Summer 2010

Hexahedral meshing of subject-specific anatomic structures using registered building blocks

Amla Natarajan
University of Iowa

HEXAHEDRAL MESHING OF SUBJECT-SPECIFIC ANATOMIC STRUCTURES USING REGISTERED BUILDING BLOCKS

by

Amla Natarajan

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

July 2010

Thesis Supervisors: Associate Professor Nicole M. Grosland
Associate Professor Vincent A. Magnotta

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

Amla Natarajan

has been approved by the Examining Committee
for the thesis requirement for the Master of Science
degree in Biomedical Engineering at the July 2010 graduation.

Thesis Committee: _____
Nicole M. Grosland, Thesis Supervisor

_____
Vincent A. Magnotta, Thesis Supervisor

_____
Tae-Hong Lim

_____
David G. Wilder

To my parents and brother, for loving me and believing in me, no matter what

Success is not final, failure is not fatal: it is the courage to continue that counts.
Winston Churchill

ACKNOWLEDGMENTS

First of all, I would like to express my heartfelt gratitude towards my academic and research advisor, Dr. Nicole Grosland, for teaching me so much through the last two years and for being eternally patient with me. I would also like to thank my research advisor Dr. Vincent Magnotta, for teaching me everything that made working on this project possible and for always finding time to answer my questions.

My sincere thanks go to Dr. Kiran Shivanna, for helping me so much through the entire course of my research. I would like to thank all my lab-mates, for assisting me when needed in research and for taking such good care of me, the last two years.

Also, thanks to Dr. Lim and Dr. Wilder for serving on my committee and making learning Biomechanics fun.

TABLE OF CONTENTS

LIST OF TABLES vi

LIST OF FIGURES vii

CHAPTER

I. INTRODUCTION 1

II. LITERATURE REVIEW 5

Finite Element Method 5
Mesh Generation 7
IA-FEMesh 11
Deformable Image Registration 14
k-D Tree Search Structures 17

III. MATERIALS & METHODS 20

Building Block Mapping Algorithm 20
Software 25
Evaluation of the Algorithm 25
Proximal Phalanx Bone 26
Proximal to Distal Phalanx Bone 27
Femur 28
C5 Posterior Vertebra 29
Evaluation Metrics 30

IV. RESULTS 32

Results of Evaluation 32
Proximal Phalanx Bone 32
Proximal to Distal Phalanx Bone 33
Femur 34
C5 Posterior Vertebra 35
Summary 36

V. DISCUSSION 38

BIBLIOGRAPHY 41

LIST OF TABLES

Table

1. A Comparison of the Quality of Meshes Obtained by Meshing the Target Surface Using Manually-Defined and Mapped Building Block Structures 37

LIST OF FIGURES

Figure

1. Mesh generation using IA-FEMesh
(a) Automatic block definition,
(b) & (c) Manual realignment of vertices,
(d) Mesh seeding of the block and
(e) Closest point projection onto surface to create the mesh 13

2. Non-rigid registration being applied to inter-subject brain matching of two selected slices from a T1weighted MR-volume 15

3. Mechanism of k-D tree nearest neighbor search. While Q is the query (search) point, and N is the current nearest neighbor, if dP<dN, the actual nearest neighbor may lie on the right of the partition plane. 19

4. Building block structure mapping algorithm. 24

5. Inputs for mapping between proximal phalanx bones:
(a) Proximal phalanx bone of two index finger specimens (Template in white and Target in gray) and
(b) Manually-defined template building block structure with respective surface. 27

6. Inputs for mapping of proximal to distal phalanx bone:
(a) Distal phalanx (target) and proximal phalanx (template) surface and
(b) Manually-defined template building block structure for proximal phalanx. 28

7. Inputs for mapping between femur bones:
(a) Two specimen-specific right femur bone surface representations (Template in white and Target in gray) and
(b) Manually-defined template building block structure for template femur surface. 29

8. Inputs for mapping between C5 posterior vertebrae:
(a) C5 posterior vertebral surfaces from two subjects (Template in white and Target in gray) and
(b) Template building block structure with surface. 30

9. Target surface with,
(a) manually-defined building block structure and
(b) mapped building block structure.
Scaled Jacobian values of mesh generated using,
(c) manually-defined building block structure and
(d) mapped building block structure. 32

10. Distal phalanx bone with,
(a) manually-defined building block structure and
(b) mapped building block structure.
Scaled Jacobian of mesh generated using,
(c) manually-defined building block structure and
(d) mapped building block structure…….33

11. Target surface with,
(a) manually-defined building block structure and
(b) mapped building block structure.
Scaled Jacobian values of mesh generated using,
(c) manually-defined building block structure and
(d) mapped building block structure.……34

12. Mapped building block structure,
(a) before editing and
(b) after editing.…….....……35

13. Scaled Jacobian values of C5 posterior vertebral mesh generated using,
(a) manually-created building block structure and
(b) mapped building blocks……..36

14. Mapped building block structure before editing, resulting in heavy distorted elements in the spinous process region.....……..39

CHAPTER I

INTRODUCTION

The use of computational methods for the study of orthopaedic biomechanics yields information that is often difficult or even impossible to obtain experimentally. The implementation of 3D models for analyses over the last few decades has played an important role in understanding bone remodeling, assessing fracture risk and designing implants and prostheses [1]. Most finite element analyses are performed using an average geometry that represents a baseline model. Although these models yield valuable information, it is necessary that the morphological differences between subjects be accounted for [2, 3]. Subject-specific modeling and analysis hold the potential for custom design and fitting of orthopaedic implants and improved surgical planning. Bringing subject-specific modeling and analysis closer to clinical settings has long been a goal in orthopaedic biomechanics research.

Three-dimensional finite element analysis is a widely-used, powerful computational tool that enables the prediction of spatio-temporal variations in stress and strain. Generating high quality, subject-specific, 3D finite element anatomic models with minimal user-intervention remains a challenge. Numerous automated tetrahedral mesh generators are available, but hexahedral meshes are often preferred due to their higher stability and improved computational accuracy over tetrahedral meshes [4, 5]. This is especially true in cases where contact analyses are to be performed. Unfortunately, the generation of hexahedral meshes often proves to be a tedious and time consuming task.

In an effort to ease the mesh development process, IA-FEMesh [6] was developed. The meshing techniques employed are based on an interactive multi-block approach. The multi-block meshing method entails the generation of a block structure which is further subdivided through mesh seeding and projected onto the surface of interest to create the hexahedral mesh. Using IA-FEMesh, complex anatomical structures have been meshed with relative ease. The ability to create finite element meshes easily and quickly is a big step towards bringing subject specific modeling closer to use in clinical settings.

Several studies incorporating subject-specific modeling exist in the literature. For example, Gardiner *et al.* [7] performed subject-specific finite element studies on the human medial-collateral ligaments and showed that subject-specific models gave comparable results to the experimental data. Both experimental and FE-predicted values supported data from previous clinical, computational and experimental studies. The hexahedral FE meshes for this study were generated on a FE preprocessor program called TrueGrid (XYZ Scientific, Livermore, CA). Anderson *et al.* [8] developed and validated subject-specific pelvic models. Volumetric tetrahedral meshes for this study were generated using CUBIT, Sandia National Laboratories, Albuquerque, NM. A survey on the current progress of patient-specific modeling has been published by Neal *et al.* [3], providing an insight into the various patient-specific analyses and applications that have been performed in the various fields of medicine, like cardiovascular and orthopaedic biomechanics, soft tissue mechanics, oral, brain, heart and tumor modeling for surgery. They also recognize that, to enable the transition of patient specific modeling to the

clinic, model developers should not only validate their models against empirical data for a given patient population but also have to demonstrate their enhancement of patient care.

Once an accurate mesh has been created and validated for a given structure, it would be ideal for this mesh to act a template for generating models for other subjects, without disrupting mesh quality. Ultimately this obviates the need to create a new mesh from scratch. Consequently, the ability to reduce the time devoted to mesh generation enhances the feasibility of translating these tools to the clinical setting. We have successfully developed two mapped meshing algorithms (a force-based method [9] and a displacement-based method [10]) to map a template mesh onto a target surface. In the force-based mapping method, the template mesh and the target surface were first aligned and the template mesh was deformed onto the surface by applying forces on the surface nodes hierarchically. The displacement-based approach involves mapping the nodes of a template building block structure to the surface of interest by computing the distance between the building block structure and the surface.

When mapping a mesh from one surface to another, element distortion (i.e., dihedral angle less than 45° and greater than 135°) is not uncommon. Moreover, as the number of elements defining the template mesh increases, not only does the tendency for distorted elements increase, so does the time required to map the mesh. Consequently, our goal was to map the building block definitions rather than the mesh. Doing so provides control over the resulting mesh; i.e., once mapped to the new surface, the building block structure and the mesh seeding may be readily edited using the interactive tools available in IA-FEMesh, if required. Moreover, the time required to map a building

block structure is significantly less than that required to map a refined mesh, ultimately reducing meshing time.

