

University of Pennsylvania
Center for Sensor Technologies

SUNFEST

NSF REU Program
Summer 2004

**INTERVERTEBRAL DISC IMAGING AND ANALYSIS
PROTOCOL ENHANCEMENT**

NSF Summer Undergraduate Fellowship in Sensor Technologies
David Jamison, IV (Engineering Mechanics) – The Johns Hopkins University
Advisors: Dawn M. Elliott, PhD and Chandra S. Yerramalli, PhD

ABSTRACT

Scientists in human tissue research often use animal tissue as alternative for human samples. Animal tissue samples are desired for their low cost and greater availability compared to human tissue. In human spine research, mouse, rat, and sheep intervertebral discs are often used as mechanical models for the human disc. They have proven effective models because they often possess mechanical properties similar to human discs. However, no one has explored whether those animal models are good geometric models of human discs. If they are, it may establish an even stronger argument for the use of these animal tissues as alternatives to human tissue. An intervertebral disc imaging and analysis protocol was devised at this lab, enabling the assessment of the geometric properties of this disc. However, additional measurements were needed so that more geometric properties could be examined. I set out to improve and expand the original protocol so that it may be used for this purpose.

Table of Contents

Introduction	3
Background	3
The Planning Stage	4
Basic Understanding of the Original Protocol	5
Devising Methods for Improvement and Expansion	5
New Protocol Execution	9
Discussion and Conclusion	10
Recommendations	11
Acknowledgements	11
References	11
Appendices	12

1. INTRODUCTION

In human tissue research, models are often used in place of an actual human tissue sample. Models are important for this type of research because they offer the scientist a way to effectively study the properties of human tissue without having the actual tissue, which in many cases is hard to come by in quantities sufficient for extensive research. Animal tissues, mathematical equations, and computer simulations can all serve as excellent human models in different ways. In the field of biomechanics, animal tissue models are often the most desired since actual mechanical tests (such as compression, tension, torsion, etc.) can be applied to the tissue to obtain its mechanical properties.

For an animal tissue to be considered a good mechanical model, the mechanical properties that are being analyzed must be similar to that of human tissue. The composition and structure of the tissue should be similar to, if not the same as, its human counterpart. In addition, an ideal model is relatively inexpensive, easy to obtain in large quantities, and has been used in other studies (justifying its use as a model).

For studies of the intervertebral disc (IVD), the most frequently used models are mouse, rat, and, to a lesser extent, sheep. Numerous studies have shown that these animals have discs with similar structure, composition, and mechanical properties to that of the human disc [1,2]

However, no literature really attempts to show if these disc models have geometric properties similar to humans. To this end, Dr. Yerramalli and Patrick Antowick at the University of Pennsylvania devised a protocol which would enable them to use a digital image of a mouse or rat disc and, using a code written in MatLab, measure the area of the sample disc. This protocol was useful; however, more was needed.

The current protocol, after my modifications, serves two purposes. The first addresses the issue of analyzing mouse, rat, and sheep discs to see if they are useful geometric models for the human spine. This will essentially strengthen the argument for the use of those animals as models for the human intervertebral disc. The second use of this protocol is comparing and contrasting the geometric properties of the lumbar and tail sections in a mouse or rat and, using these similarities and differences, try to explain the similarities and differences in the mechanical properties of those two disc sections.

This paper explains my role in the development of the protocol. It shows how I came to understand the original protocol with the help of Dr. Yerramalli, the ideas I came up with to improve the protocol, and the steps that led to the final product.

2. BACKGROUND

In spring 2004, Patrick worked on his senior design project here at this lab. The underlying goal of his project was to find a way to compare the geometric measurements of the lumbar and tail sections of the mouse intervertebral disc. These measurements would then be used to compare and explain the differences in mechanical properties of

the two sections of the mouse spine. His procedure is outlined as follows [also see fig. 1]:

1. Dissect lumbar and tail sections of the mouse spine.
2. Remove individual discs from those sections, recording the disc removed (for example, L1L2). One cut is made transecting the disc while one transects the adjacent vertebrae, thus freeing the disc from the rest of the spine.
3. Stain the disc so that, when viewed under a microscope, the nucleus pulposus can be distinguished from the annulus.
4. Use a microtome to shave off micrometer-thick sections of the disc so as to obtain a flat surface, ideal for imaging
5. Prepare a slide of the individual disc.
6. Take pictures of the disc using the SONY digital camera attached to an overhead light microscope in the lab.
7. Run the MatLab program code that takes the saved digital image of the stained disc and then calculates nucleus area, annulus area, and gross disc area.
8. Record measurements. [3]

