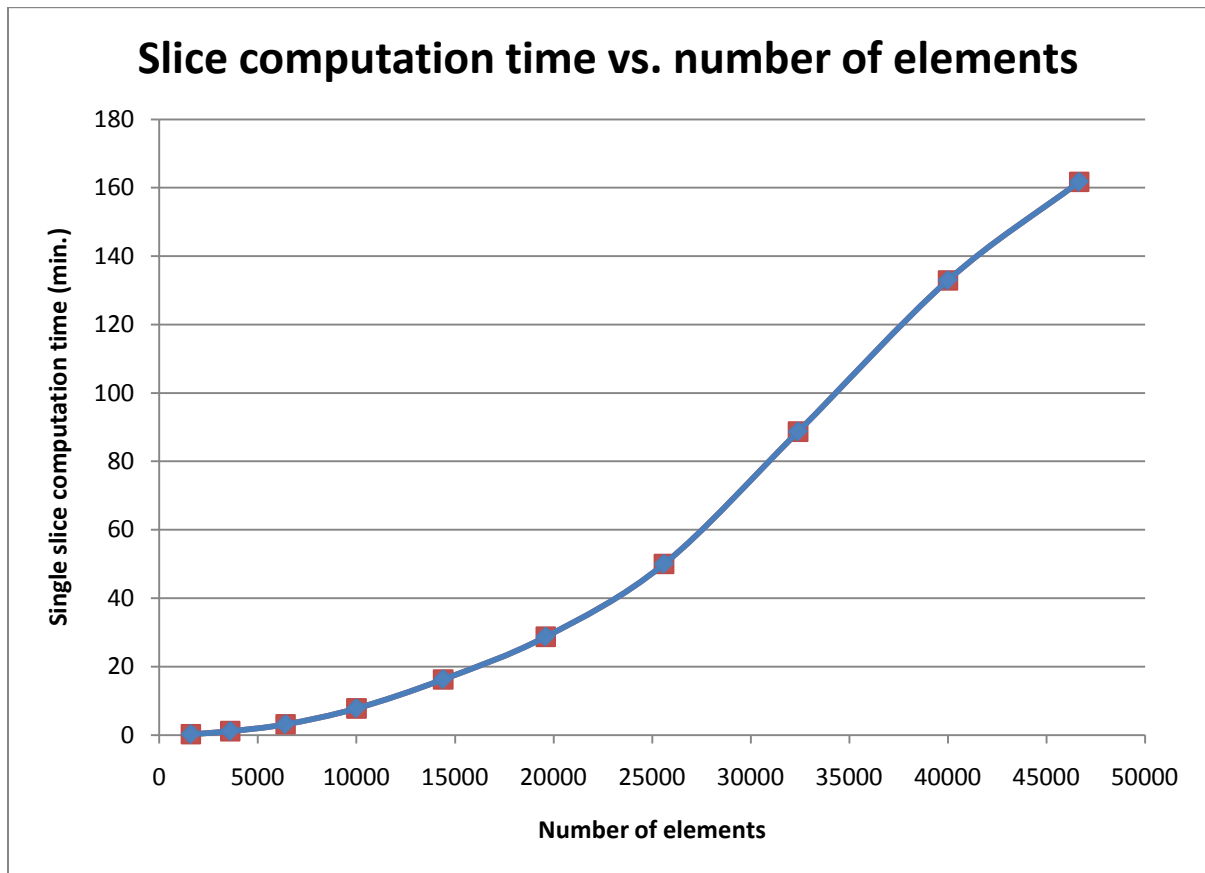


Figure 30: Single slice processing times

Although the computation time for full-resolution ideal images were still far from real-time, decreasing the number of elements greatly increases the process speed by reducing the length of the priority queue. The qualitative and quantitative analysis of the deformations also revealed that the elements were being deformed excessively. Modifying the algorithm to allow for more flexibility of the soft tissues would be expected to both increase the accuracy of the deformations and reduce the size of the queue by only moving elements that will truly be deformed.

CONCLUSIONS

Summary and conclusions

A new software application was created which enabled users to interactively deform medical images through the use of a generalized ChainMail algorithm with a priority queue based on tissue classifications. The algorithm was evaluated for computation time and accuracy for two constraint schemes and seven image fidelities.

The difference in RMS error between the two schemes suggests that, with finer parameter tuning, the accuracy of the displacement could be increased.

Although the single slice computation time was more than three hours for the highest fidelity ideal data set, the drastic time decrease for decreasing element numbers suggests that decreasing the number of elements might provide better computation time with similar accuracy. Because the deformation algorithm also depends on the constraints to evaluate which elements should be deformed, better parameter tuning also holds promise in minimizing computation time by decreasing queue size.

