Medical Imaging

Image Generator to Support the Application of a Haptic Device for the Simulation of Arthroscopic Surgery

Bradley University

Department of Electrical and Computer Engineering

By: Renata Zabawa

Project Advisor: Dr. Thomas L. Stewart

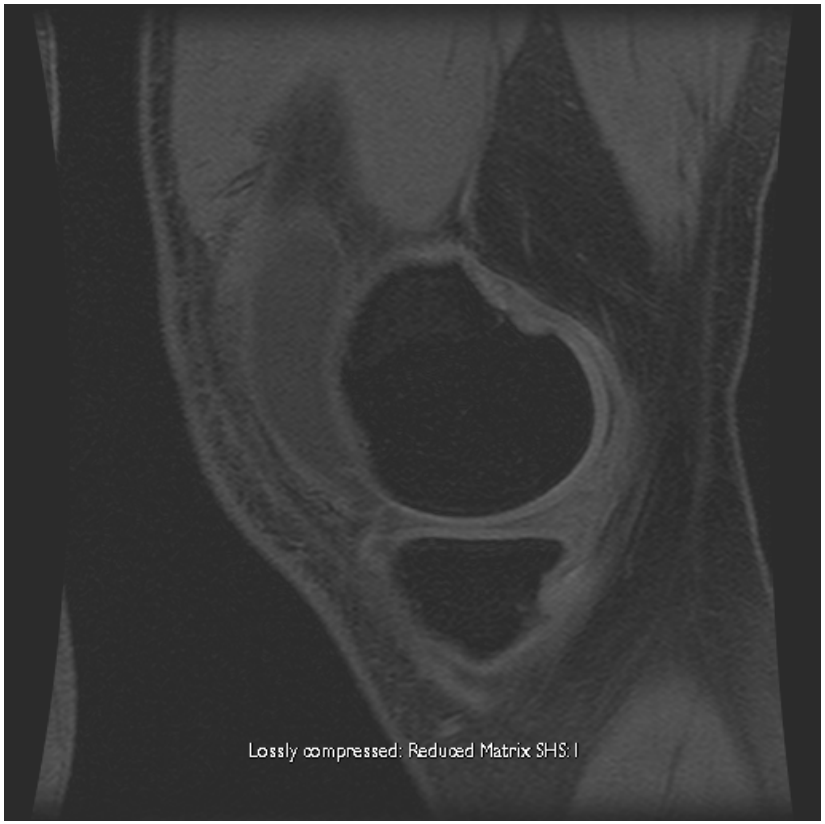
March 7, 2006

Contents

- Project Introduction
- System Block Diagrams
- Software Block Diagrams
- Original Schedule
- Tasks Completed
- Tasks Remaining
- Updated Schedule
- Questions

Project Introduction

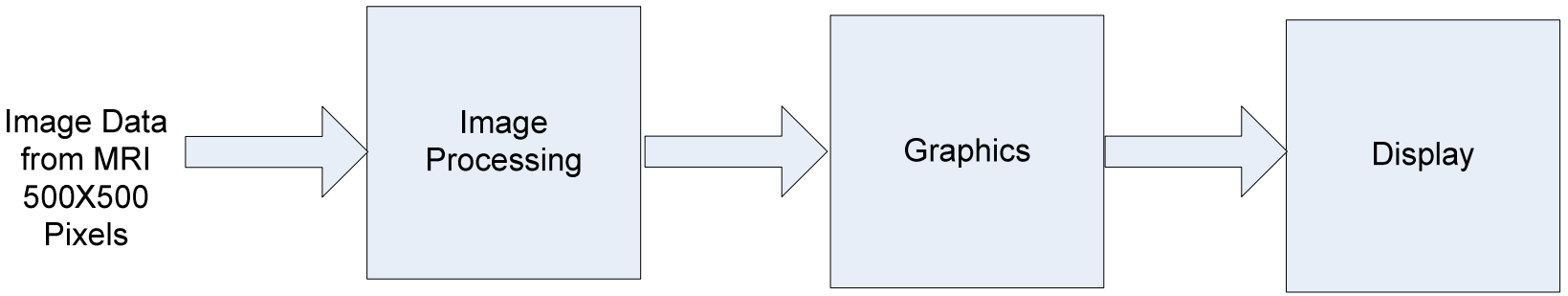
- Magnetic Resonance Imaging (MRI)
- MRI show cross section knee
- Create 3-D model of cartilage
- Simulate surgeon's view during arthroscopic surgery
- Simulate the arthroscopic surgery with a haptic feedback system



