EECE 496: Engineering Project

Work Flow Design for Conversion of Medical Image Data to a Three-Dimensional Computer Model

Tricia Pang, #15213036

Technical Supervisor: Dr. Sidney Fels

Spring 2008

2971 W. 30th Ave.
Vancouver, British Columbia
V6L 1Z3

Prof. K. Cheung & Ms. J. Pavelich
EECE 496 Coordinators
Department of Electrical and Computer Engineering
2332 Main Mall
Vancouver, British Columbia
V6T 1Z4

April 10, 2008

Dear Dr. Fels,

Please accept the accompanying Final Report entitled "Work Flow Design for Conversion of Medical Image Data to a Three-Dimensional Computer Model".

This report summarizes the work that I have completed in the EECE 496 course, under your supervision. Over the past term, I was involved in designing a semi-automatic procedure to accurately and effectively extract a three-dimensional human tongue model from magnetic resonance imaging data. The work flow stages investigated include image viewing, pre-processing, segmentation, morphing and post-processing.

Thank you for your patience and guidance on this project. I would also like to express my gratitude towards the other faculty and students who I have supported me throughout this project.

Sincerely,

Tricia Pang

Tricia Pang

EECE 496: Engineering Project

Work Flow Design for Conversion of Medical Image Data to a Three-Dimensional Computer Model

Tricia Pang, #15213036

Technical Supervisor: Dr. Sidney Fels

ABSTRACT

This report proposes a work flow for the creation of a three-dimensional computer model from magnetic resonance imaging (MRI) data. The work flow described focuses on the extraction of a tongue model, which can then be imported into an existing 3D biomechanical model of the jaw to study medical disorders such as obstructive sleep apnea (OSA). Through the research of existing tools and the implementation of multiple MATLAB modules, a work flow including stages of image processing, segmentation and deformable transformations allows for an accurate model to be easily customized for a particular medical patient.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

GLOSSARY

Artisynth

A 3D biomechanical modelling toolkit environment used to visualize OPAL model.

Obstructive Sleep Apnea

A disorder characterized by breathing obstruction while sleeping, often caused by the collapsing of the airway's soft tissue walls.

Reference Model

A model created by a medical expert that shows an accurate representation of the anatomic structure.

Segmentation

An image analysis process in which a digital image can be divided into distinct regions for each object within the scene.

Slice

A 2D image generated by viewing a 3d image from a certain view axis. The slice directions are:
- Sagittal (vertical division dividing body into left and right portions)
- Axial (horizontal division dividing the body into top and bottom portions)
- Coronal (vertical division dividing body into back and front portions)

LIST OF ABBREVIATIONS

EECE	Department of Electrical and Computer Engineering
HCT	Human Communications Technology
MRI	Magnetic Resonance Imaging
OPAL	Dynamic Modeling of the Oral, Pharyngeal and Laryngeal Complex for Biomedical Engineering
OSA	Obstructive Sleep Apnea

1.0 INTRODUCTION

The “Dynamic Modeling of the Oral, Pharyngeal and Laryngeal Complex for Biomedical Engineering” (OPAL) Project is being carried out by the Human Communications Technology (HCT) Laboratory in the Department of Electrical and Computer Engineering (EECE) at the University of British Columbia. The OPAL project is concerned with the creation of a three-dimensional computer model of the mouth and throat region, which will be used to study various medical disorders.

The “Work Flow Design for Conversion of Medical Image Data to a Three-Dimensional Computer Model” project is an extension of the OPAL project. It aims to improve the process of extracting individual anatomic components from medical images and converting them to three-dimensional images so that the OPAL mouth and throat model can be easily customized for a particular medical patient. This work is completed under the supervision of Dr. Sidney Fels, and collaboration with various other professors and graduate students in the Departments of EECE, Dentistry and Respiratory Medicine. In particular, this project is carried out in conjunction with medical specialists in Obstructive Sleep Apnea (OSA) disorders. Their patients are prescribed to wear dental appliances to prevent the collapsing of soft tissue surrounding the airway during sleep, so they hold particular interest in changes to the anatomical structure of the tongue and jaw area with use of the appliance.

This report outlines the design and implementation of a semi-automatic procedure for carrying out image segmentation and surface mesh construction of the tongue using magnetic resonance imaging (MRI) data. Existing segmentation methods are assessed, and the most effective procedures and software tools are identified.

2.0 OVERVIEW OF WORKFLOW

The methodology in designing the workflow involves creating a rough draft of the possible workflow, and then executing each task using a variety of different tools to determine the most effective method.

2.1. Workflow

After the first set of MRI scans had been acquired, the initial task was to choose the best software for browsing the images. The following task was finding the most effective segmentation tool for extracting a region from the images. The selected tool, Livewire 3D, was then used to segment a variety of data to determine which scan settings are easiest to segment. The resulting feedback was provided to the project team to facilitate the decision on which MRI protocol settings to use for the following two sets of scans.

Subsequently, the segmentation was improved on by pre-processing the image prior to segmentation. Cropping, interpolation and sharpening of the image all assisted in a more efficient and accurate segmentation.

After segmentation, there were two methods for removing noise from the segmentation output. The typical approach is post-processing the image, but another possibility for achieving a reasonable model is morphing an existing reference model into the segmented image. The latter approach was taken and resulted in a smoother surface mesh that, after application of an affine transform and registered points, should be easily fit into an existing 3D biomechanical model in the Artisynth environment, so that simulations can be performed.

The final workflow is summarized in the following figure.