Future work

Currently, a straightforward conversion from material properties to ChainMail constraints does not exist. A thorough analysis of the tie between ChainMail constraints and material properties, leading to a development of guidelines for parameter selection, would allow more accurate displacement predictions within the ChainMail algorithm. This may also decrease computation time by eliminating unnecessary deformations caused by unrealistic parameters.

Another limitation to the current method is the reliance on manual user tissue classification. This represents a significant time investment when analyzing entire images. Investigation and implementation of image segmentation algorithms would increase the usability of the software by automating the most intensive task.

Analysis can also be performed to determine the minimum number of elements to accurately represent a high-fidelity image. Decreasing the number of elements in the mesh and tying the elements' deformations to a texture has the possibility to decrease computation speed while improving the image quality of the visualization.

As the execution time is reduced by modifying parameters, automating segmentation, and reducing the mesh size, it would be appropriate to expand the algorithm to represent the interactions between slices in addition to the current intra-slice interactions.

The accuracy of the method may also be improved by increasing the number of tissue classifications to better represent the variations in properties of soft tissues.

Finally, the method will be modified to include more accurate boundary conditions. This will ensure better displacement propagation through the tissues.

REFERENCES

- [1] Dhawan, A. P., "Medical Imaging Modalities," *Medical Imaging Analysis*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2003, pp. 53-109.
- [2] Beutel, J., Kundel, H. L., and Van Metter, R.L., "X-ray Production, Interaction, and Detection in Diagnostic Imaging," *Handbook of Medical Imaging*, 1st ed., SPIE Press, Bellingham, 2000, pp. 3-74.
- [3] Imaginis, "Brief History of the CT," 2009, [<http://www.imaginis.com/ct-scan/history.asp>. Accessed 8/5/09.]
- [4] Hornak, J. P., "The Basics of MRI", 2008, [<http://www.cis.rit.edu/htbooks/mri/inside.htm>. Accessed 1/11/09.]
- [5] "The DICOM Standard", *DICOM Digital Imaging and Communications in Medicine*, January 2008, [www.medical.nema.org/Dicom. Accessed 7/4/2009.]
- [6] Hopper, K. D., Tunç lyriboz, A., Wise, S.W., Neuman, J. D., Mauger, D. T., & Kasales, C. J., "Mucosal Detail at CT Virtual Reality: Surface versus Volume Rendering," *Radiology*, Vol. 214, No. 2, 2000, pp. 517-522.
- [7] Cotin, S., Delingette, H., and Ayache, N., "Real Time Elastic Deformations of Soft Tissues for Surgery Simulation," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 5, No. 1, 1996, pp. 535-540.
- [8] Al-Khlaifah, and A. Roberts, D., "Survey of Modelling Approaches for Medical Simulators," *Proceedings of the 5th Conference on Disability, Virtual Reality, and Associated Technology*, Oxford, UK, 2004, pp. 321-329.
- [9] "Knee Arthroscopy Simulation using Volumetric Knee Models," Mitsubishi Electric Research Libraries, 2007, [<http://www.merl.com/projects/kneesystem2/> . Accessed 8/5/09.]
- [10] Radetzky, A., Rudolf, M., Stefanon, W., Starkie, S., Davies, B., and Auer, L. M., "Simulating Minimally Invasive Neurosurgical Interventions Using an Active Manipulator," *Proceedings of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2000, pp. 578-587.
- [11] Radetzky, A., and Nurnberger, A., "Visualization and Simulation Techniques for Surgical Simulators using Actual Patient's Data", *Artificial Intelligence in Medicine*, Vol. 26, No. 3, 2002, pp. 255-279.
- [12] Reinig, K. D., Rush, C. G., Pelster, H. L., Spitzer, V. M., and Heath, J. A., "Real-Time Visually and Haptically Accurate Surgical Simulation," *Studies in Health Technology and Informatics*, Vol. 29, pp. 542-545.
- [13] "The Visible Human Project," February 2003, [http://www.nlm.nih.gov/research/visible/visible_human.html. Accessed 8/5/2009.]
- [14] Immersion Corporation, "Medical and Surgical Simulators," 2009, [<http://www.immersion.com/markets/medical/products/index.html>. Accessed 8/5/09.]
- [15] Kühnapfel, U., Çakmak, H., and Maaß, H., "Endoscopic Surgery Training using Virtual Reality and Deformable Tissue Simulation," *Computers and Graphics*, Vol. 25, No. 5, 2000, pp. 671-682.