MRI Scan of Knee

Picture Provided By Dr. Stewart

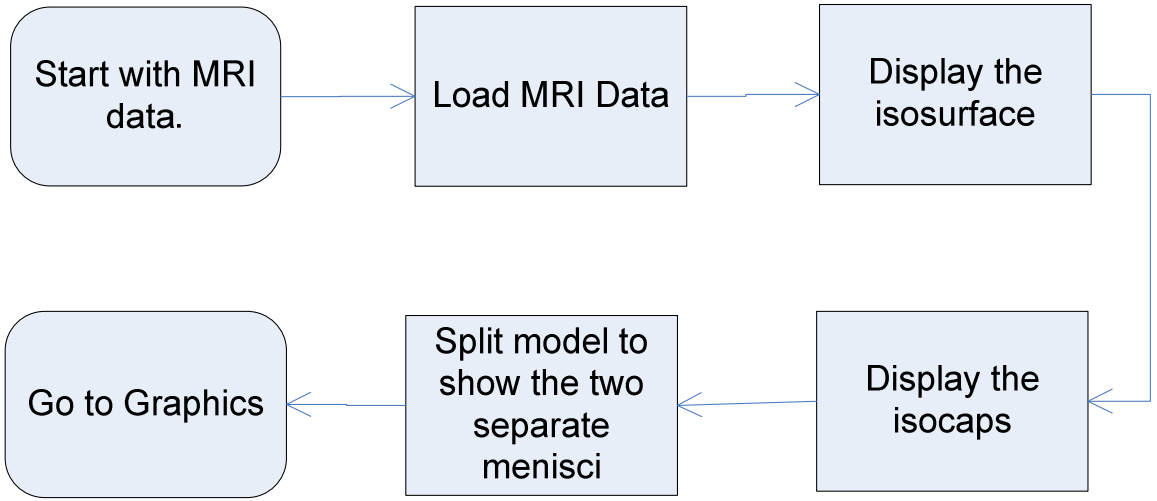
System Block Diagram



Software Block Diagrams

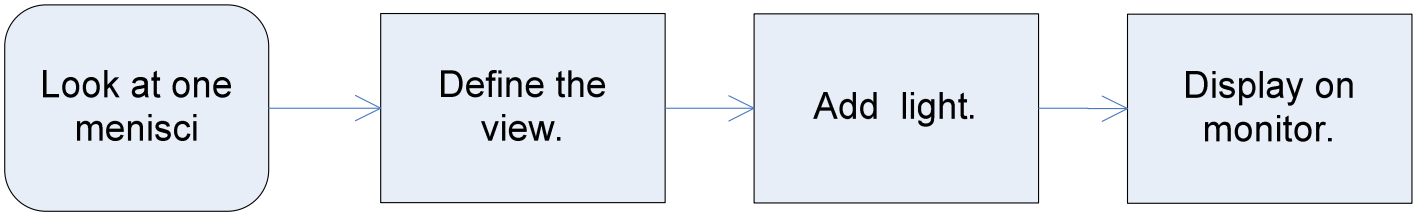
- Image Processing Block Diagram
- Graphics Block Diagram

Image Processing Block Diagram



- Creates model of cartilage
  - Isosurfaces-displays overall structure of cartilage
  - Isocaps-reveal details of interior cartilage

Graphics Block Diagram



- Takes the model of cartilage and creates the simulation of an arthroscopic meniscus surgery