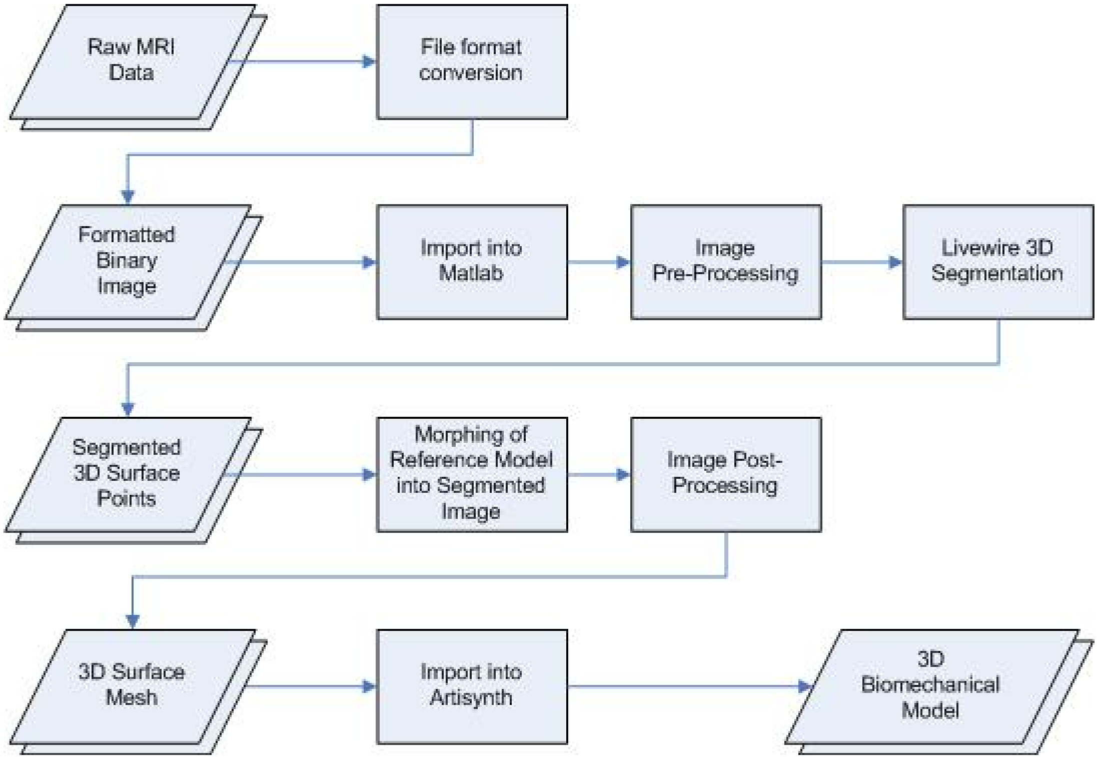


Figure 1: Diagram of Proposed Workflow

2.2. Software Tools

MATLAB is the primary software application used, as it has excellent capabilities in image processing. Furthermore, Livewire 3D is built on MATLAB, and MATLAB also supports MATITK, a medical imaging processing library toolkit. As the process is designed for use by users in the medical field, to be carried out by for numerous patients, another aim is for the procedure to be quick and easy to use. By using MATLAB, most actions can be automated and executed within the same application.

Post-processing can be carried out in Amira, a software platform specializing in the visualization and manipulation of bio-medical data.

3.0 DESIGN & IMPLEMENTATION

Each stage of the workflow outlined in the section above was investigated to determine the most effective and efficient methods of carrying out the procedure.

3.1. MRI Scan and Image Viewing

An MRI data set consists of a series of images where each image shows a thin slice of the subject. These images are all taken along a specific axis at equally spaced intervals. As shown in the following diagram, the result is a three-dimensional image, but due to the spacing intervals, the two rightmost views show losses in data.

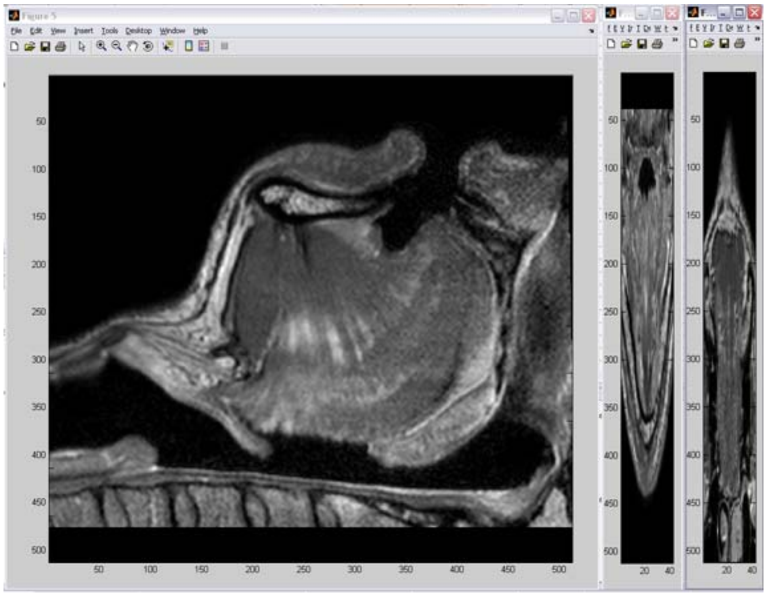


Figure 2: Sagittal MRI data set with views from all three slice directions

MRI scanning is the imaging technique of choice as it is a relatively safe procedure, compared to the high levels of radiation that can be absorbed from CT scans and X-ray scans. Moreover, MRI scans show a noticeable gradient contrast between brightly-coloured tissue and dark-coloured bones.

Three sets of MRI scans have been performed on one of Dr. Fel's graduate students to represent a subject without the sleep disorder. The initial scan on December 19, 2007 was used as a trial to determine the best MRI protocol settings to use for future scans. The second set was performed on February 20, 2008 with a rest appliance, and a third set was performed on February 27, 2007 using a protrusion appliance.

To view and analyze images, various applications have been tested. Three different programs are needed:

1. An application for reading raw DICOM images, which allows for preview and browsing of images
2. An application to convert the DICOM images to Analyze format, as this format is easier to use with MATLAB
3. An application for viewing images in Analyze format

The sections below describe the top software application for each of these purposes, and the optimal scan protocol settings for segmentation. A more detailed analysis of each application is located in Appendix A-1.

3.1.1. DICOM Viewers

OsiriX (Mac OS X only) and eFilmLite (Windows) are the top GUI applications for browsing through raw MRI DICOM images, in the format provided by UBC Hospital. They provide simple methods to navigate between images and series, show images as thumbnail tiles, view header data, and adjust image settings.

3.1.2. Conversion from DICOM to Analyze Format

The images are converted to Mayo/SPM "Analyze" Format so that they can be processed by MATLAB. MRIConvert is the only tool found that correctly converts DICOM images to this format. A useful feature of this tool is the ability to automatically perform the conversion on several directories of files, without the need for any user intervention.

3.1.3. Analyze Format Viewers

MRIcro is the top GUI application for viewing images in Analyze format. Its slice viewer tools are useful for the manipulation of images. Instructions on using MRIcro to view the Analyze images are located in Appendix A-2.

3.1.4. MRI Scan Protocols

Feedback on the scan protocols was given project team after segmentation was completed on the initial set of images to determine the settings performed the best.

Furthermore, inversion recovery (IR) scans have been performed to suppress the tongue muscle. In theory, an image without the tongue present can be subtracted from a normal T1 image to result in a segmented tongue. After the scan was performed, the method of image subtraction was not investigated further, but should be revisited in the future as it has potential to perform a good segmentation.

3.2. Image Pre-Processing Techniques

Image processing prior to segmentation is done entirely in MATLAB. After the image is loaded, it is cropped and interpolated, and other techniques are applied to aid segmentation. The majority of the MATLAB code is run automatically with minimal user intervention. Suggestions on image processing were taken from Gonzalez, Woods and Eddin's Digital Image Processing Using MATLAB [1].

Appendix B contains a summary of all MATLAB scripts written for image pre-processing, and additional code snippets.

3.2.1. Loading Analyze Images

The module readanalyze.m [2], obtained from the MATLAB Central File Exchange, loads the analyze images into MATLAB. The file readanalyze() file parses the header file of an Analyze image and reads the image data into a matrix.

3.2.2. Cropping

MATLAB easily runs out of memory when performing operations on large data matrices, so cropping the file at this stage allows for further interpolation in another axis. Reducing the matrix size also reduces the processing time of later calculations.

The script Crop.m has been implemented to crop every slice of the image according to user-defined boundary lines. The image of a single slice before and after cropping is shown below.