- [16] Biesler, D., and Gross, M. H., "Open Surgrey Simluation," *Proceedings of Medicine Meets Virtual Reality*, 2002, pp. 57-63.
- [17] Webster, R. W., Zimmerman, D. I., Mohler, B. J., Melkonian, M. G., and Haluck, R. S., "A prototype haptic suturing simulator," *Medicine Meets Virtual Reality 2001*, 2001, pp. 567-569.
- [18] Ursine, M., Tasto, J. L., Nguyen, B. H., Cunningham, R., and Merrill, G. L., "CathSim: An Intravascular Catherization Simulator on a PC," *Medicine Meets Virtual Reality 1999*, 1999, pp. 360-372.
- [19] Liu, A., Tendick, F., Cleary, K., and Kaufmann, C., "A Survey of Surgical Simulation: Applicatoin, Technology, and Education," *Presence*, Vol. 12, No. 6, 2003, pp. 599-614.
- [20] Bro-Nielsen, M., "Modeling Elasticity In Solids Using Active Cubes- Appliction to Simulated Operations," *Proceedings of the First International Conference on Computer Vision, Virtual Reality, and Robotics in Medicine*, 1995, pp. 535-541.
- [21] Cotin, S., Delingette, H., and Ayache, N., "Real Time Elastic Deformations of Soft Tissues for Surgery Simulation," *IEEE Transactions on Visualizaton and Comptuer Graphics*, Vol. 5, No. 1, 1996, pp. 535-540.
- [22] Promayon, E., Baconnier, P., and Puech, C., "Physically-Based Model for Simulating the Human Trunk Respiration Movements," *Proceedings of the First Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine and Medial Robotics and Comptuer-Assisted Surgery*, 1997, pp. 379-388.
- [23] Terzopoiolos, D., and Fleischer, K., "Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture," *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, 1988, pp. 269-278.
- [24] Sagar, M. A., Bullivant, D., Mallinson, G. D., and Hunter, P. J., "A Virtual Environment and Model of the Eye for Surgical Simulation," *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, 1994, pp. 205-212.
- [25] Zienkiewicz, O. C., *The Finite Element Method*, McGraw-Hill, New York, NY, 1977.
- [26] Székely, G., Brechbuhler, R., Hutter, R., Rhomberg, A., Ironmonger, N., and Schmid, P., "Modelling of Soft Tissue Deformation for Laparoscopic Surgery Simulation," *Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention*, 1998, pp. 57-66.
- [27] Paulsen, K. D., Miga, M. I., Kennedy, F. E., Hoopes, P. J., and Hartov, A., "A Computational Model for Tracking Subsurface Tissue Deformation During Stereotactic Neurosurgery," *IEEE Transactions on Biomedical Engineering*, Vol. 46, No. 2, 1999, pp. 231-225.
- [28] Wittek, A., Miller, K., Kikinis, R., and Warfield, S. K., "Patient-Specific Model of Brain Deformations: Application to Medical Image Registration," *Journal of Biomechanics*, Vol. 40, No. 4, 2007, pp. 919-929.
- [29] Bro-Nielsen, M., "Surgery Simluation Using Fast Finite Elements," *Proceedings of the 4th Interanational Conference on Visualization in Biomedical Computing*, 1996, pp. 529-534.

