

Digital Image Analysis of Fundus Photographs on the WWW

*Ghassan Hamarneh
Image Analysis Group
Department of Signals and Systems
Chalmers University of Technology
Gothenburg, Sweden.*

*jessi@s2.chalmers.se
www.s2.chalmers.se*

Abstract

This report covers the project work done as part of a graduate Java programming course. The objective of this project was to develop a tool for studying the retinal fundus photographs of children using Java and other Internet technologies. The report includes overviews of the different steps of the project, a detailed description of the web-based system, lists of the computer code used, and suggested future work.

Acknowledgments

This project was done in collaboration with the Eye Clinic at Sahlgrenska University Hospital and the Department of Informatics at Göteborg University. The following were directly involved in the project:

Ann Hellström, MD, PhD
Eye Clinic - East Paediatric Ophthalmology
Sahlgrenska University Hospital
Email: ann.hellstrom@medfak.gu.se

Nina Lundberg, Ph.D student
Department of Informatics
Göteborg University
Email: nina.lundberg@bratwustle.hbg.lth.se

Mohammad Afshar-Beroz, MSEE student
Department of Informatics
Göteborg University
Email: s93beroz@student.adb.gu.se

Table of Contents

ABSTRACT	3
ACKNOWLEDGMENTS	5
TABLE OF CONTENTS	7
INTRODUCTION	9
IMAGE AND INFORMATION ACQUISITION	10
MORE DETAILS	10
HTML CODE FOR INFORMATION FORM	10
PERL CODE FOR HANDLING FORM DATA	12
HTML CODE FOR SUBMITTING AN IMAGE	13
THE MEASUREMENTS	14
CODE FOR .HTACCESS	15
THE .HTPASSWD	15
SUGGESTED STEPS	15
CALCULATED RESULTS	17
THE JAVA APPLET	18
MORE DETAILS	18
HTML CODE FOR APPLET PAGE	18
THE APPLET COMPONENTS	18
<i>Help and About</i>	18
<i>Image Loading</i>	18
<i>Settings</i>	19
<i>Options</i>	19
<i>Client Information</i>	19
<i>Output</i>	19
<i>Active Tips</i>	19
<i>Measurement</i>	19
<i>The Image</i>	20
JAVA CODE	20
HELP INFORMATION	43
<i>Digital Image Analysis of Fundus Photographs on the WWW</i>	43
<i>HELP</i>	43
REFERENCES & RELATED WWW LINKS (URLS)	47
REFERENCES	47
URLS	47
FUTURE	49
GLOSSARY	50

Introduction

This report covers the project work done as part of a graduate Java programming course. This project was done in collaboration with the Eye Clinic at Sahlgrenska University Hospital and the Department of Informatics at Göteborg University. Besides learning Java, the objective of this project was to develop a tool for studying the retinal fundus photographs of children. In clinical practice, we would like to collect images along with descriptive information from specialists around the world. As images become available, different measurements are performed and the results of such measurements are stored and made ready for analysis and presentation. The measurement process is required to be done by certain specialists working within different computer environments, therefore the platform independent Java programming language was chosen.

The following figure presents the projects form the perspective of the users: the client physicians and the measuring personnel. The client first visits the web site of the project, fills a form with needed information, submits needed information along with a fundus photograph and after a while checks for the results of the analysis. Once the information is received from the client, a trained personnel performs certain computer-assisted measurements. These measurements, along with other automated analysis and calculations, make up the results that are published on the web.

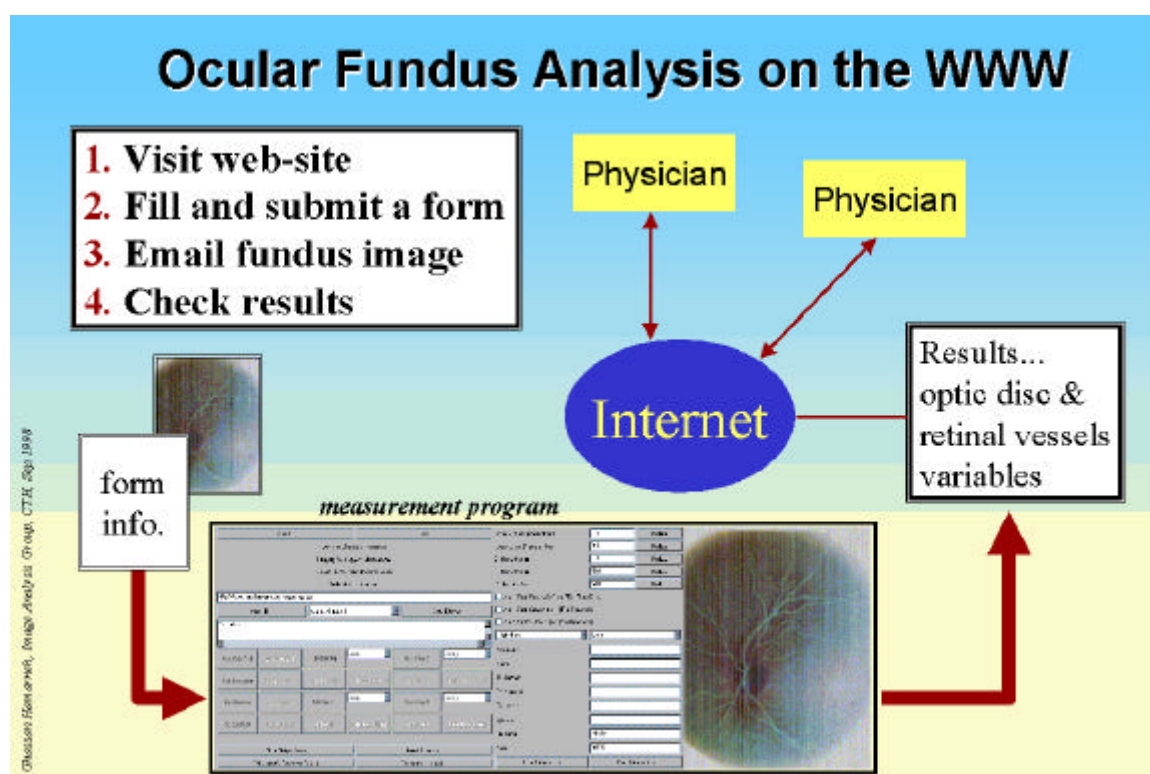


Figure 1. Schematic diagram of the project from the users' perspective. The client visits the website and submits necessary information, measurements are done, and results are presented for the client on the web.