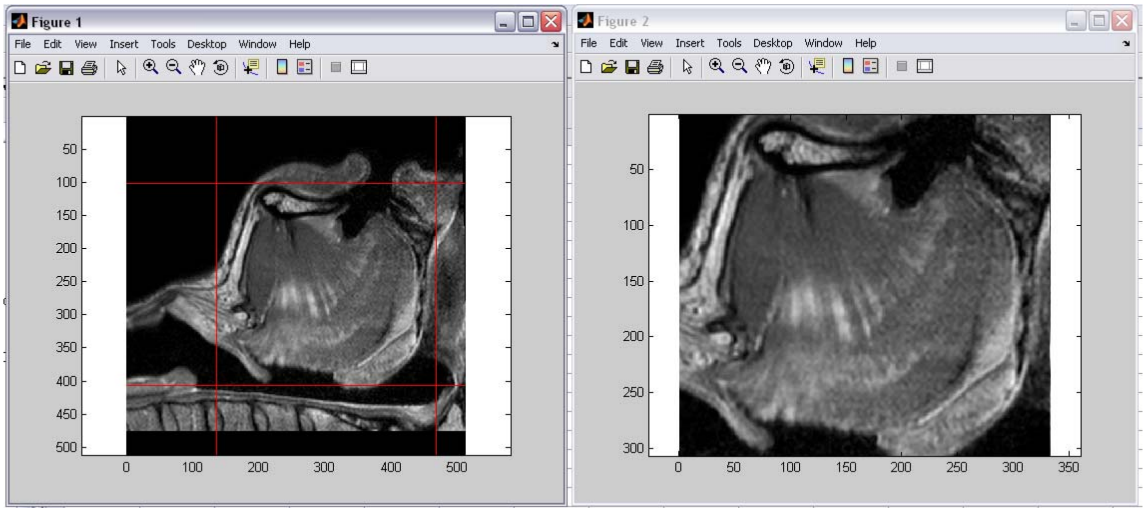


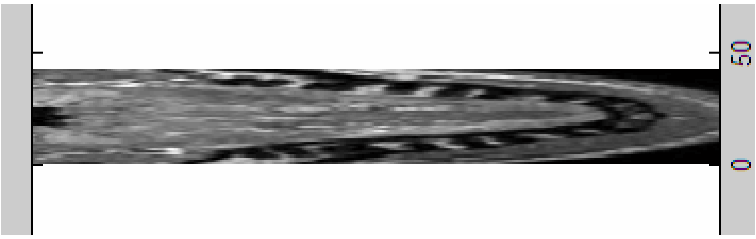
Figure 3: Cropping Tool used to Reduce the Dimensions of Each Slice

3.2.3. Interpolation

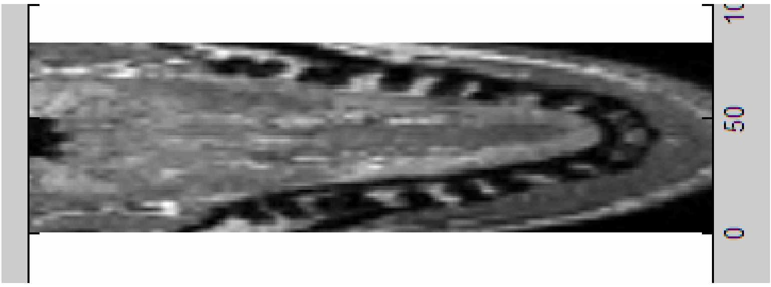
The script Interpolate.m is implemented to interpolate the image. Since the slices on the two auxiliary views are compressed, as shown in Figure 2, interpolation on the minor axis is required for a good segmentation. For example, if the Sagittal view describes the x-y plane, more slices along the x-z and y-z planes need to be introduced.

The interpolation amount is limited though, as an out-of-memory problem often occurs if the number of additional slices is too high.

A shortcoming occurs because the interpolation is not infallible and cannot perfectly predict the slices that are missing. As shown in the figure below, interpolation is definitely helpful for segmentation, but the original scan in that slice direction is by far clearer. In Figure 4(b), the soft tissue between the teeth seem to merge into the tongue, but in Figure 4(c) it is obvious from the actual scan that there is actually a gap between the tongue and the teeth.



(a)



(b)

(c)

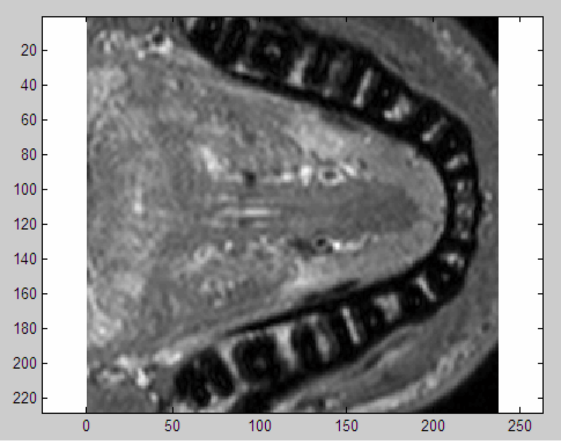


Figure 4: Axial View: (a) Unaltered slice from Sagittal scan; (b) Slice from Sagittal scan after interpolation; (c) Slice from Axial scan

3.2.4. Sharpening

The script Sharpen.m sharpens the through the use of a high pass filter. A Laplacian low-pass filter is applied to the image, and the result is subtracted from the original data set to obtain a sharper image. Although the noise is slightly enhanced, the gradient difference is greatly improved, which results in a smoother segmentation.

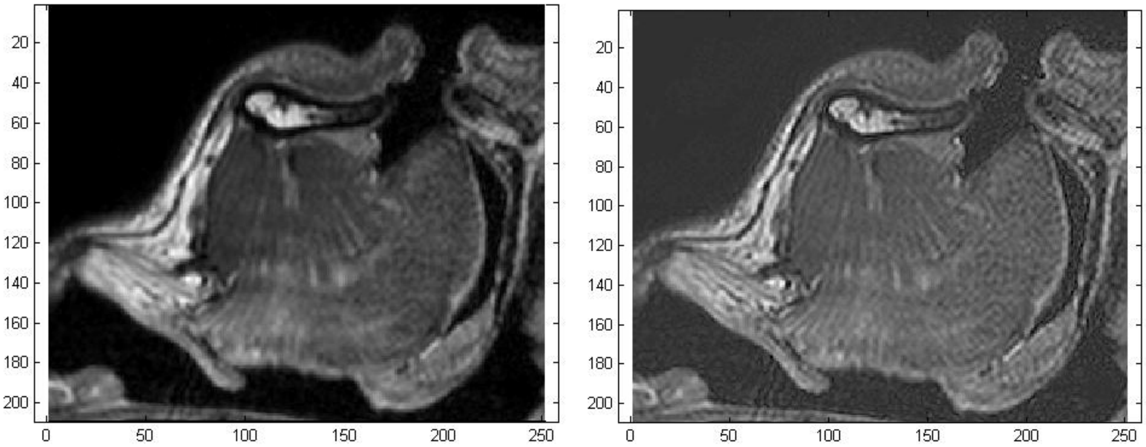


Figure 5: Images Before and After Application of a High Pass Filter

3.2.5. Noise Reduction

Noise reduction is applied through Gaussian filtering and non-linear diffusion, but these techniques are not very useful. Although they successfully reduce noise, they decrease the contrast in the edges along a gradient, resulting in an inferior segmentation result.

3.2.6. Brightness & Contrast

Changing the brightness and contrast is of no benefit to the segmentation as it may help the user see the outlines, but it has no effect on the segmentation tool's ability to detect an edge. The brightness and contrast of a slice can be adjusted from within the Livewire segmentation tool.