Actual Views of Arthroscopic Knee Surgery

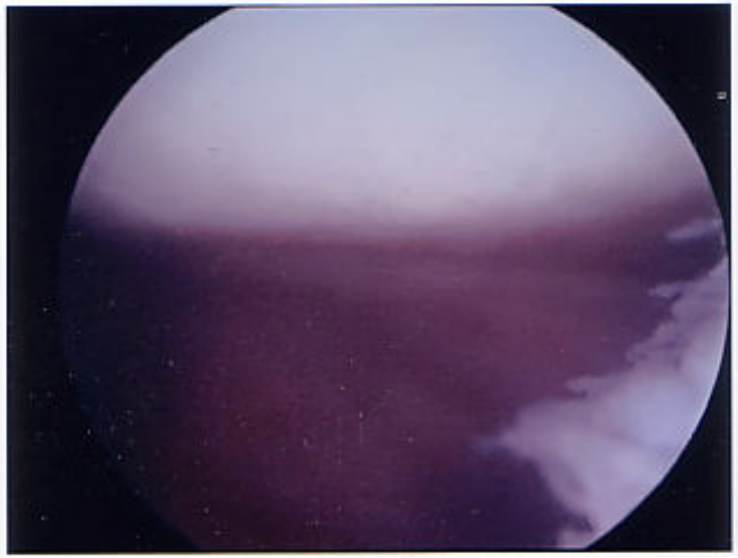


Figure 1: Arthroscopic Surgery View of Torn Cartilage

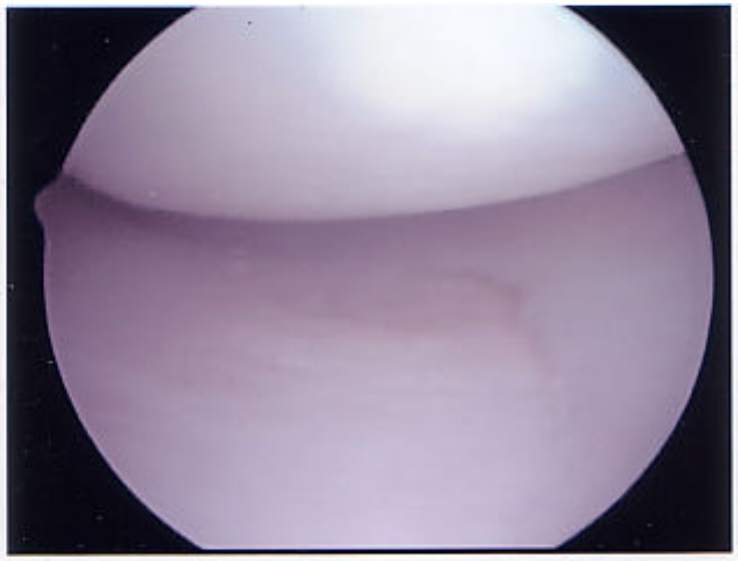


Figure 2: Arthroscopic Surgery View of Healthy Cartilage

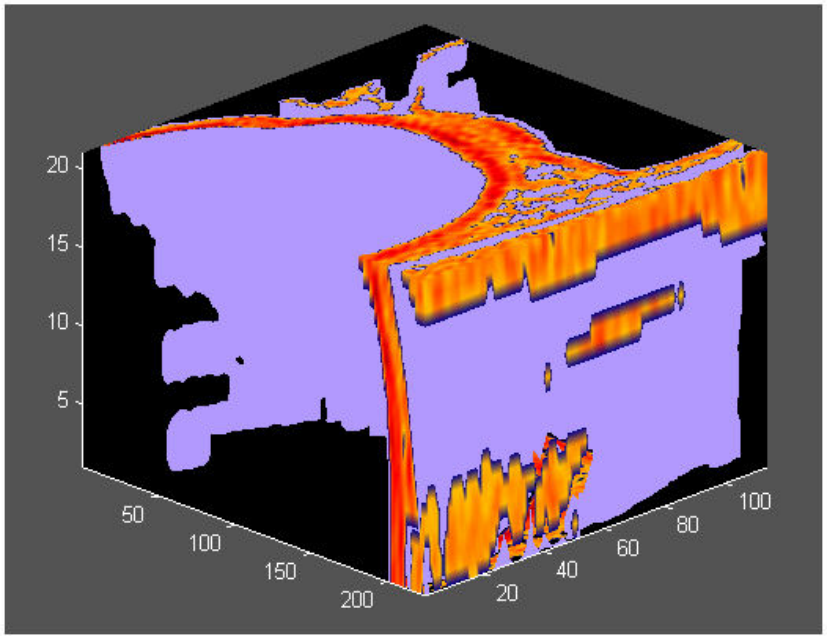
Original Schedule with Progress

Date	Task
12/06/05	Present Project
12/07/05	Study Day
12/16/05	Research rendering routines for cartilage model
01/19/06	Address rendering issues of cartilage model
01/26/06	
02/02/06	
02/09/06	Address light issues of cartilage model
02/16/06	Research light and view control on cartilage
02/23/06	
03/02/06	Address issues for the light and view control
03/09/06	
03/16/06	
03/23/06	Research haptic feedback system
03/30/06	Implement haptic device with cartilage model
04/06/06	
04/13/06	Documentation, Presentation
04/20/06	
04/27/06	