Image and Information Acquisition

In order to acquire a larger number of images from a wide variety of locations around the world, a homepage for this project is put up on the Internet. Any interested person possessing a retinal fundus image fills and submits a form describing the image and then emails the image to a specified recipient. The image and the data are thus received and stored on a server.

More details

An HTML (Hyper Text Markup Language) file is written, containing some textual or graphical information about the project and instructions on how the interested person should proceed. The file also contains a form with GUI (Graphical User Interface) components such as labels, textboxes, radio-buttons, and buttons. The form is basically built up using the <FORM> tag with the different GUI components incorporated using mainly the <INPUT> tag with different attributes.

The screenshot shows a web browser window with the title "Fundus www - Form - Microsoft Internet Explorer". The page content is as follows:

Digital Image Analysis of Fundus Photographs on the WWW

Please complete and submit the following form.

Family Name

First Name

Sex Male
 Female

Age [when fundus photographed]

Gestational Weeks at Birth

Diagnose

Visual Acuity of Right Eye [0 - 1.5]

Left Eye [0 - 1.5]

Refraction of Right Eye

Left Eye

Name of Camera

Magnification Factor [30° - 70°]

E-mail

Additional Optional Information

Submit Information Reset Fields

Figure 2. The web-form used to collect information about the fundus photograph.

HTML Code for information form

The following represents the actual code used. The More important lines of code are highlighted in bold.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">  
<html>  
<head>
```

```

<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
<meta NAME="GENERATOR" CONTENT="Microsoft FrontPage 3.0">
<meta name="Microsoft Border" content="none, default">
<title>Fundus www - Form</title>
</head>

<body>
<h2 align="center">Digital Image Analysis of Fundus Photographs on the WWW</h2>
<h3 align="center"><font color="#000080">Please complete and submit the following form.</font></h3>
<form action="store.cgi" method="post">
<div align="center"><center><table border="0" cellpadding="3" cellspacing="1">
<tr>
<td align="right" valign="top"><font face="Times New Roman,Times"><strong>Family Name</strong></font></td>
<td align="left"><font face="Times New Roman,Times"><input type="text" name="surname"
size="10"></font></td>
</tr>
<tr>
<td align="right" valign="top"><strong><font face="Times New Roman,Times">First Name</font></strong></td>
<td align="left"><input type="text" name="givenname" size="10"></td>
</tr>
<tr>
<td align="right" valign="top"><strong><font face="Times New Roman,Times">Sex</font><br>
</strong></td>
<td align="left"><input TYPE="RADIO" NAME="sex" VALUE="male"><font
face="Times New Roman,Times">Male</font><br>
<input TYPE="RADIO" NAME="sex" VALUE="female"><font face="Times New Roman,Times">Female</font></td>
</tr>
<tr>
<td align="right" valign="top"><font face="Times New Roman,Times"><strong>Age</strong></font></td>
<td align="left"><input type="text" name="age" size="10"><font
face="Times New Roman,Times"><strong> [when fundus photographed]</strong></font></td>
</tr>
<tr>
<td align="right" valign="top"><strong><font face="Times New Roman,Times">Gestational
Weeks at Birth</font></strong></td>
<td align="left"><input type="text" name="week" size="10"></td>
</tr>
<tr>
<td align="right" valign="top"><strong><font face="Times New Roman,Times">Diagnose</font></strong></td>
<td align="left"><input type="text" name="diagnose" size="20"></td>
</tr>
<tr>
<td align="right" valign="top"><font face="Times New Roman,Times"><strong>Visual Acuity of
Right&nbsp;Eye</strong></font></td>
<td align="left"><font face="Times New Roman,Times"><input type="text" name="vright"
size="10"><strong> [0 - 1.5]</strong></font></td>
</tr>
<tr>
<td align="right" valign="top"><font face="Times New Roman,Times"><strong>Left&nbsp;Eye</strong></font></td>
<td align="left"><input type="text" name="vleft" size="10"><font
face="Times New Roman,Times"><strong> [0 - 1.5]</strong></font></td>
</tr>
<tr>
<td align="right" valign="top"><strong><font face="Times New Roman,Times">Refraction of
Right Eye</font></strong></td>
<td align="left"><font face="Times New Roman,Times"><input type="text" name="right"
size="10"></font></td>
</tr>
<tr>
<td align="right" valign="top"><strong><font face="Times New Roman,Times">Left Eye</font></strong></td>
<td align="left"><input type="text" name="left" size="10"></td>
</tr>
<tr>
<td align="right" valign="top"><strong><font face="Times New Roman,Times">Name of Camera</font></strong></td>
<td align="left"><input type="text" name="camera" size="20"></td>
</tr>
<tr>
<td align="right" valign="top"><strong><font face="Times New Roman,Times">Magnification
Factor</font></strong></td>
<td align="left"><input type="text" name="enlargment" size="10"><strong><font
face="Times New Roman,Times"> [30<sup>°</sup> - 70<sup>°</sup>]</font></strong></td>
</tr>
<tr>
<td align="right" valign="top"><strong><font face="Times New Roman,Times">E-mail</font></strong></td>
<td align="left"><input type="text" name="mail" size="40"></td>
</tr>
<tr>
<td align="right" valign="top"><strong><font face="Times New Roman,Times"><strong>Additional
Optional Information</strong></font></strong></td>
<td align="left"><textarea NAME="add" ROWS="10" COLS="50"></textarea></td>
</tr>
<tr>
<td align="right" valign="top"></td>
<td align="left"><strong><input type="submit" value="Submit Information"><input
TYPE="reset" VALUE="Reset Fields"></strong></td>
</tr>
</table>
</center></div>
</form>
<p>&nbsp;</p>
</body>
</html>

```

A main component of the form is the Submit Button, when this button is pressed, the information contained in the form elements are posted to the server and handled by a CGI (Common Gateway Interface) script written in PERL (Practical Extraction and Report Language). This PERL CGI script saves the information to a file on the server to be accessed later on by the measurement program.

PERL Code for Handling Form Data

The following represents the actual PERL code written to process the information submitted from the web form.