This thesis is organized in the following manner: Chapter 2 consists of a literature review of the important topics such as the previous work done in the field and those that form a basis of the mapping algorithm. Chapter 3 describes the materials and methods that were used in order to develop and evaluate the mapping algorithm. The results obtained from the evaluation of the algorithm are reported in Chapter 4. Chapter 5 presents a discussion on the results of the evaluation, the various benefits and drawbacks of the algorithm and the proposed solutions to the drawbacks.

CHAPTER II

LITERATURE REVIEW

Finite Element Method

Numerical simulations of physical models (that are described by partial differential equations) are extensively used for design, dimensioning and validation purposes in engineering analysis. One of the most frequently used methods is the finite element method (FEM). In this method, a continuous problem (the initial PDE model) is replaced by a discrete problem that can be computed. The solution to this discrete problem is an approximate solution to the initial problem whose accuracy is based on the approximations that were made in the numerical process [11]. It is necessary to assess the accuracy and if the accuracy criteria are not met, the process should be repeated with refined solution parameters (finer meshes) until the desired accuracy is reached (convergence). Only accuracy can be tested using a convergence study; the validity is assessed using experimental verification or other means. The use of unreliable finite element methods is simply unacceptable in engineering practice [12].

The finite element method was first developed for the analysis of problems in structural engineering in the early fifties. However, it was soon realized that the technique could be applied in various other fields as well. In 1972, the use of the finite element method was extended to the field of orthopaedics to evaluate stresses in bones [13]. Since then, the use of FEM has grown exponentially and has played an important role in various facets of orthopaedic biomechanics research and design analysis; in

understanding not just the properties of bone, but also other tissues like ligaments, in assessing fracture risk, in designing implants and prostheses. [14].

The main steps involved in the finite element method (for a simple linear system) are: a) defining the computational domain, b) mesh construction, c) interpolation, d) constructing a matrix and right-hand side to complete the system corresponding to the discretization of the initial equations based on element connectivity, and e) computing the solution of the system.

Let us assume the following to derive the equation for the linear elastic finite element problem.

- The displacements of the finite element assemblage are infinitesimally small and that the material is linearly elastic.
- The nature of the boundary conditions remains unchanged during the application of the loads on the finite element assemblage.

To solve the boundary value problem (BVP) using the finite element method, two essential steps must be taken.

- The BVP has to be rephrased in its variational form.
- The variational form should be discretized in a finite dimensional space.

The essence of the variational approach is to calculate the total potential energy (Π) of a linear elastic body. This is given by,

$$\Pi = \frac{1}{2}\int_{\Omega} EA\left(\frac{du}{dx}\right)^2 dx - \int_{i} uf\ dx$$

Where,

$$\sigma = E\ \frac{du}{dx}$$

σ is the stress vector,

E is the Young's modulus,

u is the displacement vector,

A is the constant cross-sectional area, and

f is the point load applied at the node i.

After the variational form of the structural model is discretized, we get,

$$\Pi = \frac{1}{2} U^T K U - U^T R$$

Where,

U is a vector of the system global displacement,

K is the structure stiffness matrix (sum of the element stiffness matrices),

R is a vector of forces acting in the direction of these displacements.

After this, the final equation for the finite dimensional linear problem whose solution will approximately solve the BVP can be found. It is given as equilibrium equations for the system of the form,

$$KU = R$$

This corresponds to a linear analysis of a structural problem because the displacement response U is a linear function of the applied load vector R; i.e., if the loads are αR instead of R, where α is a constant, the corresponding displacements are αU. When this is not the case, a non-linear analysis is performed.

Mesh Generation

Mesh generation is an important step in the finite element method. Mesh generation techniques have rapidly developed since the advent of the finite element

method in the 1950s. Early mesh generation methods mainly gave rise to structured meshes (meshes consisting of quadrilaterals in two dimensions or hexahedra in three dimensions); each vertex of such meshes can be readily defined as an array of indices. Therefore, any mesh that has a high degree of ordering is called a *structured mesh*. More recently, many developments have been made in order to cater to the development of models with extremely complex geometries. These meshes were difficult to mesh as structured entities, so the search of an efficient alternative to address this problem gave rise to what are known as *unstructured meshes*. Generally, most workstations can generate unstructured tetrahedral meshes for complex 3D domains of an arbitrary shape in reasonable time. However, improvements on the robustness, reliability and optimality of the meshing techniques are still expected. Certain fields of engineering require specific features from analyses (like Computational Fluid Dynamics), which cannot be attained by using structured or unstructured meshes. The combination of the two, however, can meet the demands. This is achieved by using hybrid meshes.

A mesh can be composed of elements of different geometric natures. A mesh can consist of a finite number of segments in one dimension, segments, triangles and quadrilateral elements in two dimensions and one dimensional segments, tetrahedral, pentahedral and hexahedral elements in three dimensions. When it comes to three dimensional finite element modeling, hexahedral elements are preferable to tetrahedral elements. Hexahedral elements generally deform under lower strain energy states, hence, hexahedral meshes provide more accurate results when compared to tetrahedral meshes [15]. Also, they are more stable and less influenced by the degree of refinement of the mesh when compared to tetrahedral meshes [5]. However, an undesirable quality in

hexahedral meshes is that the generation of a hexahedral mesh is a difficult task. Splitting of the elements of a given tetrahedral mesh by means of 4 hexes is one method to give rise to a hexahedral mesh. This, however, gives rise to poorly shaped elements [16]. Therefore, direct construction of hexahedral elements is more desired. The following are the various hex meshing methods,

1. Advancing-front based method [17]: Given a surface, the various points that will be suitable candidates for creating a hex element will be chosen and elements will be built progressively inward. Several techniques have been proposed to achieve this (such as plastering [18]), some of which involve the introduction of some non-hexahedral elements (pyramids, prisms, tets, etc.), leading to mixed meshes.
2. Grid-based method [19]: The grid-based method involves generating a fitted three dimensional grid of hex elements on the interior of the volume. Hex elements are added at the boundaries to fill gaps where the regular grid of elements does not meet with the surface. Therefore, the elements at the boundaries are often of poor quality.
3. Method using mid-surface (Medial surface method) [20]: A mid-surface is constructed to partition the domain in terms of regions whose topology is simpler so that a direct method can be used to mesh them. This method has not been successfully applied to all kinds of geometry.

Voxel meshing [21] is a popular hexahedral mesh generation technique where 8 node cubic elements can be generated directly from the voxels of a CT dataset. The material properties of each element are proportional to the Houndsfield number

associated with the corresponding voxel in the CT dataset. However, the main drawback of this method is that the mesh has a ‘stair-stepped’ appearance due to which the surface is not smooth. This can be a big disadvantage as contact analyses cannot be performed using meshes generated using this technique.

Mapped meshing [22] is another hexahedral mesh generation technique, where the object to be meshed is decomposed into a number of objects and provides hexahedral elements through a direct mapping of the element onto the geometry. Complex geometry may call for significant amount of user intervention, but the method is highly accurate. Various mapped meshing and mesh morphing studies have been performed in the past. Couteau *et al.* [23] proposed the Mesh-Matching method based on a grid projection algorithm, which was tested only on long bones. Chabanas *et al.* [24] extended the application to work on maxillofacial structures. Gibson *et al.* [25] generated a model of the neonatal head by fitting a surface mesh of an adult head over the surface of a neonatal head. As mentioned before, we have successfully developed two mapped meshing algorithms (a force-based method [9] and a displacement-based method [10]), to map a template mesh onto a target surface. O’Reilly *et al.* [26] utilized a mesh-morphing algorithm to create patient-specific models of the spine, where the template nodes were moved to the target location after a comparison of the target and template. Baldwin *et al.* [27] have also used mesh morphing techniques to create subject-specific knee models. In their technique, morphing a structure onto another involved the division of the template mesh into groups of elements bounded by control points on the group corners. The control handles were used to align and reshape the template mesh over the target image.

In the case of tetrahedral meshing, there are many algorithms available to generate large meshes in a very short amount of time. The few automated hexahedral mesh generation algorithms that are available can be used to mesh only a certain class of geometries. Because of this, a significant amount of time in generating a hexahedral mesh is devoted to decomposing (cutting up) a model into pieces for which a known hexahedral mesh generation algorithm will succeed [28]. As mentioned before, we have developed a hexahedral mesh generator called IA-FEMesh [6], which is discussed in detail in the following section.

IA-FEMesh

IA-FEMesh is a hexahedral mesh generator that employs a multi-block technique, where structured meshing techniques are applied to a series of interconnected sub-grids or blocks. These multi-block grids are a powerful extension of the structured mesh. While the individual blocks remain structured, the blocks fit together in an unstructured manner, harnessing both the advantages of structured and unstructured meshes. The multi-block technique affords geometric flexibility while retaining computational efficiency.

An image dataset (CT/MR) is processed to produce a triangulated surface representation of the region of interest (stl/vtk format). This surface is the base of the mesh generation process in IA-FEMesh. A series of blocks, called a building block structure is built interactively as shown in Figure 1. The number of blocks in the structure depends upon the complexity of the underlying surface and is left to the discretion of the user.