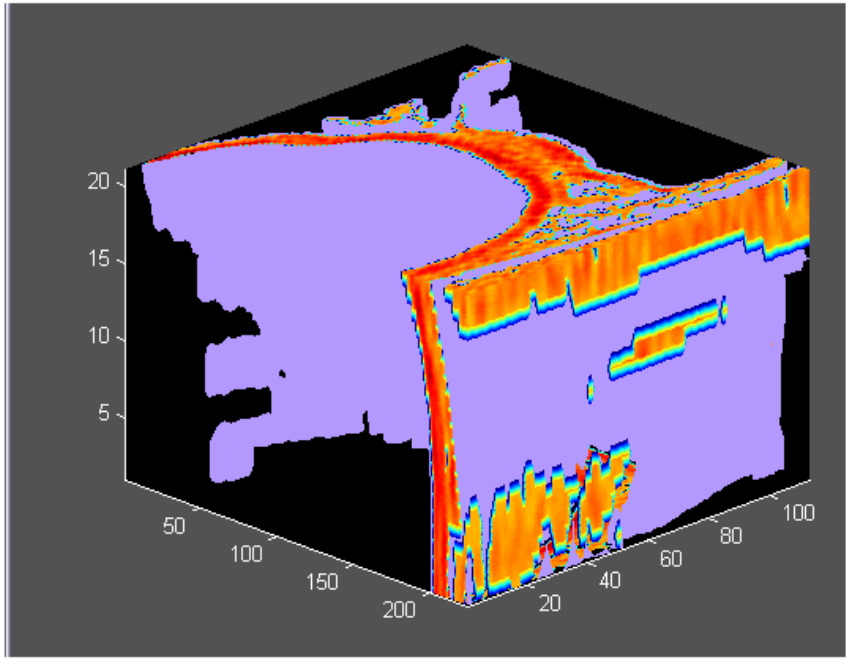
Rendering Progress

- 3 Different Rendering Routines
  - **Painters** – Original Rendering method in matlab. DOES NOT support light.
  - **ZBuffer** - Renders pixels only visible in scene. Allows light.
  - **OpenGL** – Faster than z-buffer and painters. Allows object transparency and light.