```
#!/usr/local/bin/perl
# CGI script for handling the information form
use CGI;

$query = new CGI;

if($query->param('spara') ne "YES") {
print "Content-type: text/html\n\n";
print "<HTML><HEAD><TITLE>formtest</TITLE></HEAD>";
print "<body>";
print "<strong>Please Confirm the Following Information Before Submission...</strong><p>";

print "<li><U>Family Name:</U> <B>" . $query->param('surname') . "</B></li>";
print "<li><U>First Name:</U> <B>" . $query->param('givenname') . "</B></li>";
print "<li><U>Sex:</U> <B>" . $query->param('sex') . "</B></li>";
print "<li><U>Age:</U> <B>" . $query->param('age') . "</B></li>";
print "<li><U>Gestational Weeks at Birth:</U> <B>" . $query->param('week') . "</B></li>";
print "<li><U>Diagnose:</U> <B>" . $query->param('diagnose') . "</B></li>";
print "<li><U>Visual Acuity Right:</U> <B>" . $query->param('vright') . "</B></li>";
print "<li><U>Visual Acuity Left:</U> <B>" . $query->param('vleft') . "</B></li>";
print "<li><U>Refraction Right:</U> <B>" . $query->param('right') . "</B></li>";
print "<li><U>Refraction Left:</U> <B>" . $query->param('left') . "</B></li>";
print "<li><U>Name of Camera:</U> <B>" . $query->param('camera') . "</B></li>";
print "<li><U>Magnification factor:</U> <B>" . $query->param('enlargment') . "</B></li>";
print "<li><U>E-mail:</U> <B>" . $query->param('mail') . "</B></li>";
print "<li><U>Additional information:</U> <B>" . $query->param('add') . "</B></li>";

print "<form action=store.cgi method=post>";
print "<INPUT TYPE=HIDDEN NAME=surname VALUE=" . $query->param('surname') . ">";
print "<INPUT TYPE=HIDDEN NAME=givenname VALUE=" . $query->param('givenname') . ">";
print "<INPUT TYPE=HIDDEN NAME=sex VALUE=" . $query->param('sex') . ">";
print "<INPUT TYPE=HIDDEN NAME=age VALUE=" . $query->param('age') . ">";
print "<INPUT TYPE=HIDDEN NAME=week VALUE=" . $query->param('week') . ">";
print "<INPUT TYPE=HIDDEN NAME=diagnose VALUE=" . $query->param('diagnose') . ">";
print "<INPUT TYPE=HIDDEN NAME=vright VALUE=" . $query->param('vright') . ">";
print "<INPUT TYPE=HIDDEN NAME=vleft VALUE=" . $query->param('vleft') . ">";
print "<INPUT TYPE=HIDDEN NAME=right VALUE=" . $query->param('right') . ">";
print "<INPUT TYPE=HIDDEN NAME=left VALUE=" . $query->param('left') . ">";
print "<INPUT TYPE=HIDDEN NAME=camera VALUE=" . $query->param('camera') . ">";
print "<INPUT TYPE=HIDDEN NAME=enlargment VALUE=" . $query->param('enlargment') . ">";
print "<INPUT TYPE=HIDDEN NAME=mail VALUE=" . $query->param('mail') . ">";
print "<INPUT TYPE=HIDDEN NAME=add VALUE=" . $query->param('add') . ">";

print "<input type=hidden name=spara value=YES>";
print "<strong>Information Confirmed?</strong><p>";print "<input type=submit value='Yes, submit!'><p>";
print "<strong>If not, please go to back to the form page and update your information <I>[use the back button of
your browser]</I>.</strong>";
print "</form></body></html>";

}
else
{
open (out, ">>info2.dat");
print out $query->param('surname'), "\n",
$query->param('givenname'), "\n",
$query->param('sex'), "\n",
$query->param('age'), "\n",
$query->param('week'), "\n",
$query->param('diagnose'), "\n",
$query->param('vright'), "\n",
$query->param('vleft'), "\n",
$query->param('right'), "\n",
$query->param('left'), "\n",
$query->param('camera'), "\n",
$query->param('enlargment'), "\n",
$query->param('mail'), "\n",
$query->param('add'), "\n\n";

close out;
open (out, ">>info.dat");
print out $query->param('surname'), "\n",
$query->param('givenname'), "\n",
$query->param('sex'), "\n",
$query->param('age'), "\n",
$query->param('week'), "\n",
$query->param('diagnose'), "\n",
$query->param('vright'), "\n",
$query->param('vleft'), "\n",
$query->param('right'), "\n",
$query->param('left'), "\n",
$query->param('camera'), "\n",
$query->param('enlargment'), "\n",
$query->param('mail'), "\n",
$query->param('add'), "\n\n";