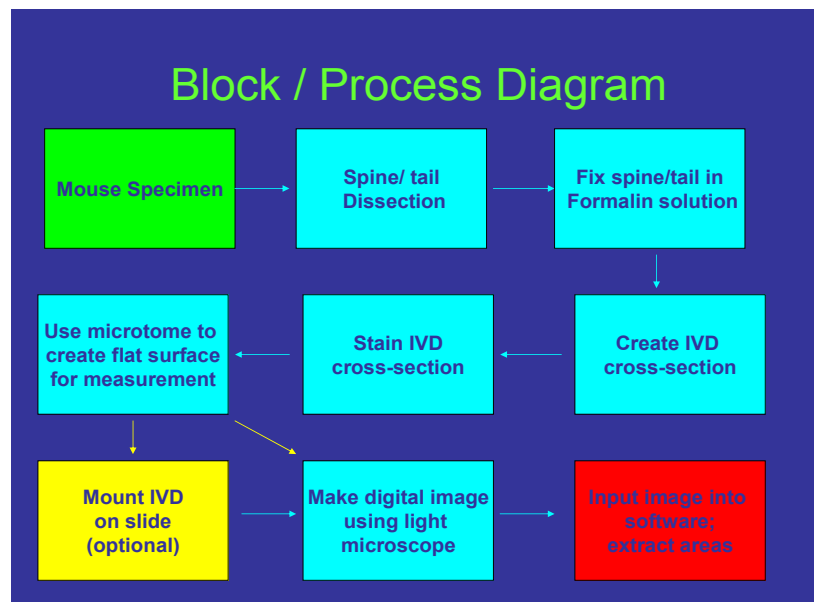


Figure 1: A block diagram of the original protocol process [3].

3. THE PLANNING STAGE

The project I carried out involved the continuation and expansion of the original IVD protocol. In particular, there were problems with the staining and imaging of the discs – many of the images are of poor quality, resulting in the inability to visually distinguish the annulus from the nucleus. Also, additions had to be made to the MatLab program so that it could make more measurements. I broke down my project into three phases: understand the original protocol, find methods of improvement and expansion, and execute the new protocol.

4. BASIC UNDERSTANDING OF THE ORIGINAL PROTOCOL

In this first phase of my project, I spoke to Dr. Yerramalli (who was the advisor for the previous project) and received his suggestions for improvement. I also reviewed the report for the original protocol to understand how it was devised and its goals, as well as see the Patrick's suggestions for improvement. His final report noted that the staining was not done very well, since the nucleus could not be distinguished from the rest of the disc.

After this I viewed the images myself and tested the MatLab program, trying to understand how it worked and what each part of the code meant. This was especially difficult because I had no prior programming experience.

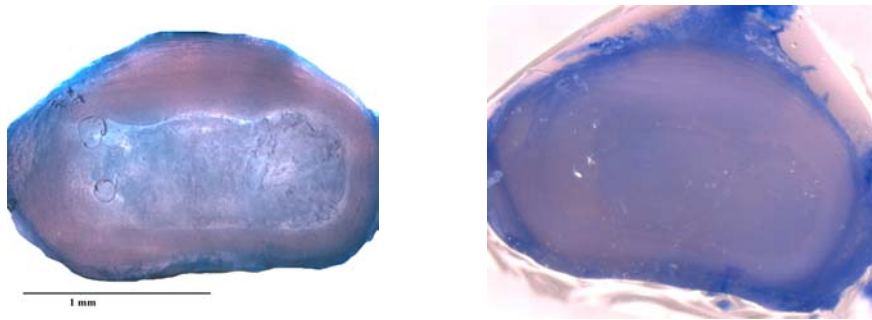
5. DEVISING METHODS FOR IMPROVEMENT AND EXPANSION

5.1 Improving the Staining and Imaging Process

The images created by the original protocol were not very useful because the nucleus could not be distinguished from the annulus in most cases. The entire disc was stained one color, making differentiation between sections nearly impossible. The original protocol called for the use of two dyes, Red Eosin and Brilliant Blue Dye. The Red Eosin dye was placed on the disc via a dropper; after five seconds the dye was washed off. The Brilliant Blue dye was then placed on the disc in the same fashion for 45 seconds before being washed off.

Given the poor staining results seen in images obtained using the original protocol, Dr. Yerramalli and I thought that we should use a more elaborate staining process, so we decided to implement the hemotoxylin and eosin staining procedure often used for immunohistochemistry. This 17-step process was employed for two reasons. First and most importantly, hemotoxylin and eosin are good at staining proteoglycans, a major component of the IVD. These stains ensure that the annulus and nucleus are distinguishable, since they contain different amounts of proteoglycans and thus will be stained slightly differently from each other. Secondly, this staining procedure calls for the stains to be left on the disc sample for 10 minutes – much longer than the amount of time that the dyes were applied in the original protocol. This new staining procedure yielded much better results than the former procedure used in the original protocol [see fig. 2].

a.)



b.)