Consider the simple case of the proximal phalanx bone of the hand where one block is enough to mesh the surface. Figure 1(a) shows the automatic definition of the building block at the request of the operator, the dimensions of which are established directly from the bounds of the surface of interest. A wide range of tools are available in the user-interface, for the manipulation of the building blocks. The building block is altered to provide control over the resultant nodal projections (see Figure 1(b) and (c)). The building block structure is assigned a mesh seeding arranged in rows, columns, and layers; the corresponding level of seed refinement is specified by the user. The mesh seeds of the building block are then projected (via closest point projection) onto the surface of interest as seen in Figure 1(d) and (e).

As a result, the mesh seeds are morphed to the bony surface as nodes, to create a mesh that is composed solely of hexahedral elements. Once the mesh is created, material properties and boundary conditions may be specified and the mesh can be used for analysis. This technique is versatile and can be applied to the meshing all types of bony surfaces.

It should be noted that while using IA-FEMesh, the generation of a good quality building block structure is the most important and often, the most time-consuming task in the mesh generation process. Therefore, using mapped building block structures for mesh generation will significantly reduce meshing time.

The building block mapping algorithm is an algorithm that performs deformable (non-rigid) registration of a template building block structure over a target surface representation. This mapped building block structure can be used to mesh the target surface using IA-FEMesh. The algorithm is explained in detail in the next chapter.

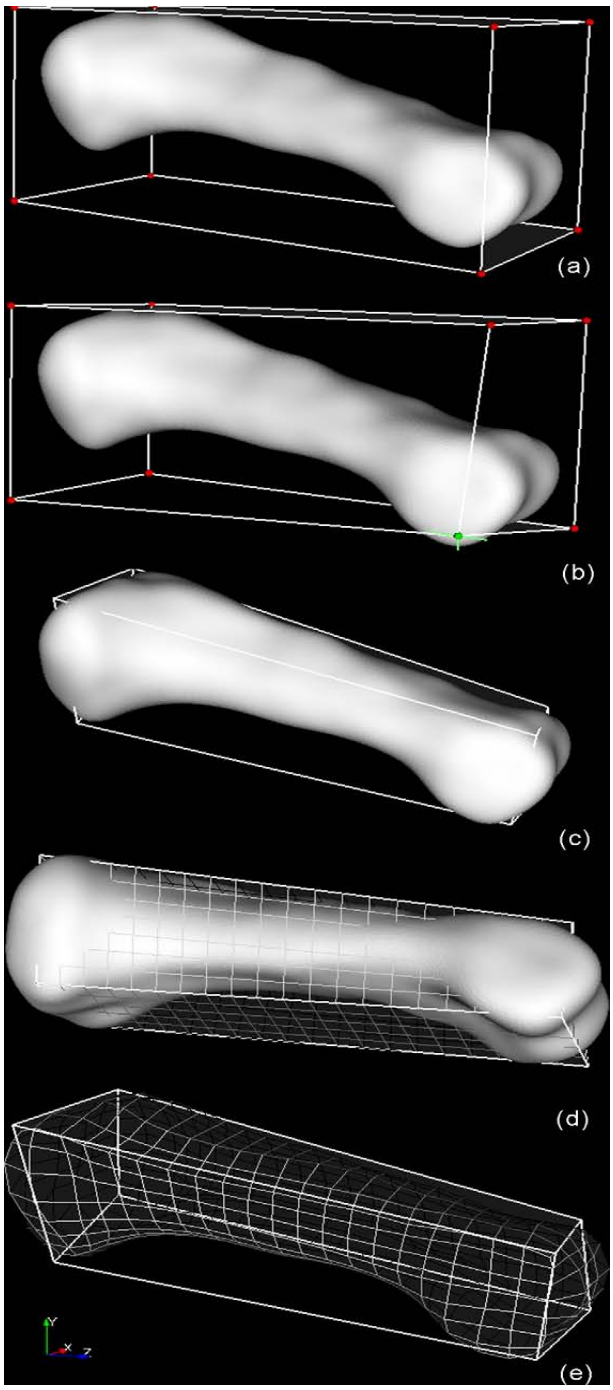


Figure 1. Mesh generation using IA-FEMesh [6]
(a) Automatic block definition,
(b) & (c) Manual realignment of vertices,
(d) Mesh seeding of the block and
(e) Closest point projection onto surface to create the mesh.

Deformable Image Registration

Image registration is the process by which the correspondence of features between images collected at different times or using different image modalities (from CT to MRI, etc.) are determined [29, 30]. This correspondence can then be used to change the orientation (by rotation, translation) or appearance (size, shape) of one of the images so that it resembles the other more closely. Then the two can be directly compared, combined or analyzed. The main component of image registration is the determination of the spatial transformation or mapping.

Image-registration is classified as either rigid or non-rigid. Rigid registrations involve only the rotation and translation of one image with respect to the other. Whereas, non-rigid (deformable) registrations involve resizing of the image, in addition to rotation and translation. Thin-plate spline, b-spline, etc. are types of non-rigid registration. Most medical image registration involves non-rigid registration operations.

There are two main approaches to deformable registration: (1) Geometry-based and (2) Voxel-based. The geometry-based approach defines the registration based on identifiable anatomic features in the images. These include functionally important surfaces, curves and point landmarks that can be matched with their counterparts in the second image. The use of such structural information ensures that the mapping has biological validity and allows the transformation to be interpreted in terms of the underlying anatomy or physiology. The voxel-based approach utilizes the variation in intensity from region to region within the images and matches corresponding regions based on the intensity. Our algorithm makes use of the geometry-based approach to non-

rigid registration applied to building block structures. Figure 2 shows the non-rigid registration being applied to inter-subject brain matching.

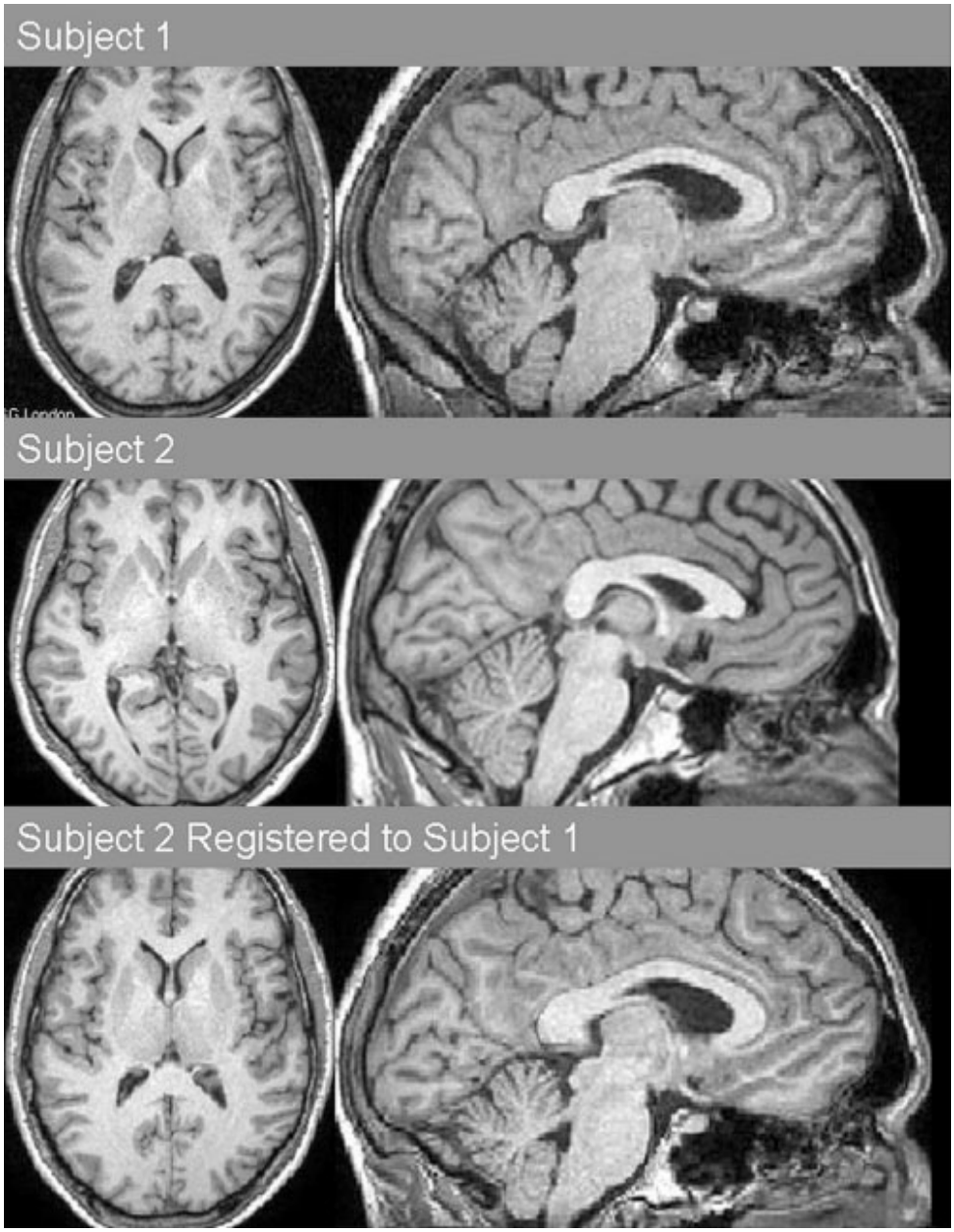


Figure 2. Non-rigid registration being applied to inter-subject brain matching of two selected slices from a T1weighted MR-volume [30].