close out;
print "Location: ok.html\n\n";
}
}
```

The user also presses a link that runs his email client program in order for him to send the retinal fundus image. This is done using the `` tag. Another more convenient way to submit the image file is to include a web form and a browse button. The user then browses through the

files on his local system and chooses the image file and submits it. This requires another CGI script to handle the form and was not done.

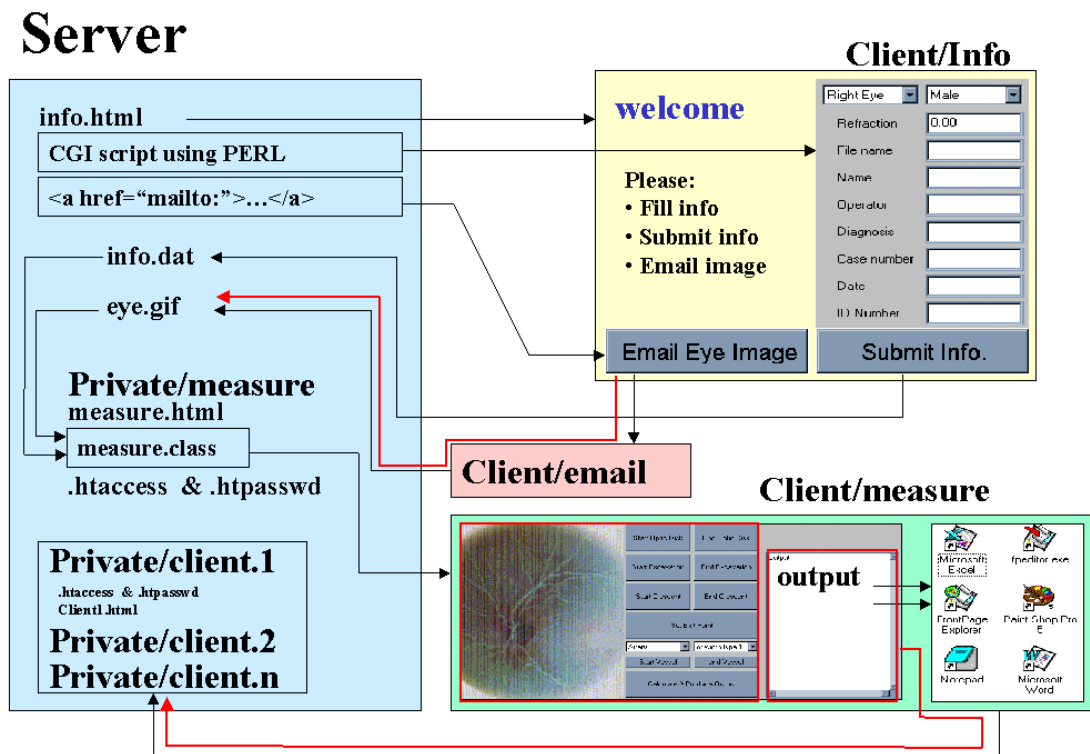
HTML Code for Submitting an Image

The important code is highlighted as before.