- [30] Coquillat, S., and Jancéne, P., “Animated Free-form Deformations: An Interactive Animation Technique,” *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH, 1991, pp. 187-196.
- [31] Ueier, U., Lopez, O., and Monserrat, C. J., “Real-Time Volumetric Deformable Models for Surgery Simulation Using Finite Elements and Condensation,” *Computer Graphics Forum*, Vol. 15, No. 3, pp. 57-66.
- [32] Wu, W., and Pheng, A. “An Improved Scheme of an Interactive Finite Element Model for 3D Soft-Tissue Cutting and Deformation,” *The Visual Computer*, Vol. 21, No. 8, 2005, pp. 707-716.
- [33] Taylor, Z., Cheng, M., and Ourselin, S., “High-Speed Nonlinear Finite Element Analysis for Surgical Simulation Using Graphics Processing Units”, *IEEE Transactions on Medical Imaging*, Vol. 27. No. 5, 2008, pp. 650-663.
- [34] Kenedi, R. M., Gibson, T., Evans, J. H., and Barbenel, J. C., “Tissue Mechanics,” *Physics in Medicine and Biology*, 1975, Vol. 20, No. 3, pp. 699-717.
- [35] Keeve, E., Girod, S., Kikinis, R., and Girod, B., “Deformable Modeling of Facial Tissue For Carniofacial Surgery Simluation,” *Computer Aided Surgery*, Vol. 3, No. 5, 1998, pp. 541-546.
- [36] Gibson, S. F., and Mirtich, B., “A Survey of Deformable Modeling in Computer Graphics”, Mitsubishi Electric Research Laboratories, Cambridge, MA, 1997.
- [37] Brown, J., Sorkin, S., Bruyns, C., Latombe, J. C., Montgomery, K., and Stephanides, M., “Real Time Simluation of Deformable Objects: Tools and Applications,” *Computer Animation 2001. The Fourteenth Conference on Computer Animation. Proceedings*, Seoul, South Korea, 2001, pp. 228-258.
- [38] Georgii, J., and Westermann, R., “Mass-spring Systems on the GPU”, *Simulation Modelling Practice and Theory*, Vol. 13, No. 8, pp. 693-702.
- [39] Sederberg, T., and Parry, S., “Free-Form Deformation of Solid Geometric Models,” *ACM SIGGRAPH Computer Graphics*, Vol. 4, no. 20, 1986, pp. 151-160.
- [40] Chua, C., and Neumann, U., “Hardware-Accelerated Free-Form Deformation,” *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware*, ACM, New York, 2000, pp. 33-39.
- [41] Rezk-Salama, C., Scheuering, M., Soza, G., and Greiner, G., “Fast Volumetric Deformation on General Purpose Hardware,” *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware*, Los Angeles, CA, 2001, pp. 17-24.
- [42] Correa, C. D., Silver, D., and Chen, M. “Volume deformation via scattered data interpolation,” *Volume Graphics*, 2007, pp. 9-16.
- [43] Gibson, S. F., “3D ChainMail: A Fast Algorithm for Deforming Volumetric Objects,” *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, SIGGRAPH, Providence, Rhode Island, 1997, pp. 149-154.
- [44] Le Fol, T., Acosta-Tamayo, O., Lucas, A., and Harignon, P., “Angioplasty Simulation using ChainMail Method,” *Medical Imaging 2007: Visualization and Image-Guided Procedures*, San Diego, CA, 2007.
- [45] Schill, M. A., Gibson, S. F., Bender, H. J., and Manner, R., “Biomechanical Simulation of the Vitreous Humor in the Eye Using an Enhanced ChainMail

- Algorithm,” *Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention*, 1998, pp. 679-687.
- [46] Plishker, W., and Shekar, R., “Virtual Colonoscopy Registration with Global Chainmail,” 2008.
- [47] Visage Imaging, “Amira,” 2008, [<http://amiravis.com>. Accessed 11/12/2009.]
- [48] Able Software Corporation: “3D-DOCTOR,” [<http://www.3d-doctor.com/>. Accessed 11/1/09.]
- [49] Kitware, Inc., “VolView”, [<http://www.kitware.com/products/volview.html>. Accessed 8/4/09.]
- [50] Rosset, A., “OsiriX Imaging Software,” [<http://www.osirix-viewer.com>. Accessed 8/4/09.]
- [51] Visual Medical Solutions, “BodyViz,” [<http://www.bodyviz.com/>. Accessed 11/16/2009.]
- [52] Foo, J. L., Martinez, M., Peloquin, C., Lobe, T., and Winer, E., “A Collaborative Interaction and Visualization Multi-Modal Environment for Surgical Planning,” *Proceedings of the 17th Annual Medicine Meets Virtual Reality*, Long Beach, CA, 2009, pp. 97-102.
- [53] “wxWidgets,” [www.wxwidgets.org. Accessed 1/11/09.]
- [54] DICOM@OFFIS , “DCMTK - DICOM Toolkit,” [<http://dicom.offis.de/dcmTk>. Accessed 1/10/2009].
- [55] SGI, “OpenGL”, [www.opengl.org. Accessed 1/11/09.]
- [56] Li, Y., and Brodlie, K., “Soft Object Modelling with Generalized ChainMail—Extending the Boundaries of Web-Based Graphics,” *Computer Graphics Forum*, Vol. 22, No. 4, 2003, pp. 717-727.
- [57] Faeth, A. J., “Expressive Cutting, Deforming, and Painting of Three-Dimensional Digital Shapes Through Asymmetric Bimanual Haptic Manipulation,” M.S. Thesis, Iowa State University, Ames, Iowa, 2009.
- [58] ISO/IEC, “Programming Languages: C++”, American National Standards Institute, pp. 801-481.
- [59] Kerdok, A. E., Cotin, S. M., Ottensmeyer, M. P., Galea, A. M., Howe, R. D., and Dawson, S. L., “Truth Cube: Establishing Physical Standards for Soft Tissue Simulation,” *Medical Image Analysis*, Vol. 7, No. 3, 2003, pp. 293-291.