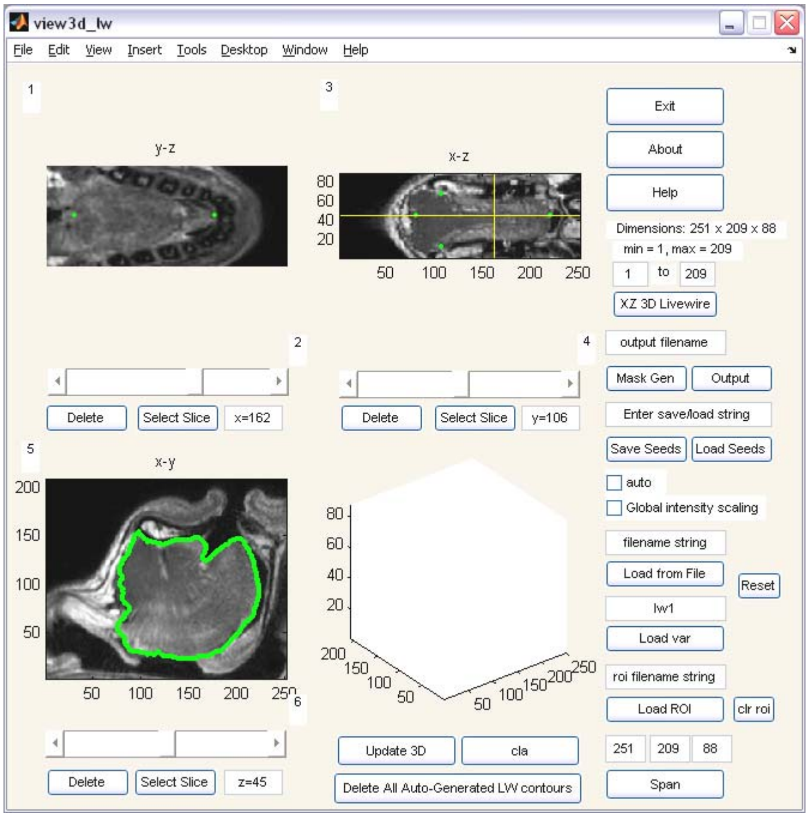
3.3. Image Segmentation Techniques

Multiple software applications for performing segmentation have been tested, such as Amira and 3D Doctor, and the results are similar to those concluded by Marko Marjanovic in his report "A Process For Converting a Set of Image Slices Into a Segmented 3D Surface Mesh" [3]. Results are typically very noisy and contain perforations and artefacts.

A tool that has not been reviewed by Marjanovic is Livewire 3D, which proves to be efficient and relatively accurate in segmenting image data as it performs well under lower resolution images with higher noise levels. Livewire 3D is an extension of Livewire 2D developed by a research groups from University of British Columbia and

Simon Fraser University. Where Livewire 2D uses a user-guided contour to outline an object on every slice in three orthogonal views, Livewire 3D requires only a few user-guided contours on two orthogonal views and then automatically mimics the process in the third dimension [4].

The figure below shows the graphical user interface for the Livewire 3D application. The user traces the object on certain y-z and x-y slices by manually selecting seed points along its boundary. These points are then automatically propagated on the x-z slice.



Dimensions: 251 x 209 x 88

min = 1, max = 209

1 to 209

x=162

y=106

z=45

251 209 88

Figure 6: Livewire 3D GUI (Segmentation Tool)

When tracing the outline of an object in the image, an active contour method is used. The user selects points along the boundary of an object, and an energy minimizing spline curve deforms to a local minima suggesting boundaries of the object. The user manually chooses which boundary to follow, but the active contour results in the need for fewer user-selected points as the gradient is followed by default. For guidance, when an outline is selected in one axis, points are shown where they intersect with slices in the other two axis.

As explained in Section 3.2.3 on Interpolation, the Livewire object tracing is far more effective in the original slice direction than on the interpolated slice directions. Figure 7 below shows a relatively smooth contour when segmenting in the original slice direction. Figure 8 illustrates the difficulty resulting from segmenting on an interpolated slice, as the contour has a tendency to bind to artefacts.

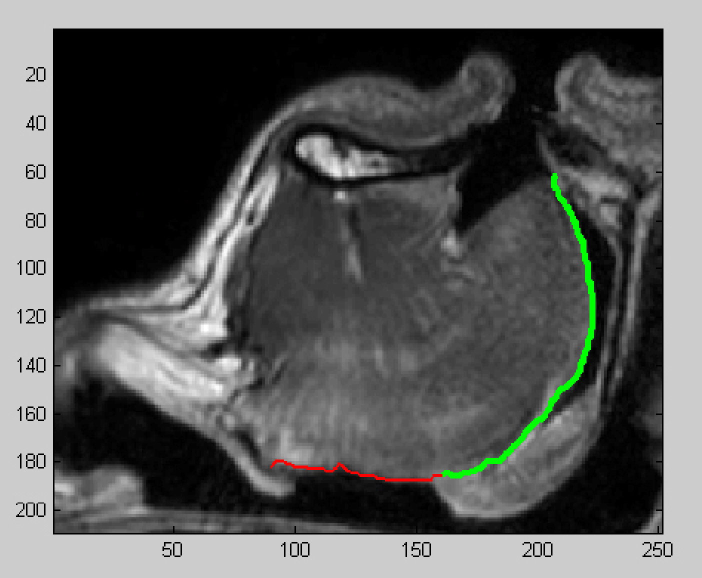


Figure 7: Livewire Segmentation in Original Slice Direction

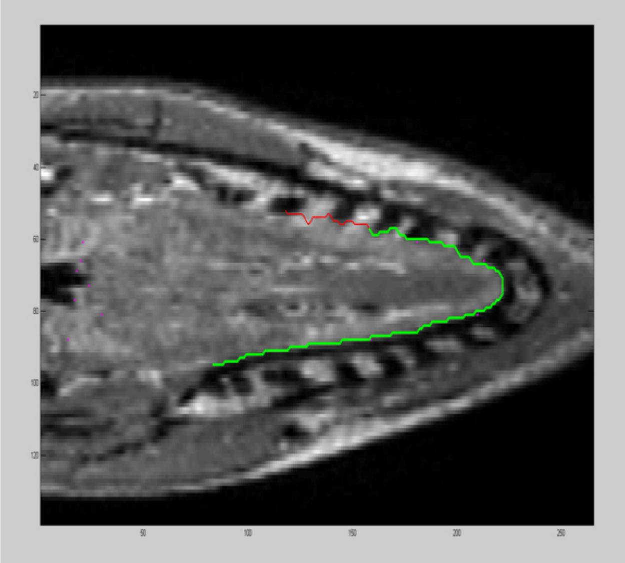


Figure 8: Livewire Segmentation in Interpolated Slice Direction (noisy in comparison)

Regardless, such complications are difficult to overcome, and among all tested applications, Livewire 3D performed better than the other tools.

3.4. Morphing Techniques

Morphing is used to transform an existing reference model into a segmented model. This method is proposed to aid in smoothening of the model and simplifies the import of the model into the existing 3D biomechanical model in the Artisynth environment. Given a good-quality reference model, morphing landmark points from the reference model to the segmented model maintains the correct tongue shape while deforming into the dimensions of the segmented tongue. An affine transform can then be applied to the result for fit into the Artisynth model.