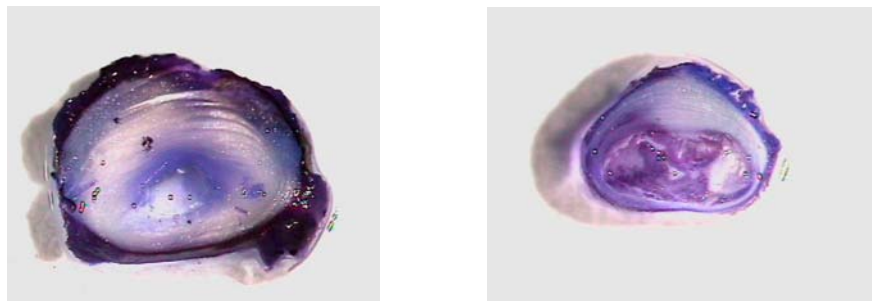


Figure 2: The two images of the discs in (a) were obtained from the original protocol. Notice the inconsistency in the pictures. In (b), the nucleus can be easily differentiated from the annulus. Also, individual lamellae can be seen in the annulus. All discs are from the lumbar region of a mouse or rat.

5.2 Improving the imaging process

The main problem with the images obtained by the original process was the lack of quality of the discs that were being imaged. I already improved the staining process to yield better visual differentiation between the nucleus and the annulus of the disc. However, another problem remained. The discs, even if stained relatively well, were not entirely in the picture. As you can see in figure 2, the disc on the left in part (a) is not fully in the picture. There are many such images taken, and would obviously be unacceptable for use. Using the SONY digital camera in the lab I was able, with good stained discs, to capture a photo that kept the entire disc in the image.

Also, I had to manipulate the focus and the lighting to obtain the best digital image from the camera attached to the overhead light microscope. I found that blocking the light coming from one of the two lamps and dimming the remaining lamp gave the best viewing results.

5.3 Modifying the MatLab code

The original disc measurement program uses a user-defined function called DiscArea. The user would type “discarea” in the MatLab the command window, along with an image for the function to be applied to and a number for the scale (how the scale was determined was never made clear in Patrick’s project report). Then the user would use the mouse to select points on the perimeter of the disc, effectively making an outline of the disc. Then this process would be applied again to the nucleus. After pressing the ENTER key, the program would calculate the area of the whole disc, the nucleus, and the annulus, give the percent of the disc occupied by the nucleus, and display a binary image [see fig. 3].

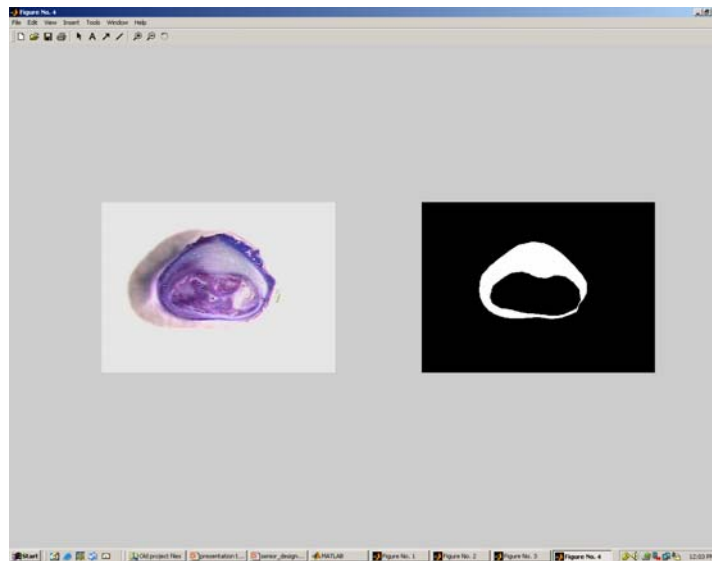


Figure 3: The original disc image on the left and the binary image on the right (showing what the user outlined and what the program uses to calculate the area measurements) come up as one figure when the DiscArea function is applied.

While the function was useful, we wanted the program to compute more measurements. In addition to the three area measurements, the percent nucleus calculation, and the binary image, I incorporated into the DiscArea function:

- A plot of the gross disc centroid and nucleus centroid superimposed on the disc image
- A plot of the gross disc and nucleus major and minor axes, also superimposed on the image
- Calculation of the distance between the two centroids
- Calculation of the lengths of all major and minor axes
- Calculation of the aspect ratio for both the gross disc and nucleus (major axis length divided by the minor axis length).