Our algorithm makes use of the geometry-based approach to non-rigid registration applied to building block structures, such as the Thin-Plate Spline. Spline-based registration algorithms use corresponding control points, in the template and target image and a spline function to define the correspondences. The Thin-Plate Spline [31] has been used extensively in the field of medical imaging. There is however, a major disadvantage: each control point belonging to a thin-plate spline has a global influence on the transformation, such that, if its position is perturbed, all other points in the transformed image change. This can be a disadvantage because it limits the ability to model complex and localized deformations because, as the number of control points increases, the computational cost associated with moving a single point rises correspondingly.

It also should be noted that while the majority of the rigid transformations mostly take only a few seconds or minutes, most non-rigid registration algorithms require a long time (minutes or hours). This time is spent either identifying a geometric set of corresponding features to match directly or automatically determining a large number of parameters by matching voxel intensities directly. This is the case when a template mesh is mapped to a target surface, as the mesh and the surface can both have a very large number of points. Therefore, using building block structures as the template will reduce the computational time.

If the correspondence between the images is not well defined, in order to improve the registration, we can optimize the images first before applying the thin-plate spline registration. Optimization refers to the manner in which the transformation is adjusted to improve the image similarity. An example of an optimizer is the Iterative Closest Point

(ICP) algorithm [32]. The algorithm iteratively performs two operations until convergence:

- Finding the closest point in one point set for each point in the other point set.
- Estimating the motion between the two point sets using the corresponding point pairs.

However, a drawback of ICP is that it requires a good initialization in order to perform well. The Iterative Closest Point (ICP) registration was the method used for initial alignment in both the force-based and distance based methods. In a separate study performed to speed up the mapping of the meshes, the k-D tree search structure [33] was used the ICP algorithm in the place of a linear search. Therefore, this modified ICP registration was also used in the building block mapping algorithm to perform the initial affine transformation [34]. k-D tree search structures have been explained in the next section.

k-D Tree Search Structures

k-D tree (where k is the dimensionality of the search space) is a multi-dimensional binary search tree for storing point data. They facilitate very fast searching, and nearest-neighbor queries [35]. For example, a two dimensional search tree might consist of two dimensional coordinates of the point on a map.

The k-D tree is a binary tree in which each non-terminal node has two descendants, a *left son* and a *right son.* Each non-terminal node partitions the points into two groups according to the position with respect to a k-dimensional partition

hyperplane, records that lie on the left side are stored in the left son sub-tree, while records that lie on the right side of the hyperplane are stored in the right son sub-tree. A hyperplane is a geometric structure that defines an m-dimensional subspace within an n-dimensional subspace, such that m<n. Therefore, a line described by 2 points is a hyperplane in 2D space, a plane described by 3 points is a hyperplane in 3D space, etc. Therefore, for a particular value (say of x axis) of the hyperplane, all points that have a lower value of the same coordinate lie to the left of the hyperplane and the points with higher values lie to the right of the hyperplane. The left and right branches are further divided by splitting hyperplanes till the end is reached.

The nearest neighbor search using a k-d tree search structure finds the point in the tree that is nearest to a given input point. Searching for a nearest neighbor in a k-D tree is done in the following manner,

- The algorithm moves down the tree starting from the root.
- When the tree branches, the algorithm saves the *current nearest* as the nearest neighbor.
- Thereafter, at each node, the current best is updated if a nearer point is found, or is left unchanged if it still is the closest point.
- The algorithm checks whether there could be any points in the other ‘son’ that are closer to the search point than the current nearest. This is done by intersecting the hyperplane with a hypersphere around the search point, such that the radius of the hypersphere is equal to the distance of the point to the current nearest (dN). The distance from the search point to the partition plane (dP) should be checked to see if it is lesser than dN. If the

partition plane is intersected by the hypersphere, then the other son has to be searched.

- This process continues till the end of the tree.

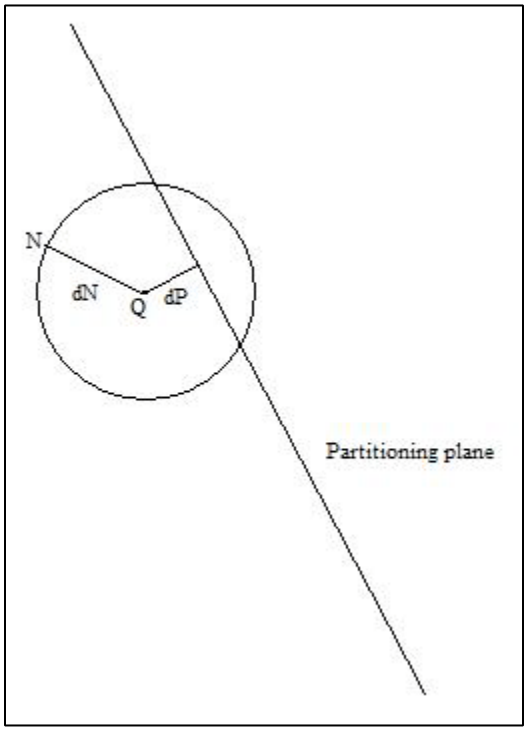


Figure 3. Mechanism of k-D tree nearest neighbor search. While Q is the query (search) point, and N is the current nearest neighbor, if dP<dN, the actual nearest neighbor may lie on the right of the partition plane [35].

CHAPTER III

MATERIALS & METHODS

Building Block Mapping Algorithm

The mapping algorithm accepts a template building block structure and a target surface as inputs and produces a building block structure corresponding to the target surface as an output. The resulting mapped building block can be used to create a finite element mesh for the target surface using IA-FEMesh.

During the first stage of the algorithm, an affine transformation is used to bring the building block structure into rough alignment with the target surface. In this algorithm, we have enhanced the ICP registration by utilizing a k-D tree search structure to generate the estimate of point pairs. The points in the k-D tree structure were used to create a k-D tree point locator and therefore the nearest corresponding point in the target surface to each point in the template structure was found. This replaced the nearest neighbor search that was used to compute the point pairs in the original algorithm. After finding the closest point pairs, the distance between them is computed in the algorithm and the updated transform from the template to the target is estimated. Computation of closest point pairs is a crucial step in ICP registration. Because the highly optimized k-D tree searching is more efficient than linear searches, they facilitate very fast and accurate nearest-neighbor queries.

In the building block mapping algorithm, after the template building block structure is roughly aligned with the target surface using the ICP registration, the external nodes of the building block structure were repositioned to the surface of the bony

representation such that the surface is completely enclosed by the building block structure. Thereafter, the displacement-based mapping algorithm was employed to map the building blocks around the target surface. In this implementation the distance between the exterior building block vertices and the target surface is computed along the direction of the building block vertex normals. The algorithm iteratively moves the vertex point towards the desired surface location using the following equation:

$$x_i = x_{i-1} + \frac{i * n}{N}(y - x_{i-1})$$

where, N is the number of iterations, i is the current iteration, y is the desired point location, n is the normal direction and x is the current point location.

This algorithm defines a non-linear mapping to locally register the exterior points of the building block structure to match the shape of the target surface, such that all exterior building block vertices lie on the target surface. The multi-block approach to FE meshing involves closest point projection of the seeded building block onto the target surface. Therefore, it is advantageous to bring the vertices/faces as close as possible to the surface in order to control the quality of the resulting mesh.

Following the non-linear mapping of the exterior points, the distribution of the interior vertices is recomputed and repositioned using a Thin Plate Spline (TPS) transform that is defined using the initial surface points as template positions and the mapped points as target positions.

The pseudo-code of the algorithm is presented below:

- Load the target surface (vtk format)
- Load template building block structure [bbs] (vtk format)
- Perform modified ICP (between surface and bbs)

- Apply transform to bbs
- Write out the resulting bbs (Write out a copy as the original bbs)
- Get geometric features out of the entire bbs using a geometry filter
- Create a point locator for the points in the target surface
- Find the number of cells (building blocks) in the bbs
- Loop through each block
    - Loop through each face of the block
        - Calculate normal along which the face is to be moved
    - Calculate cell center
    - Check whether all points of a face are outside the surface or not
        - Loop through each point in a face
            - Get point coordinates
            - Check if the point lies outside surface or not
            - If yes, increment counter
            - Find the closest point on surface to cell center and distance between the two points
        - If the variable counter equals 4, face lies outside surface, so make the distance negative so that the normals are flipped
    - Change coordinates of each point by moving the point along normal through the distance
- Write out modified building block structure (expanded faces)
- Create point locator for surface points
- Compute normals on bbs surface

- Use the target point and the original points of the bbs for computing the Thin Plate Spline (TPS) to reorder the internal nodes

- Apply the transform to the bbs

- Write out the final mesh (vtk format)

In order to modify the vtkMimxIterativeClosestPointTransform source and header files to implement a kDtree search structure, replace the cell locator with kDtree point locator and change the code to work with points as opposed to cells. Therefore, all the appropriate header files should be edited and included. Check to see if the variable names and function names in all related files are consistent. Since IA-FEMESH is built using VTK version 5.2.1, the kdtreepointlocator class is not available. Hence, a compatible version of the kdtree class and kdtreepointlocator have to be created in the IA-FEMESH source.