Firstly, the Artisynth software is used to apply forces on the reference tongue until it correctly reflects the orientation and position of the tongue during the MRI scan. Gravity is positioned as if the patient is on his back, and the tongue muscles are tensed so that they push against the top of the mouth. The Artisynth model is exported, a Perl script is written to parse the file, and the resulting data imported into MATLAB to obtain the surface points for the reference tongue model.

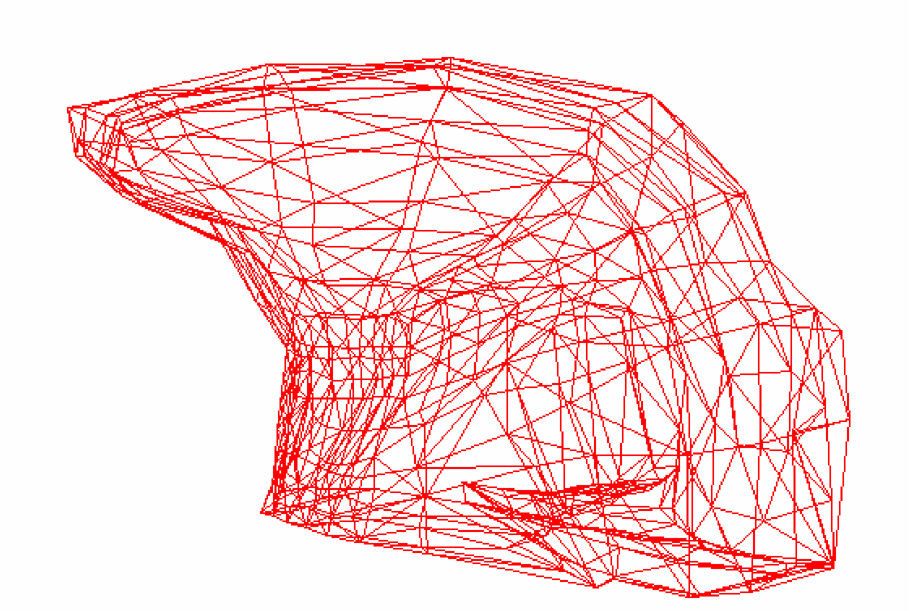


Figure 9: Artisynth Reference Tongue Model

Since the dimensions of the reference model differs from the segmented model, a MATLAB script, FitToRefModel.m scales and shifts the models until they roughly fit over each other. This improves the performance of the morph. An example of the overlaid surface points are shown in the following figure.

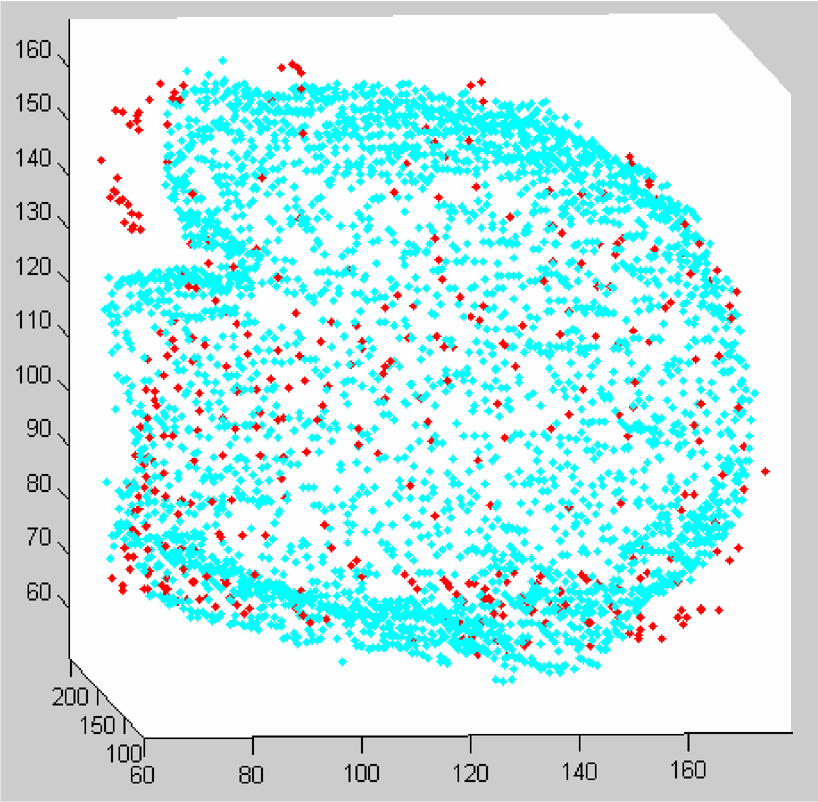


Figure 10: Overlaid Surface Points of Reference Model (red) and Segmented Model (blue)

A MATITK toolkit function, thin-plate spline interpolation, is used for morphing. This warping method requires minimal computation time (<5 minutes), and is practical since it can be run from within MATLAB. A drawback of the thin-plate spline interpolation is that user intervention required: the user must manually choose corresponding points between two models. Another alternative transform is the B-spline transform which is entirely automated, but typically takes 12-24 hours to run.

In MATLAB, a module has been implemented to allow the user to manually click points corresponding points from both the reference model and the segmented model, as shown in the following figure.

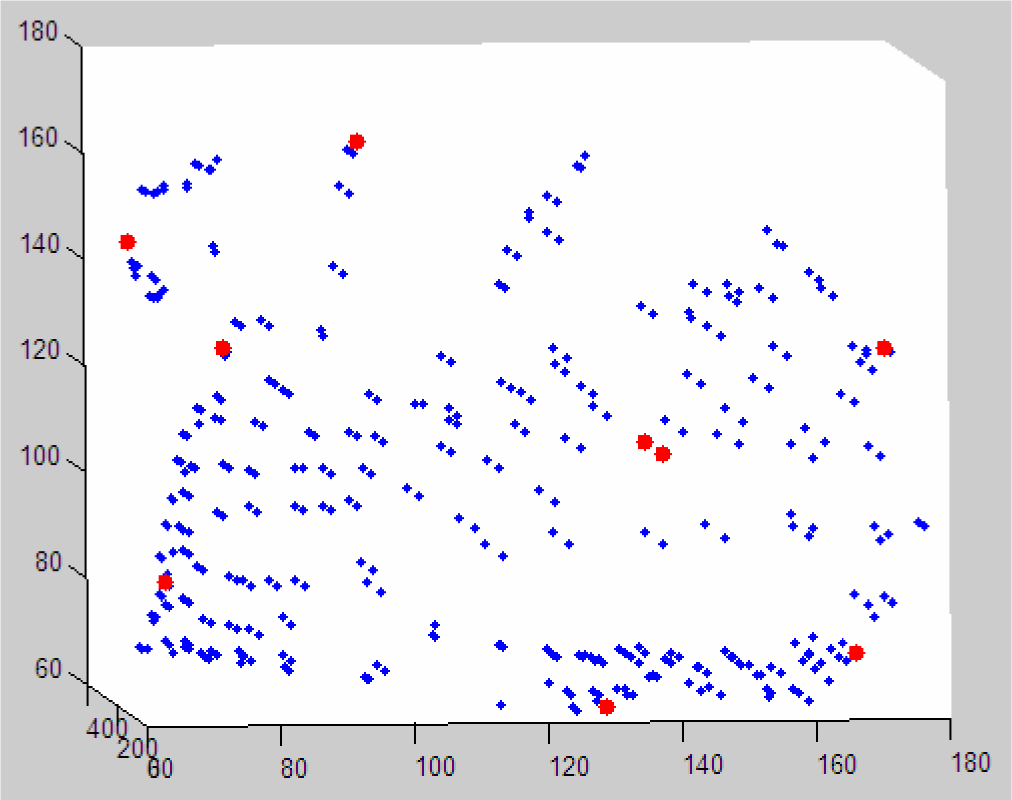


Figure 11: Tool for Manual Selection of Landmarks for Morphing

The following diagram shows landmark points of the tongue [5] that are used as a guideline for selecting corresponding points.