Rendering

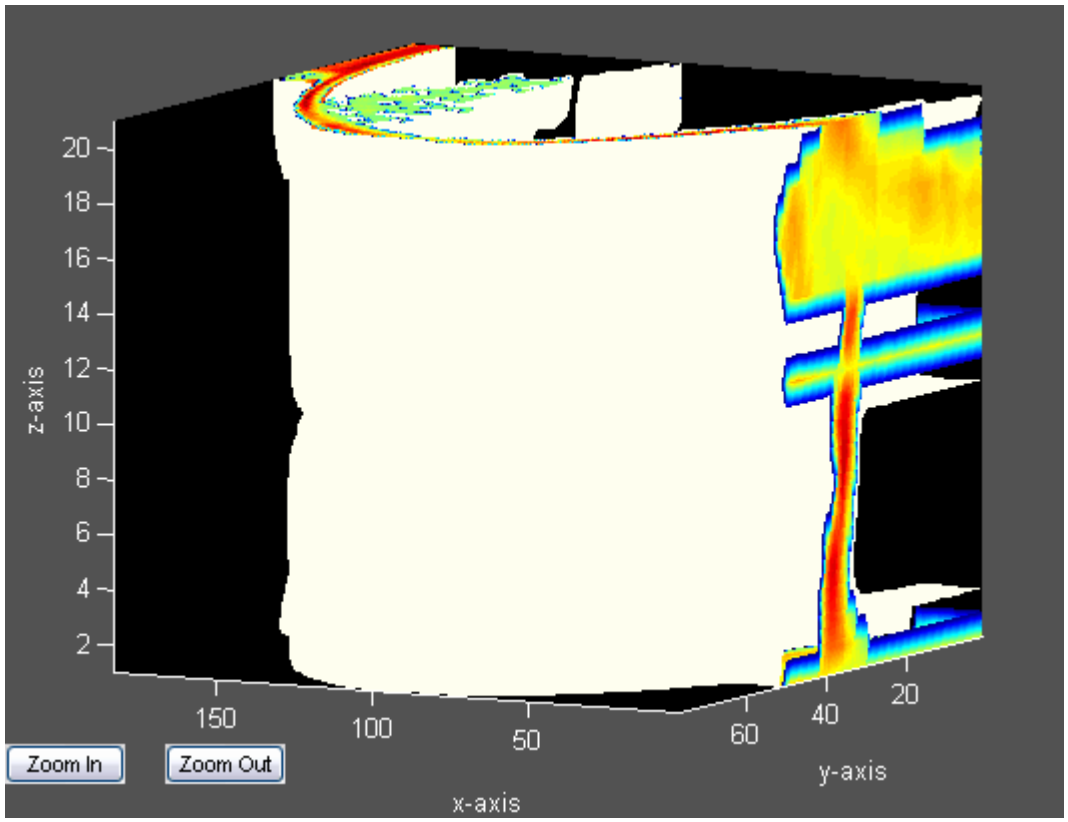


Rendering with OpenGl



Rendering with Zbuffers

Cartilage Model Split



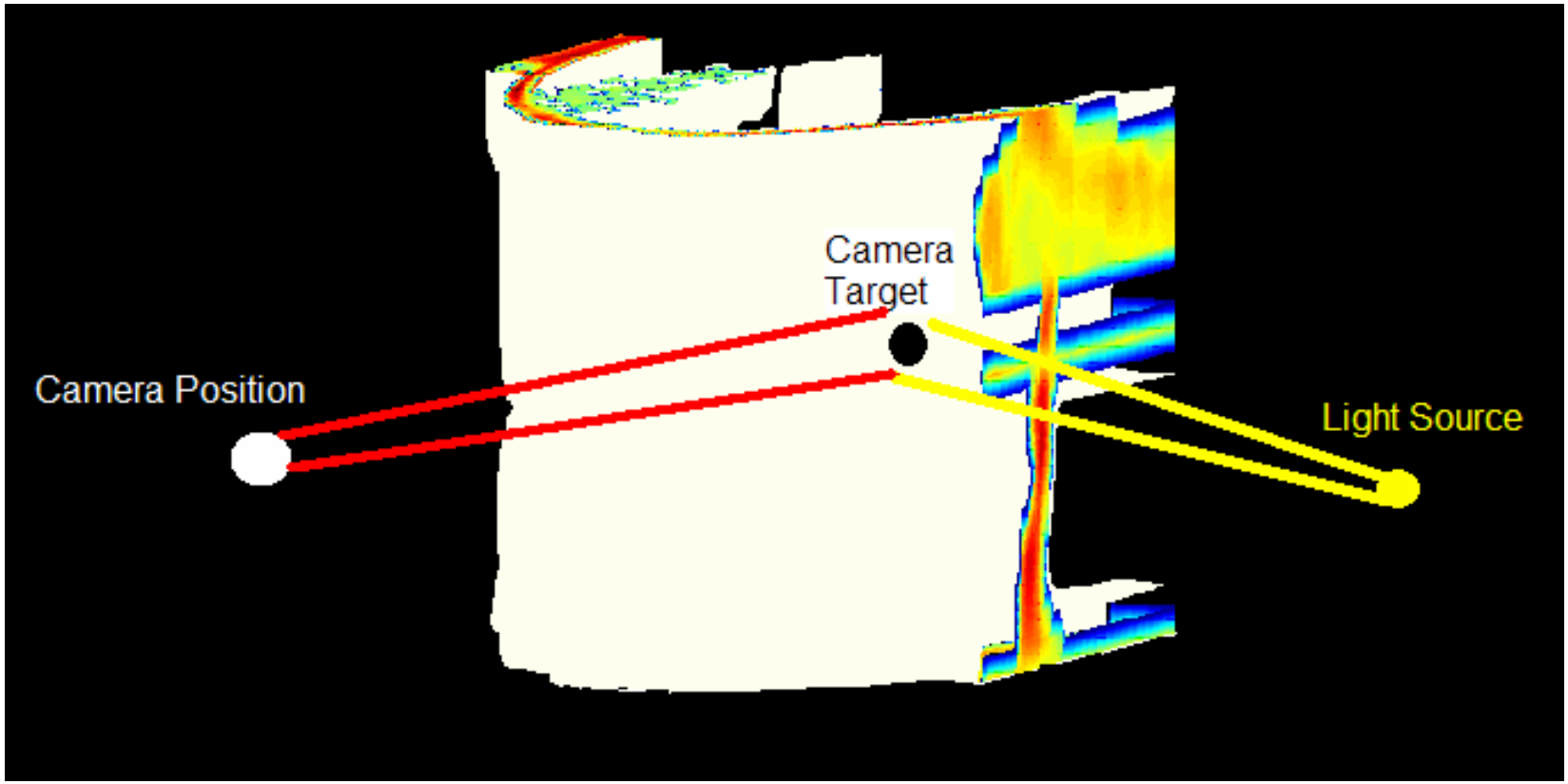
View

- Set Target