Figure 4 shows a pictorial representation of the building block mapping algorithm.

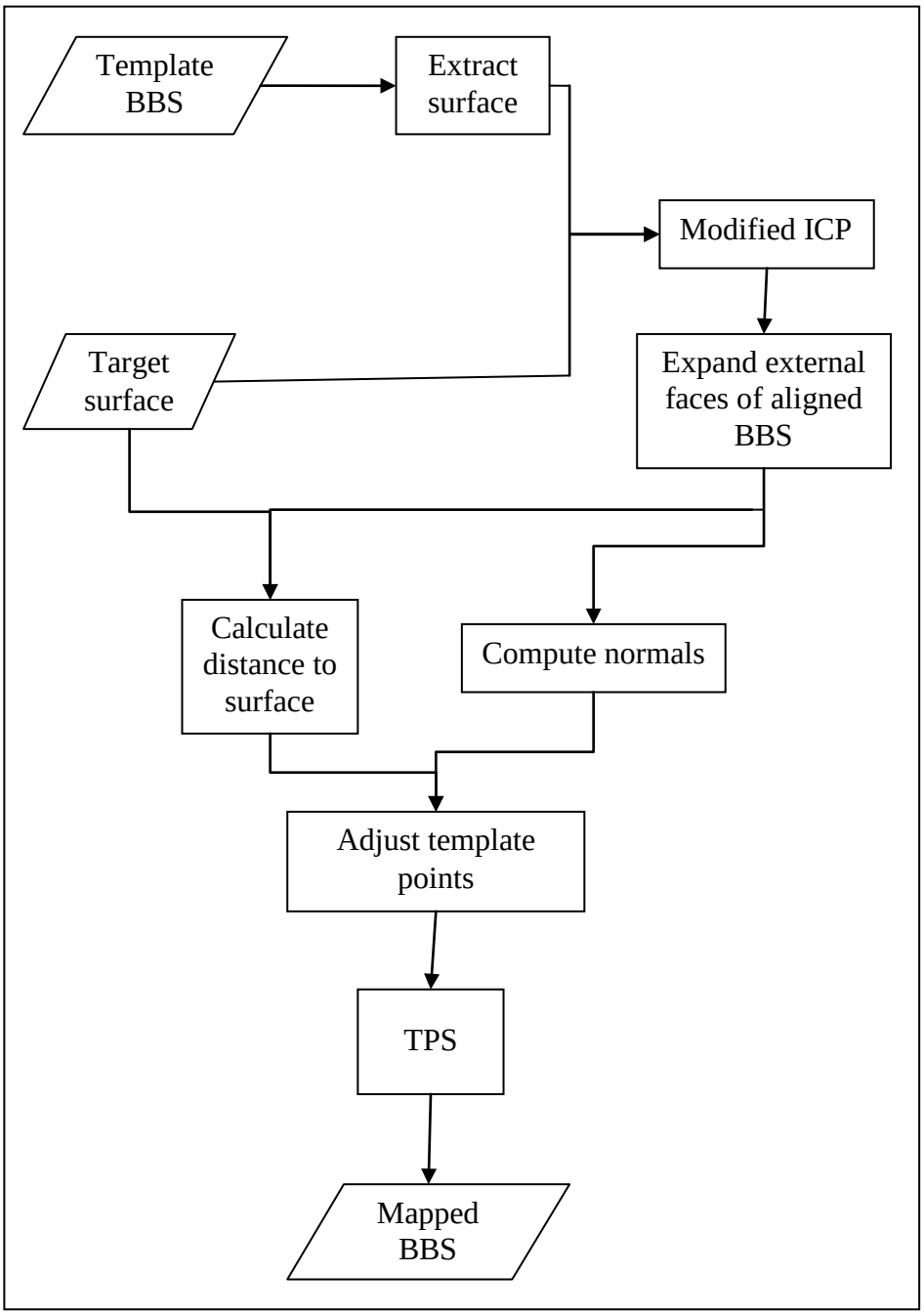


Figure 4. Building block structure mapping algorithm.

Software

The algorithm for this application was developed in C++ using the Visualization Toolkit (VTK), an open-source software system for 3D computer graphics, image processing and visualization [36].

Evaluation of the Algorithm

A variety of bone morphologies were used to evaluate the feasibility of mapping a building block structure. The datasets used for this study were obtained from human cadaveric specimens and included: 1) proximal phalanx bone of the index finger from two specimens, 2) proximal and distal phalanx bones from a single index finger specimen, 3) right femur from two specimens and 4) the posterior region of the C5 vertebra from two specimens.

The bony surface representations were obtained by scanning cadaveric specimens of the hand, cervical spine, and femur. Using a Siemens SOMATOM Emotion 6 Slice CT scanner, three-dimensional voxel datasets of the skeleton were acquired (matrix=512x512, FOV=480x480mm, KVP=110kV, Current=170mA, Exposure=70mAs) with an in-plane resolution of 0.9375mm and 1.25mm slice thickness. Regions of interest for the phalanx bones, cervical vertebrae and the femur were obtained using the BRAINS2 image analysis suite [37]. The resulting binary segmentations were imported into 3D Slicer [38] where triangulated surfaces were generated. These were then subjected to Laplacian smoothing (while preserving their anatomic integrity) to give rise to the final surface representations.

The building block structure of a specimen that was created manually [6], specifically for the corresponding surface, was mapped to the surface representation of a second specimen differing in size, shape and spatial orientation. The templates used were building block structures that produced high-quality meshes with no or few distorted elements that could be eliminated during mesh improvement.

The mapped building block structure was edited to correct any irregularities (if present) and then used to create a mesh of the surface in IA-FEMesh. This mesh was compared to the mesh created using a building block structure that was manually defined for that specific surface. No smoothing or untangling operations were performed on any of the meshes while comparing the mesh quality based on the volume and scaled Jacobian metrics.

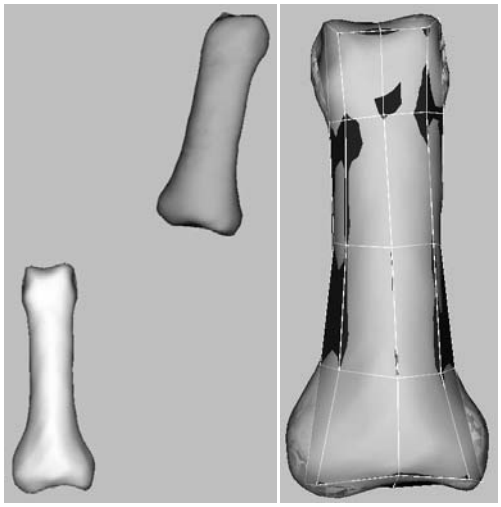
The datasets used to evaluate the efficiency of the mapped building block structure were as follows:

Proximal Phalanx Bone

This dataset consisted of the proximal phalanx bone from two index fingers (Figure 5a). A manually generated building block structure was used to mesh one of the surfaces (Figure 5b). These blocks, in turn, served as a template block structure and were mapped to the second phalanx surface (i.e., target surface); thereby enabling a mesh of the second bone to be readily generated.

The mapped building block structure was used to mesh the target surface. The optimum number of elements for the proximal phalanx bone used in this case was determined though a convergence study performed by DeVries *et al.* [39], to be 4550. A

mesh of the target surface was generated using the manually-defined building block structure with 4560 elements and a mesh was created using the mapped building block structure with a total of 4536 elements. The quality of these two meshes was compared.



a. b.

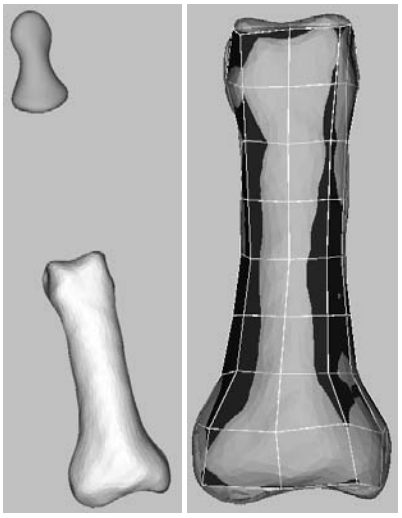
Figure 5. Inputs for mapping between proximal phalanx bones:
(a) Proximal phalanx bone of two index finger specimens (Template in white and Target in gray) and
(b) Manually-defined template building block structure with respective surface.

Proximal to Distal Phalanx Bone

This dataset was chosen to demonstrate the ability of the algorithm to perform well between different anatomical structures that have relatively simple morphologies (Figure 6a). The template building block (target surface in the previous case) was that of a proximal phalanx bone, that was created manually, specifically for this proximal phalanx bone surface (Figure 6b). A study was performed where building block structures of varying refinements were used as templates. Based on this study, a more

refined building block structure was used for mapping between anatomically different structures.

The mapped building block structure was used to mesh the target surface. The optimum number of elements for the distal phalanx bone specimen was determined to be 3402 elements through a convergence study performed by DeVries *et al.* [39]. Both the meshes (the one created using the manually-generated and mapped building block structures) were created with a mesh seeding such that the number of elements was 3432.



a. b.