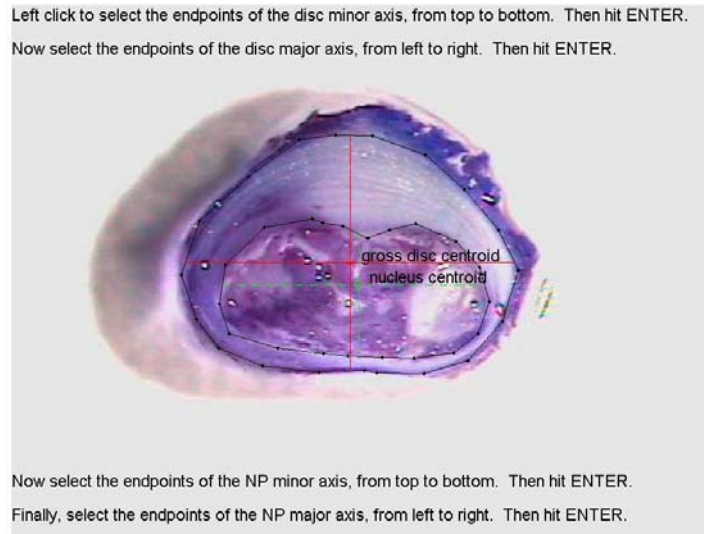


Figure 4: Output from the new DiscArea function, showing the original disc with the two centroids and two sets of major and minor axes superimposed on the disc itself. The user-selected outlines of the disc and nucleus are also in the image.

To calculate the centroid, the program takes the x and y coordinates of all the points clicked (for the outline of the nucleus and of the entire disc), computes the averages of the x and y coordinates separately, and then plots those points on the image (thus superimposing it on the disc). I included a function to make a line connecting the points that were selected by the user as the outline of the disc and nucleus. This outline was important for obtaining the major and minor axes. The user then chooses the endpoints of the major and minor axes (which are easy to spot due to the line which outlines the disc and nucleus), making sure that the axes intersect at their respective centroid. Once the endpoints were chosen, another line function connects those points, thus superimposing the axes onto the disc image [see fig. 4]. Calculations of the lengths of those axes as well as the aspect ratio were made in pixels and then expressed in the desired unit of measurement (usually millimeters).

To make the conversion from pixels to the desired unit, I created a new scale calculation. Previously, the scale was assumed to be fixed; however, if this program was to be used on many different images of disc samples from various animals, and if different cameras were used to produce the images, the scale could not be assumed to be the same for all images and thus, it would have to be specified each time. For a new scale calculation to be possible, a ruler would be required on every disc image so it could be referenced later. A line function that I wrote into the DiscArea program allows the user to manually select the length of the unit of measurement from the ruler in the image. The program then stores the length of that unit (in pixels) to be used as the pixel-to-unit conversion factor.

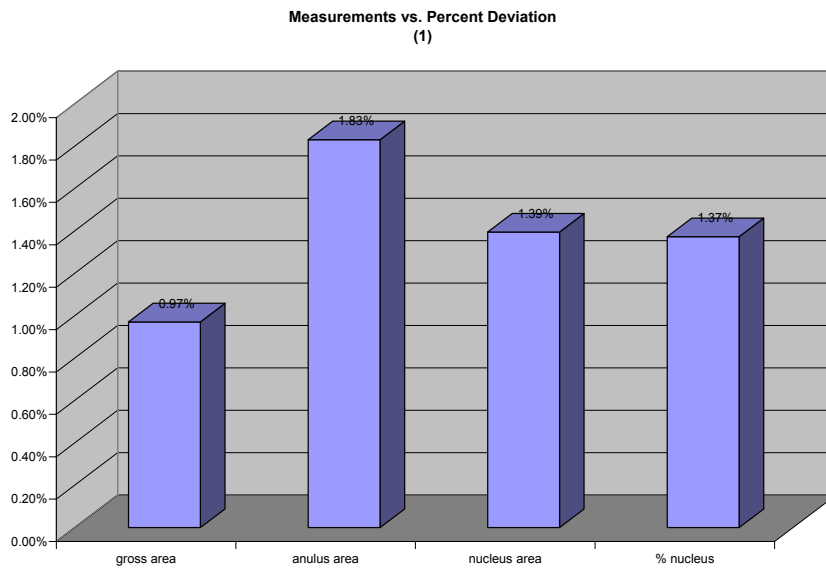
In addition to the extra calculations and measurements, I wanted to make the DiscArea function more user friendly. Originally, the directions were rather unclear and flashed on the command window before being covered by the disc image, making them unreadable. To combat this problem, I inserted an indefinite pause function into the code to halt the

running of the DiscArea code until the user presses any key, which should not happen until he finishes reading the directions; only then would the image appear on the screen. Once the figure containing the image appears, more directions are shown on the figure itself, giving the user step-by-step instructions. Figure 4 illustrates some of the on-screen directions.