- Set Camera Position

Light

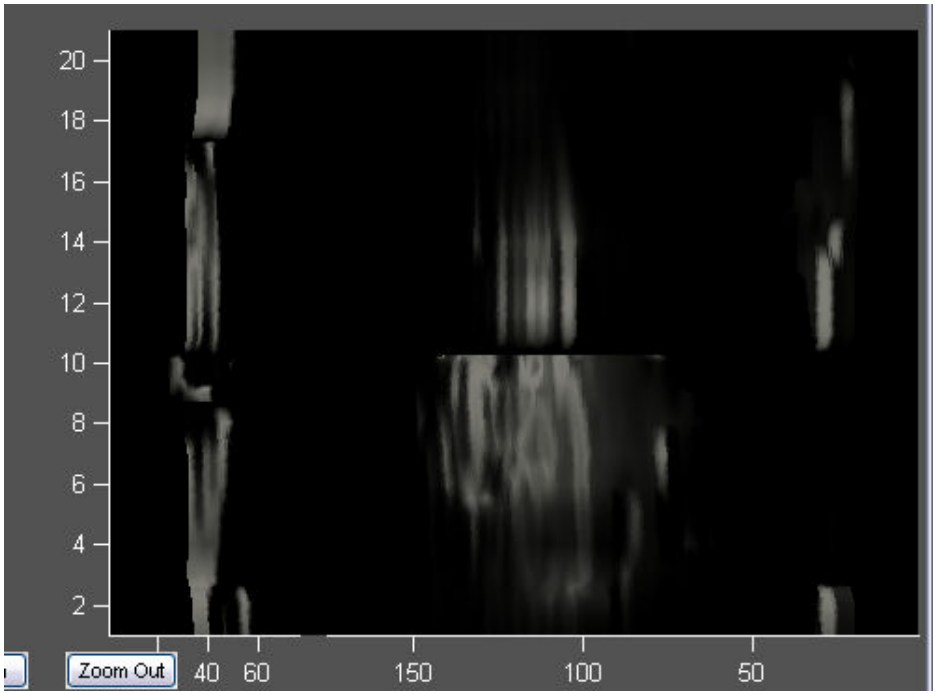
- Properties
  - Ambient Strength
  - Specular Exponent
  - Specular Strength
  - Diffuse Strength
- Set Position of Light