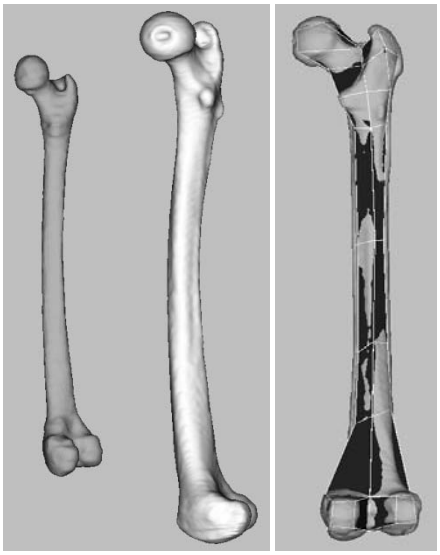
Figure 6. Inputs for mapping of proximal to distal phalanx bone:
(a) Distal phalanx (target) and proximal phalanx (template) surface and
(b) Manually-defined template building block structure for proximal phalanx.

Femur

This dataset consisted of the right femur bone from two subjects (Figure 7a). A manually generated building block structure was used to mesh one of the surfaces (Figure 7b). These blocks, in turn, served as a template block structure and were mapped to the

second phalanx surface (i.e., target surface); thereby enabling a mesh of the second bone to be readily generated.

The mapped building block structure was edited where required. This building block structure was then used to mesh the target surface. A mesh of the target surface was generated using the manually-defined building block structure with 13888 elements and a mesh was created using the mapped building block structure with a total of 13977 elements. The quality of these two meshes was compared.



a. b.

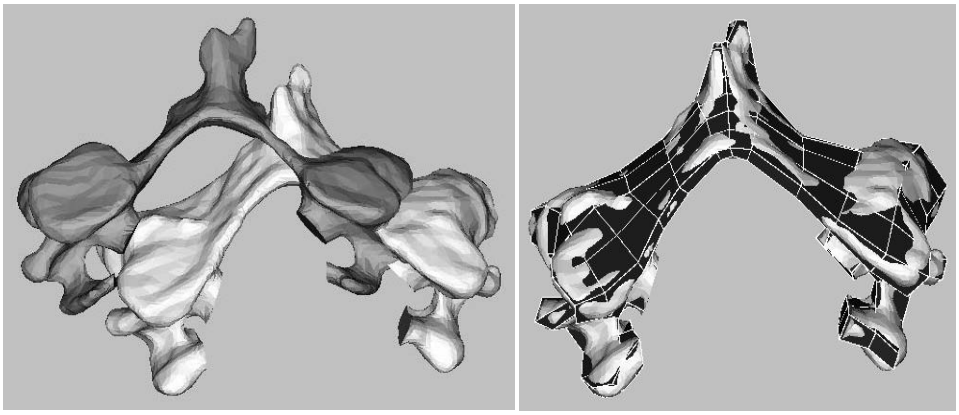
Figure 7. Inputs for mapping between femur bones:
(a) Two specimen-specific right femur surface representations (Template in white and Target in gray) and
(b) Manually-defined template building block structure for template femur surface.

C5 Posterior Vertebra

Here, the dataset consisted of a bony surface representation of the posterior region of the C5 vertebra (vertebral body clipped off from intact vertebral surface) belonging to

one subject and the building block structure corresponding to the posterior region of the C5 vertebra belonging to another subject (see Figure 8). This template building block structure was created specifically for the template surface for a previous study [40]. This dataset was chosen to demonstrate the ability of the algorithm to perform well with complex anatomical structures.

A mesh was generated using the manually-generated building block structure with 7277 elements and a mesh was created using the mapped building block structure with a total of 7256 elements.



a. b.

Figure 8. Inputs for mapping between C5 posterior vertebrae:
(a) C5 posterior vertebral surfaces from two subjects (Template in white and Target in gray) and
(b) Template building block structure with surface.

Evaluation Metrics

The registration algorithm was evaluated by via an assessment of the resulting mesh quality, in terms of zero-volume and distorted elements. Every mesh that is used for

Finite Element Analysis must satisfy certain quality conditions. Consequently, the metrics used included element volume and the scaled Jacobian [41, 42]. `Jacobian' is the matrix involved in the mapping between any tetrahedron over the canonical one. If an element cannot be mapped onto the canonical element, it is considered distorted. A good quality mesh should preferably have no such elements or a few that can be corrected using mesh improvement tools like smoothing and untangling. A tetrahedral element can be mapped with a single Jacobian matrix whereas hexahedral elements have a different Jacobian at each vertex.

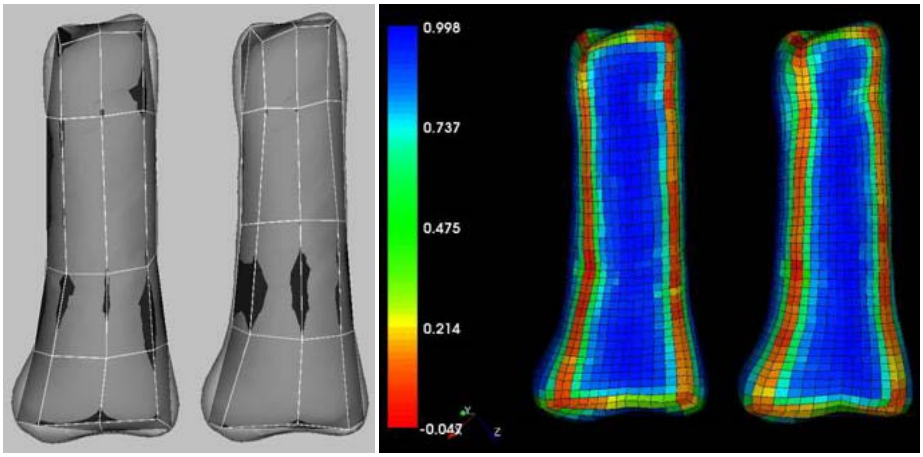
CHAPTER IV

RESULTS

Results of Evaluation

Proximal Phalanx Bone

The dataset was run through the algorithm and the algorithm running time was about 5 seconds. Figure 9b shows the mapped building block structure with the target surface.



a. b. c. d.

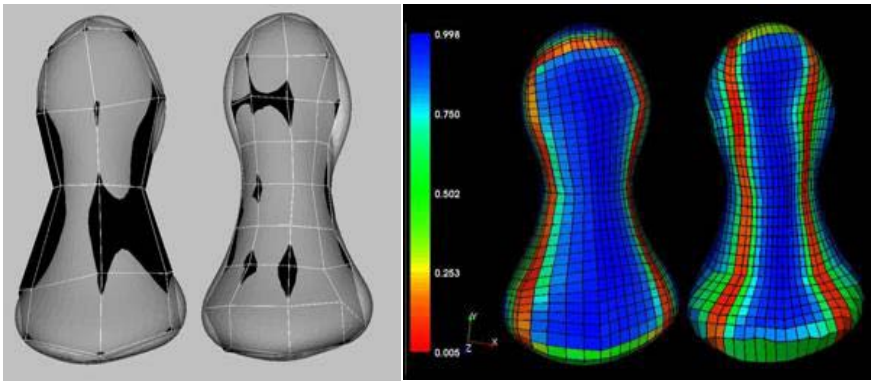
Figure 9. Target surface with,
(a) manually-defined building block structure and
(b) mapped building block structure.
Scaled Jacobian values of mesh generated using,
(c) manually-defined building block structure and
(d) mapped building block structure.

There were no distorted elements in the mesh created using the manually-defined building block structure while 2 distorted elements were found in the mesh created using

the mapped building block structure. Figure 9(c-d) shows a comparison of the quality of meshes obtained using manually-defined building block structures and mapped building block structures.

Proximal to Distal Phalanx Bone

The algorithm running time was about 35 seconds. Figure 10b shows the mapped building block structure with the target surface. See Figure 10c-d for a comparison of the quality of meshes obtained using manually-defined building block structures and mapped building block structures.



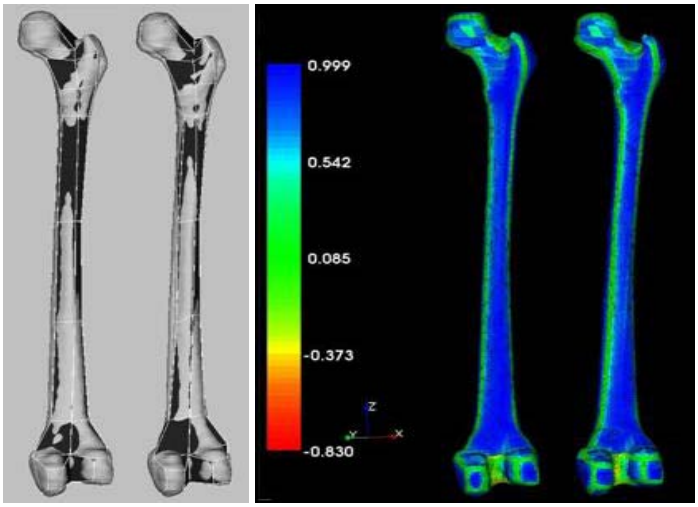
a. b. c. d.