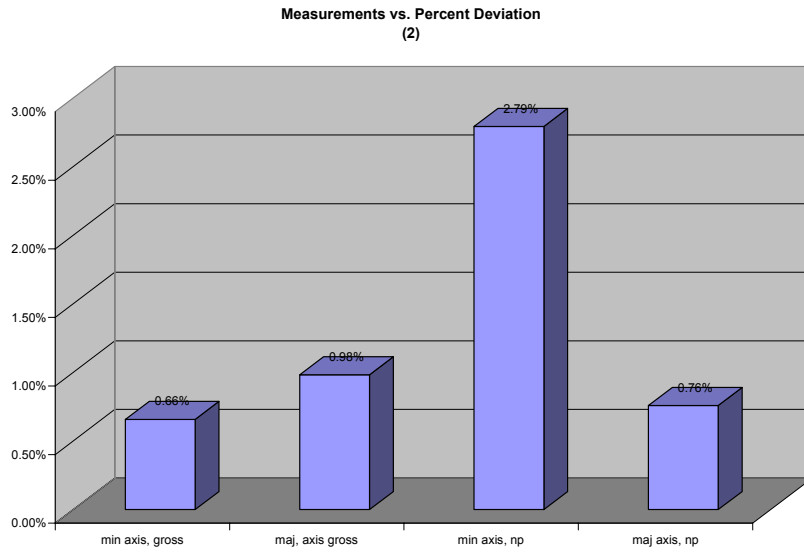
6. NEW PROTOCOL EXECUTION

Once I made all the improvements to the original protocol, I carried out the entire protocol from start to finish. The process turned out to be much more effective and useful than the original. When applying the DiscArea function to the same image multiple times, it yielded low percent deviations and showed that the calculations are accurate and repeatable [see figure 5].

a.)



b.)



c.)

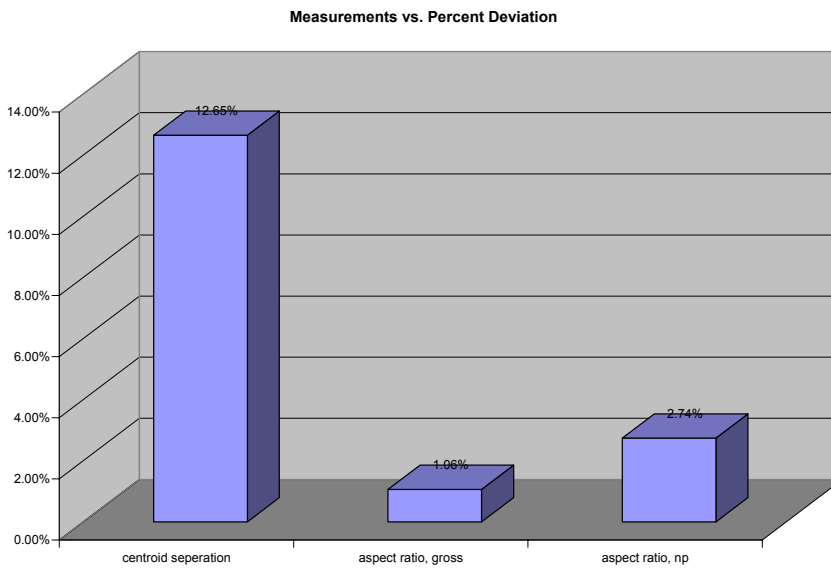


Figure 5: Here are graphs of the percent deviation for (a) gross disc area, nucleus area, annulus area and percent nucleus, (b) major and minor axes for both gross disc and nucleus, and (c) the centroid separation, and aspect ratio for nucleus and gross disc. The DiscArea function was applied 15 times to the same disc image. As you can see, only the centroid separation value has a percent deviation above three percent, and it is still on the order of 10^{-6} meters.

7. DISCUSSION AND CONCLUSIONS

By improving the disc staining and imaging, and by making additions to the DiscArea function in MatLab, the new protocol is much improved from its predecessor. Now this protocol can be applied to the many samples of rat and mouse discs that we have in the lab to see if the different geometric properties of the lumbar and tail sections of the spine correlate to differences in mechanical properties. Also, the MatLab program can be

applied to prepared samples of sheep and human lumbar disc samples as well as the mouse and rat. This will, in the end, enable us to see whether or not the animal models we use are good geometric approximations of the human intervertebral disc.