Cartilage Model with Light and Camera Position Set



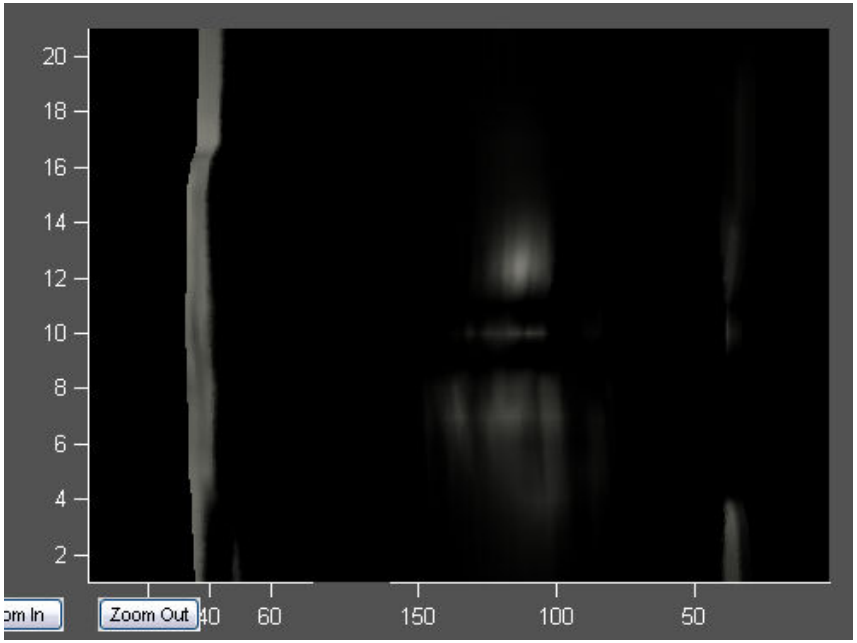
Light

- Light is Distorted

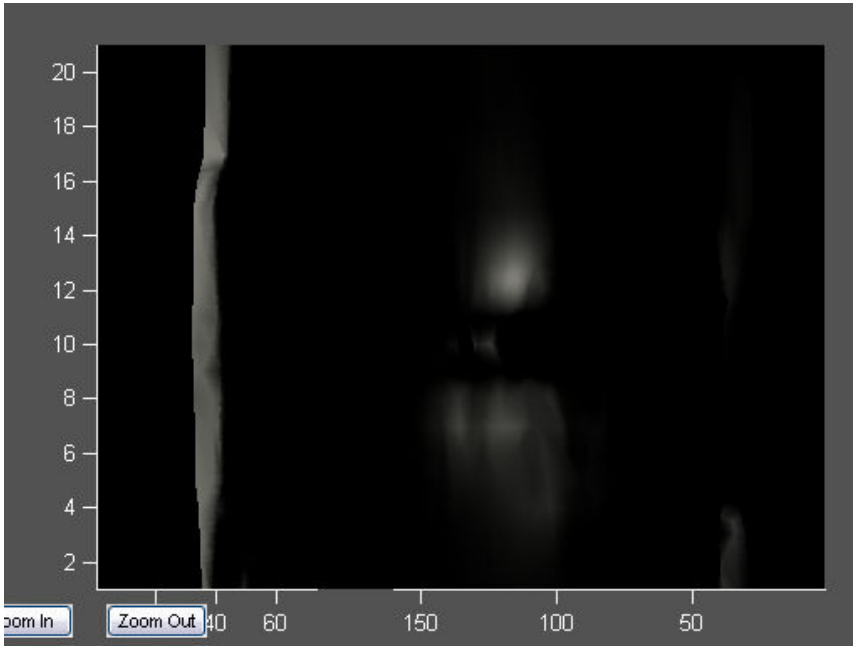


Light

- Smooth data with a mean averaging filter



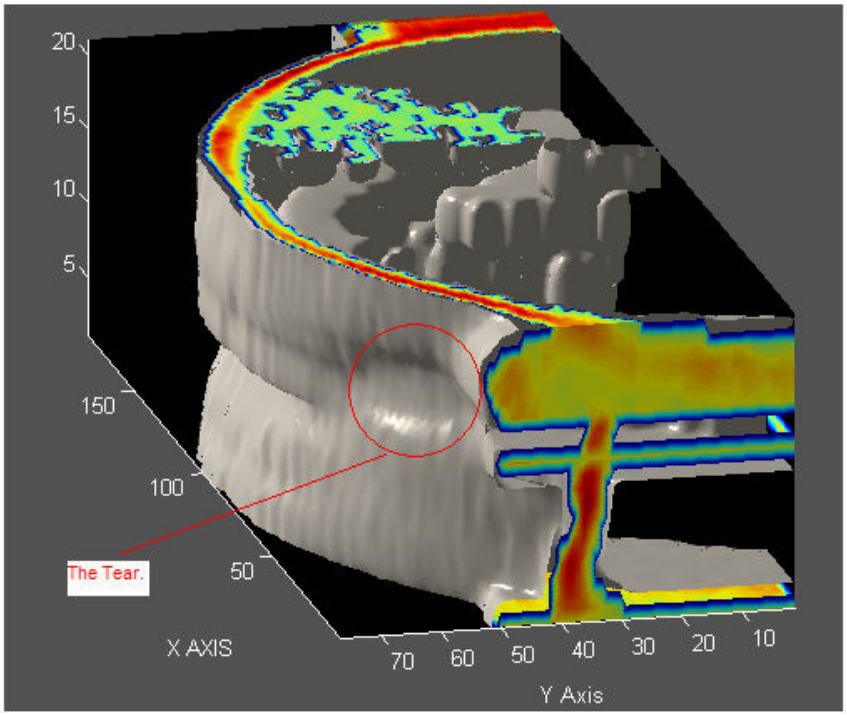
- Reduce the number of faces in patch P while trying to preserve the overall shape of the patch