```

<html>
<head>
<title>Fundus wvw - send image</title>
</head>
<body>
<h2 align="center">Digital Image Analysis of Fundus Photographs on the WWW</h2>
<h3 align="center"><font color="#000080">Please send us the eye-images (GIF or JPEG
format) either </font></h3>
<h3 align="center"><font color="#000080">as an Email attachment to
<a href="mailto:jessi@s2.chalmers.se">jessi@s2.chalmers.se</a></font></h3>
<h3 align="center"><font color="#000080">or if you are using Netscape then you can upload
the image file directly to the server.</font></h3>
<form method="POST" action="_derived/nortbots.htm" onSubmit="location.href='_derived/nortbots.htm';return false;"
webbot-action="--WEBBOT-SELF--" WEBBOT-onSubmit="location.href='_derived/nortbots.htm';return false;">
<!--webbot bot="SaveResults" U-File="_private/form_results.txt" S-Format="TEXT/CSV" S-Label-Fields="TRUE" startspan
--><input TYPE="hidden" NAME="VTI-GROUP" VALUE="0"><!--webbot bot="SaveResults" endspan i-checksum="43374" --><div
align="center"><center><p><input type="file" value="Button" name="B1"><input type="submit" value="Upload"
name="B2"><input type="reset" value="Reset" name="B3"><strong><small> </small><font color="#FF0000">[not fully
implemented]</font></strong></p>
</center></div>
</form>
<h3 align="center"><br>
<font color="#800000">Thank you!</font></h3>
<p align="center"><a href="applet.html"><strong>Go to the Measurement Page</strong></a></p>
</body>
</html>

```



Ghassan Hamarneh, June '98

Figure 3. Schematic diagram showing the project information management.

The Measurements


In order to be able to do the measurements by persons working on different computer platforms and in different places, the measurement program is needed to be put on the Internet too. The program is required to be accessible only by authorized persons and it should have access to the images and their associated information. In order to make the HTML page containing the applet available only for authorized persons, it is put in a special directory along with two other files (.htaccess and .htpasswd) that describe the users and their encoded passwords, that are allowed to view this page.

**Applet done by Ghassan Hamarneh in Collaboration with the
Eye Clinic in the Eastern Hospital**
[\[See Overview \]](#)

<p>About Help</p> <p>Done by Ghassan Hamarneh Imaging 66, Image Analysis Group Department of Signals & Systems Gothenburg - Sweden</p>	<table style="width: 100%;"> <tr><td>Inner Circle Diameter (mm)</td><td>1.0</td><td>Update</td></tr> <tr><td>Outer Circle Diameter (mm)</td><td>3.0</td><td>Update</td></tr> <tr><td>X Pixels Per mm</td><td>50</td><td>Update</td></tr> <tr><td>Y Pixels Per mm</td><td>50</td><td>Update</td></tr> <tr><td>Calibration Factor</td><td>1.0</td><td>Update</td></tr> </table> <p><input type="checkbox"/> Auto (Start Excavation) on [End Optic Disk] <input type="checkbox"/> Auto (Start Crescent) on [End Excavation] <input type="checkbox"/> Auto (Set Exit Point) on [End Crescent]</p> <table style="width: 100%;"> <tr><td>Right Eye</td><td>Male</td><td>Age</td><td></td></tr> <tr><td>Last Name</td><td></td><td>First Name</td><td></td></tr> <tr><td>Gest. Weeks</td><td></td><td>Diagnosis</td><td></td></tr> <tr><td>Wa-R</td><td></td><td>Wa-L</td><td></td></tr> <tr><td>Ref-R</td><td></td><td>Ref-L</td><td></td></tr> <tr><td>Camera</td><td></td><td>Magnification</td><td></td></tr> <tr><td>Email</td><td></td><td>Additional</td><td></td></tr> <tr><td>File Name</td><td>Info.dat</td><td>Case</td><td></td></tr> <tr><td>ID Number</td><td></td><td>Date</td><td>990121</td></tr> <tr><td>Operator</td><td></td><td>Load</td><td>Clear</td></tr> </table>	Inner Circle Diameter (mm)	1.0	Update	Outer Circle Diameter (mm)	3.0	Update	X Pixels Per mm	50	Update	Y Pixels Per mm	50	Update	Calibration Factor	1.0	Update	Right Eye	Male	Age		Last Name		First Name		Gest. Weeks		Diagnosis		Wa-R		Wa-L		Ref-R		Ref-L		Camera		Magnification		Email		Additional		File Name	Info.dat	Case		ID Number		Date	990121	Operator		Load	Clear
Inner Circle Diameter (mm)	1.0	Update																																																						
Outer Circle Diameter (mm)	3.0	Update																																																						
X Pixels Per mm	50	Update																																																						
Y Pixels Per mm	50	Update																																																						
Calibration Factor	1.0	Update																																																						
Right Eye	Male	Age																																																						
Last Name		First Name																																																						
Gest. Weeks		Diagnosis																																																						
Wa-R		Wa-L																																																						
Ref-R		Ref-L																																																						
Camera		Magnification																																																						
Email		Additional																																																						
File Name	Info.dat	Case																																																						
ID Number		Date	990121																																																					
Operator		Load	Clear																																																					
<p>http://www.s2.chalmers.se/images/s2/.../img.gif Sample Image 1</p> <p><input type="button" value="View URL"/> <input type="button" value="View Local"/> <input type="button" value="View Choice"/></p> <p>Output >></p> <p style="text-align: center;"><input type="button" value="Clear Output Area"/> <input type="button" value="Insert Separator"/></p> <p style="text-align: center;"><input type="button" value="Calculate && Generate Output"/> <input type="button" value="Generate Information"/></p> <p style="text-align: center;">Loading image...</p>																																																								

Current Mode = 0 : NONE

Start Optic Disk	End Optic Disk
Start Excavation	End Excavation
Start Crescent	End Crescent
Set Exit Point	End Exit Point
Start Vessel	Artery
End Vessel	Direct Last Vessel
Start Bend	Artery
End Bend	End Optic Disk
Start Bend	Artery
End Bend	End Optic Disk
Start Bend	Artery
End Bend	End Optic Disk



92 1200
- 3
960926

Figure 4. The measurement program on the Web (HTML page containing the Java applet)

Code for .htaccess

Here is an example of what can be put in the .htaccess file.

```
AuthName          "Private files"
AuthType          Basic
AuthUserFile      /path_to_password_file/.htpasswd
<Limit GET POST>
    require valid-user
    satisfy all
</Limit>
```

The .htpasswd

A typical password file contains something like:

```
username1:encrypted_password1
username1:encrypted_password2
```

and can be produced by first creating the file (notice the flag -c) and then adding other users

```
htpasswd -c /path_to_password_file /.htpasswd username1
```

```
htpasswd /path_to_password_file/.htpasswd username2
```

repeating the last line as necessary and filling the passwords as you go.

The measurement program is written in Java (explained later on). The GUI of the program is shown in the next figure and is helpful while reading the next section.

Suggested Steps

The suggested step to be followed in order to perform a complete measurement is:

- **Loading the image:** this could be done in three ways:
 - ♣ Typing the URL of the image, and pressing the button 'View URL'.
 - ♣ Typing the name of the local image file and pressing the button 'View Local'.
 - ♣ Choosing a sample from the drop down list box, and pressing 'View Choice'The method suitable for our application is the second, since the image is supposed to be stored locally.
- **Settings:** completing the settings, resolution and calibration parameters (ex. Inner Circle Diameter, Calibration factor)
- **Options:** setting any of the three Auto-Start options by clicking the corresponding check boxes.
- **Client Info:** to obtain the client info, first the name of the file is typed (if different from default), followed by pressing the 'Load' button.
- **Measurements:** all measurements are done by first pressing the 'Start' button of a certain type, then using the mouse to click on the image and mark the desired contour or curve, and then pressing the 'End' button for the same type. Some measurements like those for vessels, require the user to select a type out of a drop down list box, this can be done, any time between pressing the 'Start' and 'End'
- **Other Information:**
 - ♣ The user can always press the 'Help' button to display a page of useful comments and hints.
 - ♣ The user is encouraged to read the status and mode tips, just above the image.
 - ♣ The user can re-input selected regions by simply re-starting, clicking and ending.
 - ♣ If there are many objects of a certain type (ex. vessels) then the user can delete those vessels previously entered from last to first.
 - ♣ At any time the user can press 'Generate Information' and read the information displayed that describes his progress in the measuring process.
- **Calculations:** with all the regions specified, the user can press the 'Calculate & Generate Output' button and obtain all the measurements needed. Some calculations can't be done if the data is not completely available, and this will be indicated in the output.

Following is an example of what is displayed when pressing the 'Generate Information' button:

```

Output >>
-----
#Pts for CNT1 (Optic Disk) = 20
{(371,249),(378,245),(401,245),(407,247),(423,257),(433,271),(443,294),(442,327),(426,348),(405,357),(372,359),(345,352),(334,333),(329,312),(331,289),(339,273),(345,263),(351,257),(359,252),(367,249)}
#Pts for CNT2 (Excavation) = 17
{(362,293),(367,293),(383,298),(386,303),(388,308),(388,315),(387,319),(383,320),(378,322),(360,323),(348,318),(344,312),(344,307),(344,301),(347,297),(351,295),(360,293)}
#Pts for CNT3 (Crescent) = 0 {}
XPNT (Exit Point): (395,308)
Pixels Per millimeter [X] =50
Pixels Per millimeter [Y] =50
Inner Circle Diameter = 6 mm = 300 xPixels = 300 yPixels
Outer Circle Diameter = 12 mm = 600 xPixels = 600 yPixels
#VSSL (Vessels) = 8
  Vessel 0: #Pts = 16, Artery. Total Length = 6.66661 mm. Line Length = 6.24157 mm
  {(396,306)(389,297)(387,292)(387,286)(387,278)(392,258)(386,208)(351,171)(309,130)(279,88)(271,72)(266,66)(255,60)(247,53)(228,43)(204,35)}
  Vessel 1: #Pts = 19, Artery. Total Length = 5.82537 mm. Line Length = 5.7164 mm
  {(387,288)(381,286)(375,283)(350,272)(343,262)(329,255)(308,240)(285,231)(245,211)(213,202)(195,200)(189,199)(179,197)(161,194)(137,189)(127,186)(120,186)(104,186)(94,187)}
  Vessel 2: #Pts = 23, Artery. Total Length = 6.15774 mm. Line Length = 5.87639 mm
  {(396,306)(400,317)(400,328)(393,339)(387,356)(383,364)(380,370)(376,380)(370,389)(363,401)(356,416)(350,425)(337,438)(316,458)(292,477)(267,498)(257,505)(249,514)(242,520)(237,524)(232,529)(217,539)(194,558)}
  Vessel 3: #Pts = 22, Artery. Total Length = 6.40119 mm. Line Length = 5.86849 mm
  {(401,329)(405,341)(412,350)(417,362)(420,372)(421,387)(421,396)(421,407)(421,425)(422,449)(428,461)(431,466)(438,475)(454,485)(463,492)(478,505)(501,514)(513,524)(529,536)(542,548)(561,558)(574,566)}
  Vessel 4: #Pts = 18, Vein. Total Length = 6.41017 mm. Line Length = 5.88422 mm
  {(397,307)(391,297)(389,292)(389,290)(388,281)(389,274)(390,263)(378,224)(370,216)(364,210)(357,203)(339,191)(317,187)(278,169)(250,149)(226,119)(185,103)(151,96)}
  Vessel 5: #Pts = 15, Vein. Total Length = 6.04594 mm. Line Length = 5.92284 mm
  {(390,291)(402,286)(410,282)(420,277)(427,271)(434,265)(454,263)(471,261)(483,257)(506,256)(523,245)(545,229)(620,193)(658,165)(668,148)}
  Vessel 6: #Pts = 18, Vein. Total Length = 6.51102 mm. Line Length = 6.30165 mm
  {(395,307)(393,317)(389,325)(385,332)(373,336)(358,342)(341,352)(336,362)(333,378)(329,386)(316,403)(285,430)(247,479)(220,508)(213,517)(204,527)(193,535)(184,541)}
  Vessel 7: #Pts = 15, Vein. Total Length = 6.20668 mm. Line Length = 6.06422 mm
  {(388,330)(392,335)(401,343)(414,350)(434,356)(450,363)(466,370)(480,381)(509,408)(528,442)(550,455)(591,491)(610,506)(629,514)(647,522)}
#GEN1 (Generation1 Vessels) = 10
  Generation1 Vessel 0: #Pts = 7, Vein. Total Length = 4.17433 mm. Line Length = 3.95464 mm
  {(372,215)(393,182)(407,128)(411,90)(409,75)(406,63)(389,18)}
  Generation1 Vessel 1: #Pts = 9, Vein. Total Length = 1.93225 mm. Line Length = 1.85235 mm
  {(259,155)(258,163)(240,176)(236,182)(230,187)(223,193)(219,197)(210,203)(186,212)}
  Generation1 Vessel 2: #Pts = 16, Vein. Total Length = 6.42188 mm. Line Length = 6.10318 mm
  {(407,284)(419,292)(425,305)(430,313)(439,320)(446,325)(455,329)(467,332)(485,334)(524,336)(539,335)(575,331)(609,330)(635,333)(655,338)(706,345)}
  Generation1 Vessel 3: #Pts = 10, Vein. Total Length = 2.9812 mm. Line Length = 2.84703 mm
  {(322,390)(313,388)(292,382)(290,378)(285,373)(280,366)(263,350)(235,336)(198,333)(192,332)}
  Generation1 Vessel 4: #Pts = 7, Vein. Total Length = 2.31599 mm. Line Length = 2.26451 mm
  {(288,427)(275,424)(257,424)(241,420)(214,415)(190,399)(180,393)}
  Generation1 Vessel 5: #Pts = 11, Artery. Total Length = 3.47064 mm. Line Length = 3.44981 mm
  {(323,452)(298,450)(285,450)(272,451)(263,454)(240,455)(231,458)(207,462)(178,464)(151,465)(116,465)}
  Generation1 Vessel 6: #Pts = 11, Artery. Total Length = 5.38064 mm. Line Length = 5.0012 mm
  {(391,256)(409,256)(418,256)(433,250)(451,240)(456,227)(460,210)(474,168)(515,112)(550,63)(574,48)}
  Generation1 Vessel 7: #Pts = 9, Artery. Total Length = 5.14957 mm. Line Length = 5.07894 mm
  {(401,318)(422,315)(430,305)(446,295)(464,286)(486,277)(556,255)(641,235)(739,216)}
  Generation1 Vessel 8: #Pts = 10, Artery. Total Length = 2.04188 mm. Line Length = 2.0026 mm
  {(448,481)(446,487)(446,495)(446,502)(445,510)(445,521)(449,536)(453,550)(463,580)(502,612)}
  Generation1 Vessel 9: #Pts = 15, Vein. Total Length = 4.59442 mm. Line Length = 4.50289 mm
  {(410,348)(415,360)(414,370)(409,379)(404,390)(400,405)(397,415)(391,425)(388,436)(378,466)(375,488)(372,508)(371,518)(369,549)(367,569)}
#GEN2 (Generation2 Vessels) = 1
  Generation2 Vessel 0: #Pts = 14, Artery. Total Length = 4.71294 mm. Line Length = 4.66154 mm
  {(430,309)(448,313)(452,320)(460,327)(470,335)(480,341)(504,356)(526,369)(560,387)(576,399)(593,408)(615,419)(631,427)(703,454)}
#GEN3 (Generation3 Vessels) = 0

```


Calculated results

- **Number of branching points:** Total, for Arteries, Veins, Unspecified, Vessels, First, Second and Third Generation Vessels.
- **Indices of Tortuosity:** Total, for Arteries, Veins, Unspecified, Vessels, First, Second and Third Generation Vessels.
- **Areas:**
 - Area of Optic Disk
 - Area of Excavation
 - Area of Crescent
 - Area of Rim

Following is an example of what is displayed when pressing the 'Calculate & Generate Output' button