8. RECOMMENDATIONS

I suggest finding another way to calculate the position of the centroid. The position of the centroid plot depends on the spacing of the points chosen to outline the disc or nucleus (the plot is just the average of the perimeter values). For example, if the user selects many more points to the right of the perimeter, the centroid will be shifted to the right. This obviously poses problems for measurements and calculations that are based on the centroid position, such as major and minor axes positions and lengths and centroid separation.

9. ACKNOWLEDGMENTS

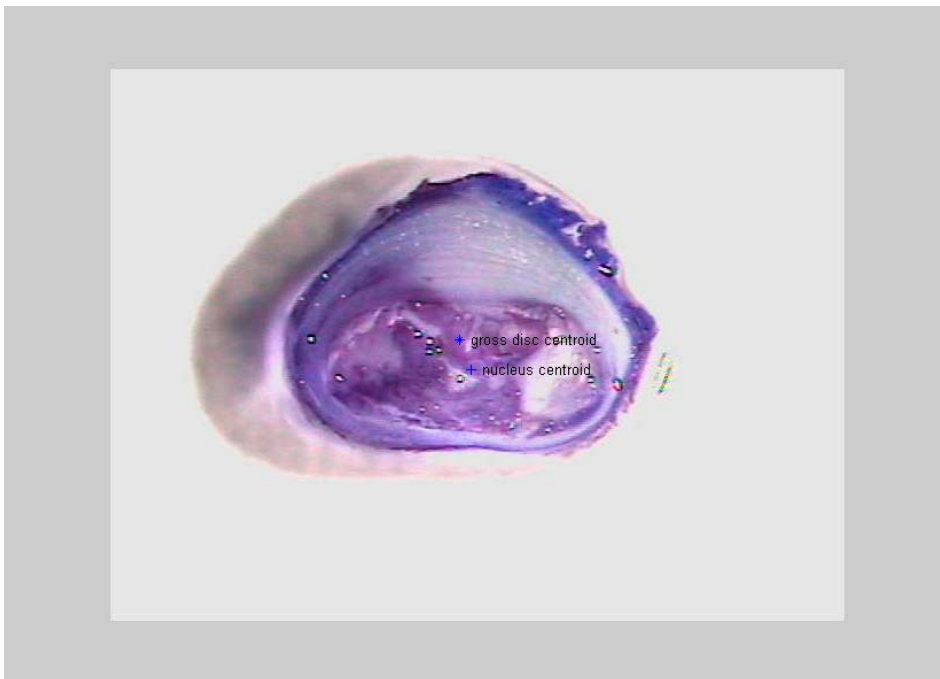
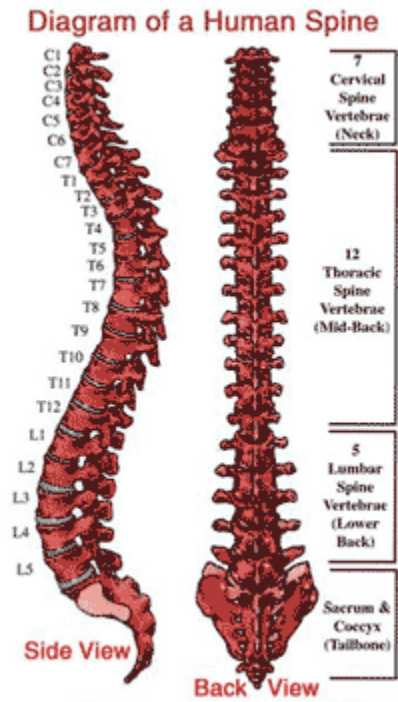
I would like to thank Dr. Louis Soslowsky, the director of McKay labs, and Dr. Dawn Elliot, my advisor, for granting me the opportunity to work in their lab this summer. Thank you to my advisors Wade Johannessen and Chandra Yerramalli for their guidance and help. Thanks to Dr. Jan Van der Spiegel, the director of the SUNFEST program. Also, I want to thank the National Science Foundation for their support through an NSF-REU grant and the Microsoft Corporation for their financial support as well.