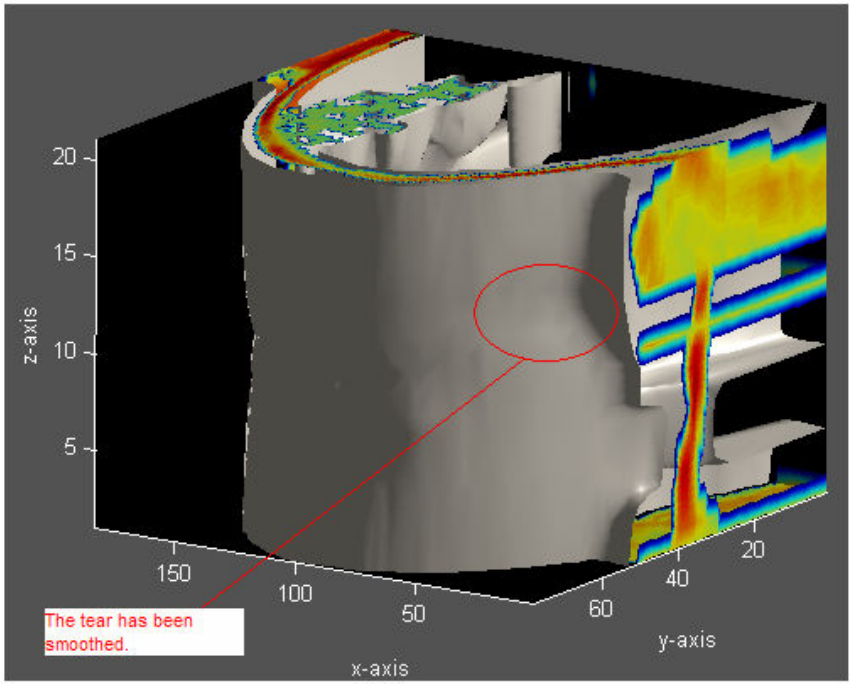


View and Lighting

- Zoom in on target

Issues for Light and View





Address Issues for Light and View

- Manually pick out the data points where the tear is
- Separate this section and smooth the two individually
- Replace the smoothed section of the tear with the manually saved data
- Smooth the overall picture

Updated Schedule

03/02/06	Address issues for the light and view control
03/09/06	
03/16/06	
03/23/06	Research haptic feedback system
03/30/06	Implement haptic device with cartilage model
04/06/06	
04/13/06	Documentation, Presentation
04/20/06	
04/27/06	

Project Summary

- Project Introduction
- System Block Diagrams
- Software Block Diagrams
- Original Schedule
- Tasks Completed
- Tasks Remaining
- Updated Schedule
- Questions

Questions?

View Femur Matlab Code

```
clc
clear
load('seg_img_femur_knee1','-mat');
newcut=squeeze(cimg6);
clear cimg4
clear cimg5
y=smooth3(newcut);
Ds = smooth3(newcut,'box',[7 7 7]);
Ds(51:62,20:40,12:15)=y(51:62,20:40,12:15);
Ds=smooth3(Ds);
figure,
whitebg('k')
```

View Femur Code Continued

```
hiso = patch(isosurface(Ds,6),...
   'FaceColor',[255/256 255/256
     240/256],...%ivory
  'EdgeColor','none',...
  'AmbientStrength',0,...
  'faceLighting','phong',...
  'SpecularExponent',3,...
  'SpecularStrength',1,...
  'BackfaceLighting','unlit',...
  'DiffuseStrength',0,...
  'SpecularColorReflectance',0);
hcap = patch(isocaps(newcut,6),...
   'FaceColor','interp',...
  'EdgeColor','none',...
  'AmbientStrength',0,...
  'faceLighting','phong',...
  'SpecularExponent',3,...
  'SpecularStrength',.5,...
  'BackfaceLighting','unlit',...
  'DiffuseStrength',0);
```

```
reducepatch(hiso, .5)
 lighting phong
isonormals(Ds,hiso);
colormap;
view(45,30)
axis tight
set(gcf,'Renderer','opengl')
xlabel('x-axis')
ylabel('y-axis')
zlabel('z-axis')
```

View Tear Light Matlab Code

```
light('Position',[30 70 14],'Style','local','Color','w')
camtarget([14 15 14])
campos([3,70,14])
```