```
Output >>
-----
Total number of Branching Points = 4
Number of Branching Points for Arteries = 3
Gen1 = 1, Gen2 = 1, Gen3 = 1
Number of Branching Points for Veins = 1
Gen1 = 1, Gen2 = 0, Gen3 = 0
Number of Branching Points for Unspecified = 0
Gen1 = 0, Gen2 = 0, Gen3 = 0
Index of Tortuosity for Everything = 1.091980233050493
Index of Tortuosity for All Arteries = 1.0696463286592335, All Veins = 1.0222439495775393, All Unspecified =
1.2652848879635152
Index of Tortuosity for All Vessels = 1.1325866967641787, All Gen1s = 1.0029471131735739, All Gen2s =
1.0105821792010004, All Gen3s = 1.042429427783474
Index of Tortuosity for Vessels of Arteries = 1.1126581564081444, Gen1s of Arteries = 1.0029256476919584, Gen2s of
Arteries = 1.0105821792010004, Gen3s of Arteries = 1.042429427783474
Index of Tortuosity for Vessels of Veins = 1.0320969820378307, Gen1s of Veins = 1.0029745436995274, Gen2s of Veins
= Not Available, Gen3s of Veins = Not Available
Index of Tortuosity for Vessels of Unspecified = 1.2652848879635152, Gen1s of Unspecified = Not Available, Gen2s of
Unspecified = Not Available, Gen3s of Unspecified = Not Available
Area of Optic Disk = 1.3152
Area of Excavation = 2.1530000000000014
Area of Crescent = 3.1968000000000005
Area of Rim = -0.8378000000000014
```

The Java Applet

More details

The measurement program is written in Java. Java is an Object Oriented Programming (OOP) language famous for the "write once run anywhere" concept (actually it is run anywhere with a Java Virtual Machine or JVM), which makes it suitable for our application. The program is included as an Applet in an HTML page using the <APPLET> tag (In HTML 4.0, this tag is deprecated (gradually phased out) and replaced by the <OBJECT> tag).

HTML Code for Applet page

Important code is highlighted.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta name="Template" content="C:\Program Files\Microsoft Office\Office\html.dot">
<meta name="GENERATOR" content="Microsoft FrontPage 3.0">
<title>measure</title>
</head>

<body bgcolor="#FFFFFF" link="#0000FF" vlink="#800080">
<div align="center"><center>

<table border="1" cellpadding="7" cellspacing="3" width="638">
<tr>
<td align="center"><a href="#down"></a></td>
<td valign="top" rowspan="2"><p align="center"><a name="up"></a><b><font size="5">Applet
done by Ghassan Hamarneh in Collaboration with the Eye Clinic in the Eastern Hospital<br>
</font><font size="4">[ </font><a href="html_report/report.html"><font size="4"><em>See
Overview</em></font></a><font size="4"> ]</font></b></p>
<p align="center">&nbsp;<b>
<applet code="measure.class" align="baseline" width="1000" height="1200">Sorry! You need a Java-Enabled
Browser!</applet>
</td>
<td align="center"><a href="#down"></a></td>
</tr>
<tr>
<td align="center" valign="top"><a href="#up"></a></td>
<td align="center" valign="top"><a name="down"></a><a href="#up"></a></td>
</tr>
</table>
</center></div>

<p>&nbsp;</p>
</body>
</html>
```

In OOP jargon, the program is mainly an instance of a class that extends the applet class but with the additional methods and properties to augment the additional functionality specific to our applications. What the applet does -simply put- is to load an image, load the additional information relating to the image, then display the image for the operator to specify different regions, and finally perform different calculations and present the results to the user.