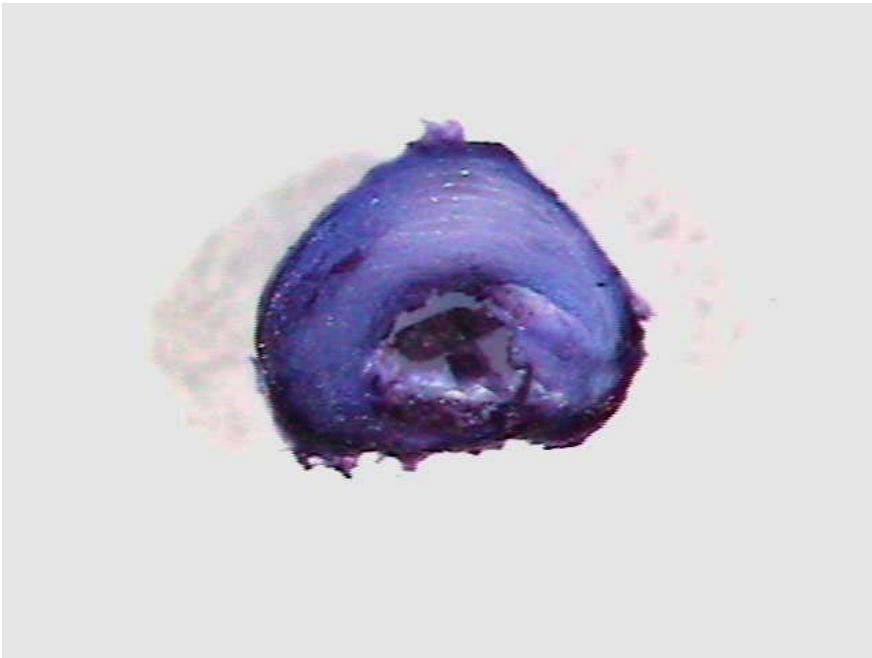
10. REFERENCES

1. D.M. Elliot and J.J. Sarver, Young Investigator Award Winner: Validation of the Mouse and Rat as Mechanical Models of the Human Lumbar Disc, *Spine*, 29, (2004), 713-722
2. H. Wilke, A. Kettler, L.E. Claes, Are Sheep Spines A Valid Biomechanical Model for Human Spines?, *Spine*, 22, (1997), 2365-2374
3. P. Antowick, "Senior Design Project Final Report", April 2004

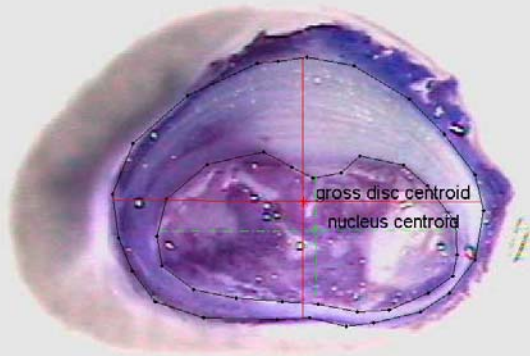
Appendix A

Human Spine diagram and other disc images





Left click to select the endpoints of the disc minor axis, from top to bottom. Then hit ENTER.
Now select the endpoints of the disc major axis, from left to right. Then hit ENTER.



Now select the endpoints of the NP minor axis, from top to bottom. Then hit ENTER.
Finally, select the endpoints of the NP major axis, from left to right. Then hit ENTER.