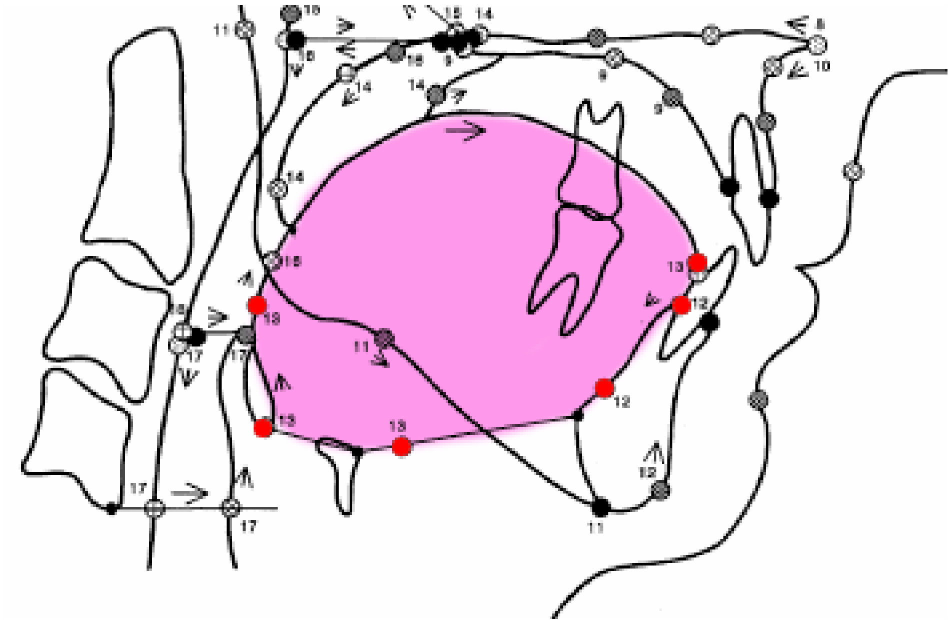


Figure 12: Tongue Landmark Points (red)

A summary of all MATLAB scripts used for morphing are located in Appendix D.

3.5. Image Post-Processing and Import into Artisynth

Amira can be used to manually perform minor smoothening, to remove any inherent noise and artefacts from the segmentation process. When complete, the final step in the workflow is importing the anatomical object into the Artisynth environment. The model must be scaled and rotated to fit into the Artisynth model, then exported as a mesh in .obj format for import to the 3D biomechanical model. The difficulty lies in automating a procedure to properly place the object into its proper position in the model, and requires much research and discussion with colleagues who have had experience in this area. These steps were not performed in this project.

4.0 RESULTS

4.1. MRI Scan Protocols

For tissue structures such as the tongue, T1 scans are easier to segment than T2 scans because of a greater gradient distinction between the jaw line and fat tissue. PD scans show the highest resolution with the most defined gradient contrast, so T1 PD scans were requested for further scans. Scans have been obtained for all directional slices (Sagittal, Axial and Coronal).

4.2. Pre-Processing

The pre-processing techniques of cropping, interpolation and sharpening are useful in improving the segmentation. These techniques are applied through an automated script in MATLAB. Execution requires approximately 3 minutes.

4.3. Segmentation

The best segmentation is obtained using Livewire 3D, but many artefacts are still present in the segmentation as a result of the poor resolution in the two non-natural slice directions.

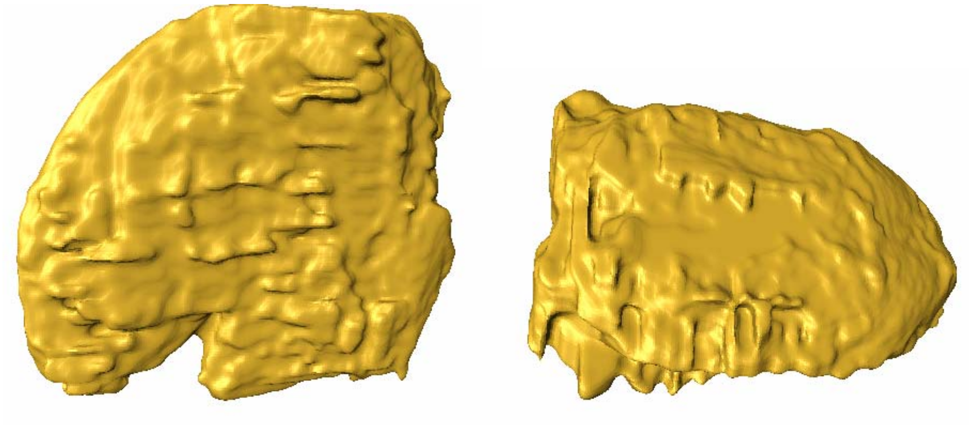


Figure 13: Livewire Segmentation Output, as viewed from Sagittal and Axial Views (noisy)

An example of the segmentation from Livewire is shown in the figure above.

On average, the best segmentations are achieved when segmenting the tongue in the Sagittal and Axial views, using 15 slices for both slice directions. Each slice requires on average 7 distinct clicks to fit the contour to the tongue boundary. Completing this takes approximately 20 minutes of manual segmentation and 5 minutes of processing and automatic point generation.

The image's original slice direction had a definite affect on the accuracy and smoothness of the segmentation. Since the segmentation has the least errors when segmenting in the original slice direction, it would be ideal if Livewire 3D were able to load images from 3 different scans, one in each slice direction. An alternative to modifying Livewire 3D is using Livewire 2D to segment slices in each direction, and combining them after segmentation by registration.

Another method for segmentation is a region-growing technique – this has been investigated but not implemented due to the short time frame of the course. In the future, it would be worthwhile to verify the effectiveness of region-growing segmentation.

4.4. Morphing

The thin-plate transform is effective in morphing the reference tongue to the segmented tongue. The manual procedure of selecting corresponding points is executed within 15 minutes.

In the future, this procedure should be more automated; for example, requiring only selection of landmarks in one image and automatically generating the corresponding point in the other. The interface for selecting points is also not easy to use, as it is hard to tell the exact location of a point due to the lack of depth in the image. The depth of a point could be shown using a cutting plane or points of different brightness.

Furthermore, experts in the medical field should be consulted for a distinct set of landmark points are that should be used. Lowe and Fleetham's paper [5] only indicates landmark points in the Sagittal direction, so additional points had to be chosen, particularly on the sides of the tongue.