Figure 10. Distal phalanx bone with,
(a) manually-defined building block structure and
(b) mapped building block structure.
Scaled Jacobian of mesh generated using,
(c) manually-defined building block structure and
(d) mapped building block structure.

There were no distorted elements in either the mesh created using the manually-defined building block structure or the mesh created using the mapped building block structure.

Femur

The algorithm running time was about 3 minutes. During mapping, the faces of two of the blocks were positioned such that meshing the surface gave rise to heavily distorted elements in those two areas. Therefore, minimal modifications were made to those two blocks in order to visibly reduce distortion. Since the structure is relatively simple with relatively few building blocks, modifying it did not take more than a couple of minutes.



a. b. c. d.

Figure 11. Target surface with,
(a) manually-defined building block structure and
(b) mapped building block structure.
Scaled Jacobian values of mesh generated using,
(c) manually-defined building block structure and
(d) mapped building block structure.

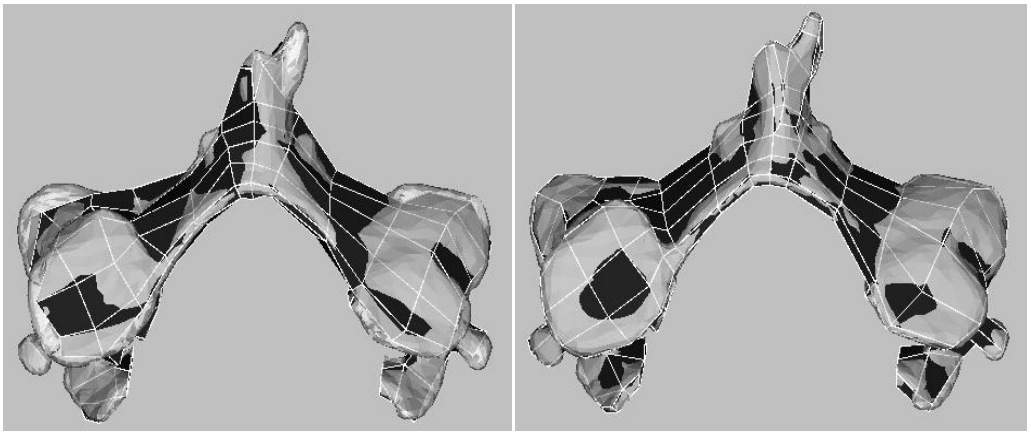
The number of distorted elements in the mesh created using the manually-defined building block structure was 66 (0.475% distorted) and the mesh created using the mapped building block structure was 91 (0.651% distorted). Figure 11b shows a

comparison of the quality of meshes obtained using manually-defined building block structures and mapped building block structures.

C5 posterior vertebra

The total processing time including the algorithm running time (about 3 minutes) for the C5 posterior vertebral dataset was under 20 minutes (the rest being spent for manual editing of the mapped building block structure).

In this case, three blocks were added at the spinous process region, to compensate for the gross difference in morphology at that region between the template and the target surfaces. Certain faces were also repositioned along the articular processes and laminae.

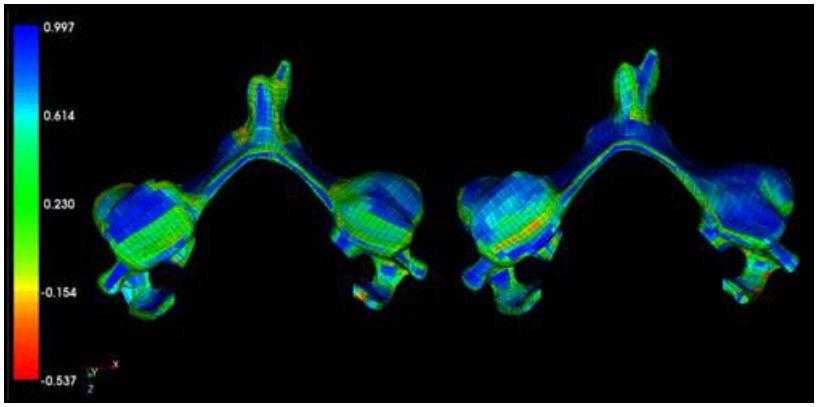


a. b.

Figure 12. Mapped building block structure,
(a) before editing and
(b) after editing.

The number of distorted elements in the mesh created using the manually-defined building block structure was 184 (2.536% distorted) and the mesh created using the

mapped building block structure was 218 (2.996% distorted). Figure 13 shows a comparison of the quality of meshes obtained using manually-defined building block structures and mapped building block structures.



a. b.

Figure 13. Scaled Jacobian values of C5 posterior vertebral mesh generated using,
(a) manually-created building block structure and
(b) mapped building blocks.

Summary

None of the meshes had any negative or zero-volume elements. In all the four cases, while using scaled-Jacobian as the mesh quality metric, the percentage of distorted elements observed in the meshes created using the mapped building block structure compared favorably to that of the mesh created with manual block definition. In both cases (manual and automated) the results reported here are before mesh improvement. Thereafter, smoothing and untangling techniques were used to successfully remove the distortion from both mesh definitions in all cases, and the improved meshes could be used for performing finite element analyses.

Table 1. A Comparison of the Quality of Meshes Obtained by Meshing the Target Surface Using Manually-Defined and Mapped Building Block Structures.

Type of mapping	**Total no. of elements**		**Min & max Scaled Jacobian values**		**No. of distorted elements**	
	Manual	**Mapped**	**Manual**	**Mapped**	**Manual**	**Mapped**
Proximal phalanx to proximal phalanx	4640	4653	0.003 0.995	-0.021 0.995	0 (0%)	3 (0.065%)
Proximal phalanx to distal phalanx	3458	3432	0.024 0.998	0.005 0.997	0 (0%)	0 (0%)
Femur to femur	14353	14439	-0.855 1.000	-0.796 0.999	125 (0.871%)	147 (1.018%)
C5 posterior vertebrae to C5 posterior vertebrae	7256	7277	-0.537 0.992	-0.537 0.997	184 (2.536%)	218 (2.996%)

CHAPTER V

DISCUSSION

The mapping algorithm was successfully applied to the phalanx bones, femur and vertebrae. The algorithm was robust while mapping between anatomically different geometries. This was demonstrated in the case where the template building block structure belonging to the proximal phalanx bone was mapped to the distal phalanx bone. The finger bone serves as a good testing structure as it has a well-defined principal axis providing a good basis for registration, whereas the principal axis is less well defined in more complex structures such as the vertebrae. The algorithm also worked well on structures of complex geometries like the vertebrae. Using the modified ICP registration, it was observed that not only a faster, but also a better initial alignment was achieved in certain cases. Also, incorporating the step in the algorithm where the external faces where repositioned to enclose the surface entirely, ensured a better mapping of the building block structure.

The main aim of this algorithm is to achieve the best mesh with minimal user intervention. In order to demonstrate the efficiency of the algorithm, no modifications were made to the resulting building block as far as possible. This demonstrates that the algorithm produces results that are comparable with those produced manually, with minimal user intervention. However, since the structures are relatively simple with fewer building blocks in these cases, manual editing of the building block structure, if needed, would likely not take more than a few minutes.

The main drawback of the algorithm is that in complex structures such as the vertebrae, the mapped building block structures may be a few blocks short or may have a few blocks in excess, in areas where there is a gross difference in morphology (observed between subject to subject or vertebral level to vertebral level). This will result in heavily distorted elements as can be seen in Figure 14.

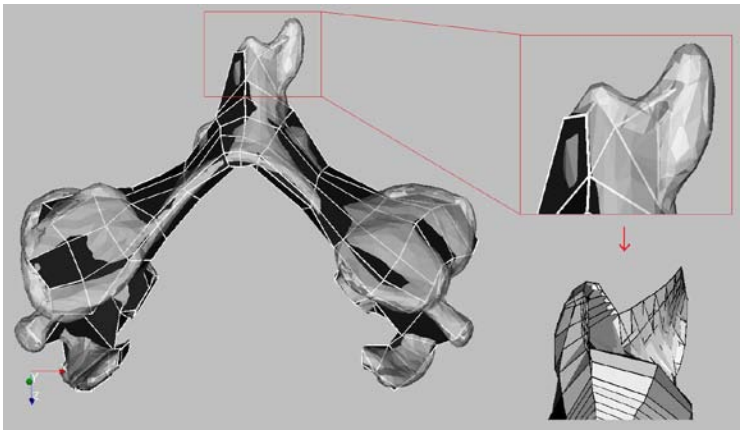


Figure 14. Mapped building block structure before editing, resulting in heavy distorted elements in the spinous process region.

The user should then realign the faces, add or delete blocks, wherever necessary, in order to obtain a high quality mesh. All these operations can be performed with relative ease in a short amount of time. For example, a C5 posterior vertebral mesh was generated in less than half an hour; including the algorithm running time, manual editing of building blocks and mesh generation and improvement. This was a significant reduction in time as compared to the original manually-defined block structure which took several hours to generate, thereby making performing large scale modeling studies like multi-level models of the spine, feasible. It should also be noted that the alignment of

the interior vertices by the Thin Plate Spline may not be satisfactory while working with certain anatomic structures with foramen and the user might have to manually align the misaligned internal vertices. In any case, the modifications will not take very long, owing to the relative geometric simplicity of the building block structures.