Appendix B

DiscArea MatLab code

```
function [areas, regions]=discarea(image, scale)
% Provides area measurements of a mouse disc
% given the image and scale input by user.
% The user selects the regions for the area calculations manually by
% approximating the disc as a polygon.

% This program was modified summer 2004 by David Jamison.

warning off
if nargin==2
% Reads and displays the image; allows calculation the unit-pixel
scale.
    %disp('To begin, you will measure the pixel-per-unit scale')
    %disp('Select the length of the unit (inch, mm, etc.) at least 5
times.')
```

```
    %disp('Press ENTER to continue.')
```

```
    %pause
    disc=imread(image);
    %imshow(disc);
    % [x,y]=ginput;
    %scale=sqrt((mean2(x)^2)+(mean2(y)^2))
    %close

% Reads and displays the image; "roipoly" command outlines the disc and
% NP. Also gives instructions for the user.
    disp('First, you will be outlining both the entire disc and the
NP.')
```

```
    disp('Press ENTER to begin')
```

```
    pause
    imshow(disc);
    axis on;
    text(1,1,'Use the crosshairs and click points to select the entire
disc.');
```

```
    text(1,40,'Right click the mouse when you have finished selecting
the region.');
```

```
    [BW,xi,yi]=roipoly;           % saves gross region as disc_gross in
binary
```

```
    text(1,70,'Now, do the same, this time outlining the nucleus.');
```

```
    [BW2,xi2,yi2]=roipoly;       % saves nucleus region as
'disc_nucleus' in binary
```

```
    X_avg_gross=mean2(xi);
    Y_avg_gross=mean2(yi);
    X_avg_np=mean2(xi2);
    Y_avg_np=mean2(yi2);
    close all;

% Displays more user instructions. Plots both the gross disc and NP
% centroids as well as the disc and NP outlines previously drawn by
the user on the image.
    disp('Now you are going to select the endpoints of the major')
    disp('and minor axes of the gross disc and NP.')
```

```
    disp('Press ENTER to continue')
```

```
    pause
```

```

figure
imshow(disc)
hold on
plot(xi,yi,'k.-')
plot(xi2,yi2,'k.-')
plot((X_avg_gross),Y_avg_gross,'r*')
plot((X_avg_np),Y_avg_np,'gd')
text(X_avg_gross + 10,Y_avg_gross - 7,'gross disc centroid')
text(X_avg_np +10,Y_avg_np - 7,'nucleus centroid')

% User selects the endpoint of the gross disc and NP major and minor
% axes.
text(1,10,'Left click to select the endpoints of the disc minor
axis, from top to bottom. Then hit ENTER.')
[x,y]=ginput;
text(1,40,'Now select the endpoints of the disc major axis, from
left to right. Then hit ENTER.')
[x2,y2]=ginput;
text(1,430,'Now select the endpoints of the NP minor axis, from top
to bottom. Then hit ENTER.')
[x3,y3]=ginput;
text(1,460,'Finally, select the endpoints of the NP major axis,
from left to right. Then hit ENTER.')
[x4,y4]=ginput;
% Superimposes the plots for gross disc and NP axes onto the image.
plot(x,y,'r-')
plot(x2,y2,'r-')
plot(x3,y3,'g-.')
plot(x4,y4,'g-.')

%Print the image
%print
pause(2)

%the following is Chandra's method of calculating major and minor axes
% More figures appear
figure;
[gross_radius,
gross_theta]=gross_axis(xi,yi,X_avg_gross,Y_avg_gross);
plot(gross_theta*180/pi,gross_radius,'*');
%ig1 = find((gross_theta*180/pi<91) & (gross_theta*180/pi > 0) )
[np_radius, np_theta]=gross_axis(xi2,yi2,X_avg_np,Y_avg_np);
pause(1)
figure
plot(np_theta*180/pi,np_radius,'+')
pause(1)
figure
subplot(1,2,1);
imshow(disc); % shows original image
subplot(1,2,2);
regions=visualize(BW, BW2); % shows binary composite image

% The following calculates and displays all measurements given the
inputs obtained above
total=find(BW); % finds nonzero elements (selected
region) of total disc

```

```

nucleus=find(BW2);          % selected region of nucleus
gross_area=size(total)/scale^2
nucleus_area=size(nucleus)/scale^2
annulus_area=gross_area-nucleus_area
percent_nucleus=nucleus_area/gross_area      %finds percent of disc
that is occupied by nucleus
Min_axis_gross=((y(2,1)-y(1,1))/scale)
Maj_axis_gross=((x2(2,1)-x2(1,1))/scale)
Min_axis_np=((y3(2,1)-y3(1,1))/scale)
Maj_axis_np=((x4(2,1)-x4(1,1))/scale)
X_Offset=abs(X_avg_gross-X_avg_np)/scale
Y_Offset=abs(Y_avg_gross-Y_avg_np)/scale
%Centroid seperation=(sqrt((X_avg_gross-X_avg_np)^2+(Y_avg_gross-
Y_avg_np)^2)/scale)
Aspect_ratio_gross=(Maj_axis_gross/Min_axis_gross)
Aspect_ratio_np=(Maj_axis_np/Min_axis_np)

% Writes the data calculated above to a .dat file.
B=[gross_area(1,1), nucleus_area(1,1), annulus_area(1,1),
percent_nucleus, Min_axis_gross, Maj_axis_gross, Min_axis_np,
Maj_axis_np, X_Offset, Y_Offset, Aspect_ratio_gross, Aspect_ratio_np];
fid = fopen('Disc_measurements.dat','a');
%fprintf(fid,'Total Disc Area, NP Area, AF Area, Percent NP,
Gross Min. Axis, Gross Major Axis, NP Min. Axis, NP Major Axis,
Centroid Offset X, Centroid Offset Y, Gross Aspect Ratio, NP Aspect
Ratio\n');
fprintf(fid,'\n');
fprintf(fid,'%6.5g, ',B);
fclose(fid);
end

function selected= visualize(gross, nucleus)
% allows user to visualize the selected regions
nucleus= +nucleus;
gross= +gross;
total=nucleus+gross;
S=size(gross);
for row=1:S(1)
    for col=1:S(2)
        if total(row,col)==2
            total(row,col)=0;
        end
    end
end
end
imshow(total);
selected=total;

function [r_g,theta_g]=gross_axis(xi,yi,xg,yg)
clk_no=size(xi);
for k=1:clk_no
    r_g(k)=sqrt((xi(k)-xg)^2 + (yi(k)-yg)^2);
    theta_g(k)=asin((xi(k)-xg)/r_g(k));%atan((xi(k)-xg)/(yi(k)-yg));
end
end

```