A drawback discovered is that the reference tongue from Artisynth is not optimal as it takes a slightly different shape from the segmented tongue. A better result would be obtained if a better quality reference tongue were created, and if it had the same properties as the segmented tongue, such as similar muscle contraction.

The results of morphing are shown in the figures below.

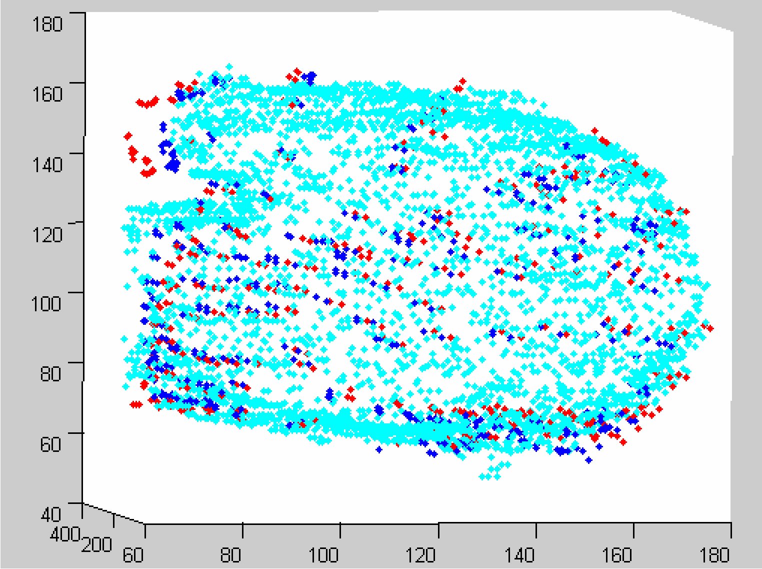


Figure 14: Result (blue) from Thin-Plate Morph of Reference Model (red) into Livewire Segmentation (cyan)

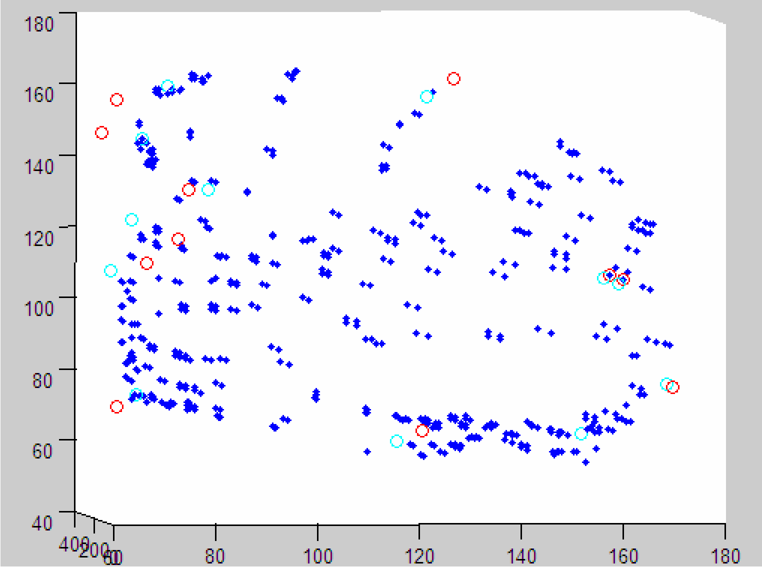


Figure 15: Result of Thin-Plate Morph (cyan), showing Landmarks Indicating the Before (red) and After (blue) Morphing Points

Figure 15 shows that the result (blue point cloud is well bound by the light blue circles (post-morph points from livewire segmentation). It is evident from the overlaid point clouds that the morphing did as was intended, but there is no automated procedure to convert the point cloud to a 3d mesh to allow better visualization of the result. This would be a good addition to the workflow.

4.5. Further Project Work

Currently there is no automated workflow for carrying out the stages of post-processing and converting the point cloud to a mesh for import – these are the next tasks in the continuation of the project.

Furthermore, a benchmark must be created by an expert in the field, so that the accuracy of result from the proposed workflow can be validated.

After the workflow has been complete, the segmentation process must be tested on other anatomical components, such as the airway, the hard and soft palate, and the epiglottis.

5.0 CONCLUSIONS

The work flow proposed in this document provides an efficient method of extracting an anatomical component from medical image data, and converting it to a three-dimensional computer model. The workflow specified can be used to complete a model for a new patient within an hour.

Pre-processing techniques of cropping, interpolation and sharpening significantly improved segmentation performed using Livewire 3D. Thin-plate morphing is satisfactory, but more research would assist in further optimizations to this method.

Further work must be completed in the stages of post-processing and conversion to mesh for Artisynth import, and a benchmark model is needed to validate the accuracy of the result.

LIST OF REFERENCES

[1] R. Gonzalez, R. Woods and Steven Eddins, "Digital Image Processing Using MATLAB." New Jersey: Prentice Hall, 2004.

[2] C. Humphries, "MATLAB Central File Exchange: readanalyze.m." July 2003. http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=3760&objectType=File

[3] M. Marjanovic, "A Process For Converting a Set of Image Slices Into a Segmented 3D Surface Mesh." University of British Columbia, December 2007.

[4] G. Hamarneh, J. Yang, C. McIntosh and M. Langille, "3D live-wire-based semi-automatic segmentation of medical images." Medical Imaging 2005: Image Processing, 2005. Proc. Of SPIE Vol. 5747.

[5] A. Lowe and J. Fleetham, "Two- and Three- Dimensional Analysese of Tongue, Airway and Soft Palate Size." Atlas of the Difficult Airway, Mosby Year Book, 1991. pp. 74-92.

APPENDIX A: MRI VIEWER ANALYSIS & INSTRUCTIONS

A-1. Image Viewer Reviews

DICOM Image Viewers

Application	Pros	Cons
1. OsiriX (http://www.osirix-viewer.com/)	• Has image browser capability! • Sid has no problems using it • Ability to view header data • Easy to adjust image settings	• For Mac OS X only
2. eFilmLite (provided with MRI scan data)	• Ability to view header data as image overlay • Easy navigation between patients, studies, series and images • Easy to adjust image settings (brightness/contrast) • Ability to view as 3D image • Measurement feature No image browser capability	
3. ImageJ (http://rsb.info.nih.gov/ij/)	• GUI and toolkit, open source • Ability to open entire series • Ability to adjust image settings • Measurement feature	• No image browser capability
4. ezDICOM (http://www.sph.sc.edu/comd/rorden/ezdicom.html)	• Simple standalone viewer GUI and command line app, with source • Can open multiple images in one directory (eg. one series)	• No image browser capability • Cannot open all images in multiple directories (eg. multiple series) • DICOM->Analyze converter batch converts, but performs

		1:1 image conversion (does not combine multiple DICOM into a single Analyze image)
5. MRIcro (http://www.sph.sc.edu/comd/rorden/mricro.html)	• Good Analyze image viewer • Slice viewer for Analyze image • Lots of useful GUI features (view header info, adjust image settings) • DICOM->Analyze converter capability	• No image browser capability • Does not recognize raw MRI image format
6. MRIcron (http://www.sph.sc.edu/comd/rorden/mricron/)	• DICOM->Analyze converter beta software • Open source version of MRIcon	• No image browser capability • Not as usable as MRIcon
7. 3D-DOCTOR (http://www.ablesw.com/3d-doctor/)	• Creation of polygonal mesh models from DICOM through manual tracing and automatic threshold segmentation	• No image browser capability Poor quality result • High learning curve • Using trial version does not allow saving •
8. Osiris (http://www.dim.hcuge.ch/osiris/01_Osiris_Presentation_EN.htm)		• Did not recognize format -- raw image required manual entry of dimensions, etc.