Therefore, the greatest advantages of this algorithm are that it leaves the control over the quality of the resulting meshes in the hands of the user, unlike other mesh morphing techniques; and the time required to generate high quality hexahedral meshes is greatly reduced.

As mentioned earlier, baseline models are not accurate representations of the individual patient and cannot be used for personalized healthcare. Using IA-FEMesh, subject-specific material properties can be used while creating finite element meshes. Including these minute but often crucial differences in geometry and material properties will enable the creation of high quality subject-specific finite element models. We are investigating methods to nullify the few drawbacks of this algorithm in order to progress to the development of a mapping algorithm that can generate the entire finite element mesh, with minimum user intervention in lesser time.

BIBLIOGRAPHY

1. Prendergast, P.J. *Finite element models in tissue mechanics and orthopaedic implant design. Clin. Biomech.* **1997**, *12*, pp 343-366.

2. Kent, D.M.; Hayward, R.A. *Limitations of Applying Summary Results of Clinical Trials to Individual Patients. J. Am. Med. Assoc.* **2007**, *298*, pp 10.

3. Neal, M.L.; Kerckhoffs, R. *Current progress in patient-specific modeling. Briefings in Bioinformatics* **2010**, *11*, pp 111-126.

4. Viceconti, M.; Bellingeri, L.; Cristofolini, L.; Toni, A. *A comparative study on different methods of automatic mesh generation of human femurs. Med. Eng. Phy.* **1998**, *20*, pp 1-10.

5. Ramos, A.; Simoes J.A. *Tetrahedral versus hexahedral finite elements in numerical modelling of the proximal femur. Med. Eng. Phy.* **2006**, *28*, pp 916–924.

6. Grosland, N.M.; Shivanna, K.H.; Magnotta, V.A.; Kallemeyn, N.A.; DeVries, N.A.; Tadepalli, S.C. *IA-FEMesh: An open-source, interactive, multiblock approach to anatomic finite element model development. Comput. Methods Programs Biomed.* **2009**, *94*, pp 94-96.

7. Gardiner, J.C.; Weiss, J.A. *Subject-specific finite element analysis of the human medial collateral ligament during valgus knee loading. J. Orthop. Res.* **2003**, *21*, pp 1098-1106.

8. Anderson, A.E.; Peters, C.L.; Tuttle, B.D.; Weiss, J.A. *Subject-specific finite element model of the pelvis: development, validation and sensitivity studies. J. Biomech. Eng.* **2005**, *127*, pp 364.

9. Grosland, N.M.; Bafna, R.; Magnotta, V.A. *Automated hexahedral meshing of anatomic structures using deformable registration. Comput. Meth. Biomech. Biomed. Eng.* **2009**, *12*, pp 35-43.

10. Magnotta, V.A.; Li, W.; Grosland, N.M. *Comparison of displacement-based and force-based mapped meshing. Proceedings of the Computational Biomechanics for Medicine III MICCAI*; 2008.

11. Frey, P.J.; George, P. *Mesh generation: Application to Finite Elements;* John Wiley & Sons, 2007.

12. Bathe, K.J. *Finite element procedures;* Prentice Hall, 1996.

13. Brekelmans, W.A.M.; Poort, H.W.; Slooff, T. *A new method to analyse the mechanical behaviour of skeletal parts. Acta orthopaedica* **1972**, *43*, pp 301-317.

14. Huiskes, R; Chao, E.Y.S. *A survey of finite element analysis in orthopedic biomechanics: the first decade. J. Biomech.* **1983**, *16*, pp 385.

15. Benzley, S.E.; Perry, E.; Merkley, K.; Clark B.; Sjaardama, G. *A comparison of all hexagonal and all tetrahedral finite element meshes for elastic and elasto-plastic analysis. Proceedings of the 4th International Meshing Roundtable*; 1995; pp 179-191.

16. Owen, S.J. *A survey of unstructured mesh generation technology, Proceedings of the 7th International Meshing Roundtable*; 1998, *3*.

17. Lohner, R. *Progress in grid generation via the advancing front technique. Engineering with Computers* **1996**, *12*, pp 186-210.

18. Blacker, T.D.; Meyers, R.J. *Seams and wedges in plastering: a 3-D hexahedral mesh generation algorithm. Engineering with computers* **1993**, *9*, pp 83-93.

19. Schneiders, R. *A grid-based algorithm for the generation of hexahedral element meshes. Engineering with Computers* **1996**, *12*, pp 168-177.

20. Li, T.S.; McKeag, R. M.; Armstrong, C.G. *Hexahedral Meshing Using Midpoint Subdivision and Integer Programming. Comput. Meth. Appl. Mech. Eng.* **1995**, *124*, pp 171-193.

21. Keyak, J.H.; Meagher, J.M.; Skinner, H.B.; Mote, C.D. *Automated three-dimensional finite element modelling of bone: a new method. J. Biomed. Eng.* **1990**, *12*, pp 389-397.

22. Cook, W.A.; Oakes, W.R. *Mapping method for generating three-dimensional meshes: past and present, Proceedings of the ASME international computer engineering conference,* San Diego, California; 1982.

23. Couteau, B.; Payan, Y.; Lavallée, S. *The mesh-matching algorithm: an automatic 3D mesh generator for finite element structures. J. Biomech.* **2000**, *33*, pp 1005.

24. Chabanas, M; Luboz, V; Payan, Y. *Patient specific finite element model of the face soft tissues for computer-assisted maxillofacial surgery. Med. Image Anal.* **2003**, *7*, pp 131.

25. Gibson, A.P.; Riley, J.; Schweiger, M.; Hebden, J.C.; Arridge, S.R.; Delpy, D.T. *Generation of patient-specific finite element meshes for head modelling. Phys. Med. Bio.* **2003**, *48*, pp 481.

26. O'Reilly, M.A.; Whyne, C.M.; *Comparison of Computed Tomography Based Parametric and Patient-Specific Finite Element Models of the Healthy and Metastatic Spine Using a Mesh-Morphing Algorithm. Spine* **2008**, *33*, pp 1876-1881.

27. Baldwin, M.A.; Langenderfer, J.E.; Rullkoetter, P.J.; Laz; P.J. *Development of subject-specific and statistical shape models of the knee using an efficient segmentation and mesh-morphing approach. Comput. Methods Programs Biomed* **2010**, *97*, pp 232.

28. Shepherd, J.F.; Johnson, C.R. *Hexahedral mesh generation constraints. Engineering with Computers* **2008**, *24*, pp 195–213.

29. Antoine Maintz, J.B.; Viergever, M.A. *A survey of medical image registration. Med. Image Anal.* **1998**, *2*, pp 1-36.

30. Crum, W.R.; Hartkens, T.; Hill, D.L.G. *Non-rigid image registration: theory and practice. Br. J. Radiol.* **2004**, *77*, pp S140–S153.

31. Bookstein, F.L. *Principal warps – thin-plate splines and the decomposition of deformations. IEEE Trans. Pattern Anal.* **1989**, *11*, pp 567–85.

32. Besl, P.J.; McKay, H.D. *A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, pp 239.

33. Bentley, J.L.; *Multidimensional binary search trees used for associative searching. Communications of the ACM;* 1975, *18*, pp 509.

34. Zinber, T.; Schmidt, J.; Niemann, H. *A refined ICP algorithm for robust 3-D correspondence estimation. Proceedings of the International Conference on Image Processing;* 2003, *2*, pp 695-698.

35. Sproull, R.F. *Refinements to Nearest-Neighbor Searching in k-Dimensional Trees. Algorithmica* **1991**, *6*, pp 579-589.

36. Schroeder, W.; Martin, K.; Lorensen, B. *The Visualization Toolkit: An Objective-Oriented Approach to 3D Graphics*; Kitware Inc., 2004.

37. Magnotta, V.A.; Harris, G.; Andreasen, N.C.; O' Leary D.S.; Yuh, W.T.; Heckel, D. *Structural MR image processing using the BRAINS2 toolbox. Comput. Med. Imaging Graphics* **2002**, *26*, pp 251.

38. *3D Slicer*. http://www.slicer.org.

39. DeVries, N.A.; Shivanna, K.H.; Tadepalli, S.C.; Magnotta, V.A.; Grosland, N.M.; *IA-FEMesh: Anatomic FE Models-A Check of mesh accuracy and validity. Iowa Orthop. J.* **2009**, *29*, pp 48–54.

40. Kallemeyn, N.A.; Tadepalli, S.C.; Shivanna, K.H.; Grosland, N.M. *Toward Patient-Specific Finite Element Mesh Development of the Cervical Spine. Proceedings of 16th Annual Symposium on Computational Methods in Orthopaedic Biomechanics*; 2008.

41. Knupp, P.M. *Algebraic mesh quality metrics. SIAM Journal on Scientific Computing* **2002**, *23*, pp 193-218.

42. Knupp, P.M. *Algebraic mesh quality metrics for unstructured initial meshes. Finite Elements in Analysis & Design* **2003**, *39*, pp 217-241.