Conversion to Analyze/SPM Image

Application	User instruction	Pro	Con
1. MRIConvert (http://lcni.uoregon.edu/~jolinda/MRIConvert/)	1. Choose input/output directories 2. Select SPM Analyze 3. Convert all	• Batch conversion • Images correctly ordered • Directories labelled by series name • Command line version • Does not require any manual entry of settings	• No viewer capabilities
2. MRIcro (http://www.sph.sc.edu/comd/rorden/mricro.html)	1. Import -> Convert foreign to Analyze... - number of files - slice - increment=1 - volume increment=0 - volumes=1 - mosaic rows=1 - mosaic cols=1 2. Select first DICOM image 3. Save .hdr file		• Converter converts only one Analyze file at a time and requires user to indicate settings (eg. # files) • Images not correctly ordered
3. MRIcron (http://www.sph.sc.edu/comd/rorden/mricron/)	Not trialed	• DICOM->Analyze converter beta software	• Converter enabled only after modification of .ini file
4. ezDICOM (http://www.sph.sc.edu/comd/rorden/ezdicom.html)		• Does not combine multiple DICOM into a single Analyze image	• DICOM->Analyze converter batch converts, but performs 1:1 image conversion

A-2. Instructions on Using MRIcro

Download viewer

- filename.hdr: Header file containing dimensional data with some processing history information

- filename.img : Image file containing stream of voxels, whose datatype and ordering are described by the header file

Microsoft Windows Instructions

1. *Download and extract scan set files*

   Click on a scan set link above and save the file (eg. to the desktop).

   Right-click on the icon of the downloaded file, and select "extract all". This will open an extraction wizard.

   Click "next" and then choose location to extract the files to.

   Click "next" and the extraction process should begin.

   When it is complete, hit "Finish".

2. *Download MRIcro viewer*

   Download mricrosetup.exe and save the file (eg. to the desktop).

   Double-click on the downloaded file to run the installer.

   Continue accepting agreements and clicking "next" to continue. By default, the installer will install all manuals, colour maps and other files, and install to "C:\Program Files\MRIcro".

3. *Run MRIcro and open the files*

To run MRIcro, click on the "Start" menu, select "Programs", point to the "MRIcro" folder and click on the "MRIcro" icon.

Click on the "File" on the toolbar and select "Open image... [Analyze]".

On the open file dialog that appears, navigate to the directory with the extracted scan set files.

Choose from the files (all viewable files will have the extension .hdr) and click "Open".

4. View the MRI scan slices

The slices can be browsed by using the toolpanel at the left side of the screen.

Below the "Slice Viewer" heading, the leftmost three icons allow viewing of the 3 slice directions. (Note that the images on the icon will not correspond to the image -- the original scan's slice direction will always appear on the first, "axial view" icon)

- The horizontal slider below the icon allows selection of a particular slice The fourth icon from the left shows all three slices

- Clicking on the MR image of a slice will change the slice views in the other two slice directions depending on the click location.

- To view an image in another window, open another instance of MRIcro When doing so, switching through slices on one window will change the slices on the second window. To disable this, click on the checkbox "Yoke" located on the toolpanel at the left.

APPENDIX B: IMAGE PRE-PROCESSING

B-1. Pre-Processing Techniques

Load Analyze Image

```
      vol = readanalyze('mri_test.hdr');
```

Cropping

```
      vol2=vol(200:399,150:449,10:26);
```

Interpolation

(amt depends on size... may run out of memory)

```
      interpAmt=0.4;
      sizex=size(vol2,2);
      sizey=size(vol2,1);
      sizez=size(vol2,3);
      [x,y,z] = meshgrid(1:sizex, 1:sizey, 1:sizez);
      [xi,yi,zi] = meshgrid (1:sizex, 1:sizey,
      1:interpAmt:sizez);
      vi = interp3(x,y,z,vol2,xi,yi,zi,'linear');
```

Sharpening

```
      w=fspecial('laplacian',0);
      w=im2double(w);
      f2=im2double(vol2);
      g2=imfilter(f2,w,'replicate');
      new = f2-g2;
```

View Slice

```
      figure;
      imagesc(squeeze(vi(:,:,size(vi,3)/2)));
      axis equal;
      colormap gray;
```

B-2. Summary of MATLAB Pre-processing Scripts

The following scripts were created for 3D pre-processing by using commands from the MATLAB API.

Script name	Description
PreProcess.m	**Runs all pro-processing scripts in sequence**
Crop.m	Interface for choosing boundary of image and cropping it
Interpolate.m	Interpolates image in user-selected slice direction
Rotate.m	Rotates the image in 3D so that it appears in the correct orientation and the slices are correctly ordered
Sharpen.m	Sharpens image
ViewSlice.m	Allows user to view a specific slice in a given slice direction

APPENDIX C: SEGMENTATION

C-1. 3D Livewire Instructions

Note: MATLAB 7.1 or higher required

Setup:

File -> Set paths -> include .p and .m script directory

Instructions:

1. Set workspace to path that contains .hdr file
2. > view3d_lw
3. "Load from file" + filename string - Load image
    1. Enter .hdr file; or if need to switch order of slice directions...
    2. Permute instructions:

        Permute to switch order of slice direction:

        ```
        > img = readanalyze('filename.hdr'); -- analyze -> 3d matrix

        > img = permute(a,[2 1 3]) -- switch slice #1 and #2 order
        ```

        "Load var" + variable string - enter variable name
4. "Select slice" - Select contour on model #1 and #2 by manual livewire

Dots shows contour from other view

Up/down arrows adjust brightness, left/right arrows adjust contrast (if can't increase brightness any more, decrease contrast and increase brightness again)

If program stalls, click out of window and click back in

When done, hover around region, press space bar

5. "XZ 3D Livewire" - View progress

   Shows rotatable 3D frame figure

   Set marker size = 2

6. "Mask Gen" - to generate mask

7. "Output" + filename - outputs to .hdr file

8. If permuted initially, permute again

   > writeanalyze(b,'new filename.hdr', [dimensions])... get this mm from MRIcro (x y z)

9. New LW code: use surface_points_gen.m to extract seed (surface) points into .hdr file

APPENDIX D: MORPHING TECHNIQUES

D-1. Summary of MATLAB Morphing Scripts

The following scripts were created for 3D pre-processing by using commands from the MATLAB API.

Script name	Description
Morph.m	**Runs all morphing scripts in sequence**
FitToRefModel.m	Scales the point cloud (eg. Livewire output) so that it approximately has the same dimensions as the reference model.
GraphResult.m	Graphs results from morphing
ImageToPoints.m	Converts an image to a point cloud
SelectPoints.m	Allows user to choose points from a point cloud graph with visual feedback of selected points
SurfacePointsFromLW.m	Extracts surface points from the MATLAB workspace output from Livewire.
SurfacePointsFromObj.m	Extracts surface points from a .OBJ image file.
ViewSlice.m	Allows user to view a specific slice in a given slice direction