The Applet Components

Help and About

- ⊖ **About** press this button to read the about information
- ⊖ **Help** press this button to view the help page

Image Loading

- ⊖ **URL** enter the URL of the image in this text field
- ⊖ **View URL** press this button to view the image of the URL
- ⊖ **Local File Name** enter the name of the local image file
- ⊖ **View Local File** press this button to view the local image
- ⊖ **Sample Images** a dropdown list box of sample images
- ⊖ **View Sample Image** press this button to view the chosen sample image

Settings

- ⊖ **Inner Circle Diameter** enter the diameter in this text field
- ⊖ **Update Inner Circle Diameter** press to update (or press return in the text field)
- ⊖ **Outer Circle Diameter** enter the diameter in this text field
- ⊖ **Update Outer Circle Diameter** press to update (or press return in the text field)
- ⊖ **X Pixels Per mm** enter the value in this text field
- ⊖ **Update X Pixels Per mm** press to update (or press return in the text field)
- ⊖ **Y Pixels Per mm** enter the value in this text field
- ⊖ **Update Y Pixels Per mm** press to update (or press return in the text field)

Options

- ⊖ **Auto [Start Excavation] on [End Optic Disk]** set check box for auto start
- ⊖ **Auto [Start Crescent] on [End Excavation]** set check box for auto start
- ⊖ **Auto [Set Exit Point] on [End Crescent]** set check box for auto start

Client Information

- ⊖ **Surname**
- ⊖ **Given Name**
- ⊖ **Sex Male/Female**
- ⊖ **Right/Left Eye**
- ⊖ **Age (when fundus photographed)**
- ⊖ **Gestational Weeks at Birth**
- ⊖ **Diagnose**
- ⊖ **Right Eye Visual Acuity [0 - 1.5]**
- ⊖ **Left Eye Visual Acuity [0 - 1.5]**
- ⊖ **Right Eye Refraction**
- ⊖ **Left Eye Refraction**
- ⊖ **Name of Camera**
- ⊖ **Magnification factor [30° - 70 °]**
- ⊖ **Email**
- ⊖ **Additional Information (optional)**
- ⊖ **Load Information** loads the latest information submitted
- ⊖ **Clear Information** clears the above fields

Output

- ⊖ **Output Area** calculation results and information will be visible here
- ⊖ **Clear Output Area** press this button to clear the area
- ⊖ **Insert Separator** press this button to insert a separator
- ⊖ **Calculate & Generate Output** press this button to calculate & generate output
- ⊖ **Generate Information** press this button to generate information about the measurement progress

Active Tips

- ⊖ **Status and Help Tips** watch this label for status information and useful tips
- ⊖ **Current Mode** the current mode of operation is displayed here

Measurement

- ⊖ **Start Optic Disk** press this button to start specifying region
- ⊖ **End Optic Disk** press this button to end specifying region
- ⊖ **Start Excavation** press this button to start specifying region
- ⊖ **End Excavation** press this button to end specifying region
- ⊖ **Start Crescent** press this button to start specifying region
- ⊖ **End Crescent** press this button to end specifying region
- ⊖ **Set Exit Point** press this button to start specifying point
- ⊖ **End Exit Point** press this button to end specifying point
- ⊖ **Start Vessel** press this button to start specifying vessel
- ⊖ **End Vessel** press this button to end specifying vessel
- ⊖ **Vessel Type (Artery/Vein/Unspecified)** choose type of vessel

- 0 **Start Gen1** press this button to start specifying vessel of generation 1
- 0 **End Gen1** press this button to end specifying vessel of generation 1
- 0 **Gen1 Type (Artery/Vein/Unspecified)** choose type of vessel of generation 1
- 0 **Start Gen2** press this button to start specifying vessel of generation 2
- 0 **End Gen2** press this button to end specifying vessel of generation 2
- 0 **Gen2 Type (Artery/Vein/Unspecified)** choose type of vessel of generation 2
- 0 **Start Gen3** press this button to start specifying vessel of generation 3
- 0 **End Gen3** press this button to end specifying vessel of generation 3
- 0 **Gen3 Type (Artery/Vein/Unspecified)** choose type of vessel of generation 3

The Image

Image the retinal fundus image is displayed here along with the regions, vessels, points, and circles.

Java Code

The complete Java code is presented here (if you want to skip this code you can go directly to page 43).

```

////////////////////////////////////
// Copyright 1998 (c)
// Ghassan Hamarneh
// Chalmers, Gothenburg, Sweden
// jessi@s2.chalmers.se
////////////////////////////////////

//          TO DO
//=====

// If you want to test exceptions
/*
    try
    {
        code to test
    }//end try & then the catch
    catch(Exception ex)
    {
        display.setText("!!!EXCEPTION"+ex.toString());
    }//end catch
*/

//C:\Program Files\Plus!\Microsoft Internet\IEXPLORE.EXE
//C:\Program Files\Netscape\Communicator\Program\netscape.exe

/*
*****

PROJECT FOR THE COURSE ON...
-----
    JAVA PROGRAMMING with focus on
    IMAGING, IMAGE PROCESSING & ANALYSIS

DONE BY...
-----
    GHASSAN HAMARNEH
    IMAGING & IMAGE ANALYSIS GROUP
    DEPT. OF SIGNALS AND SYSTEMS
    CHALMERS UNIVERSITY OF TECHNOLOGY
    GOTHENBURG - SWEDEN

IN COLLABORATION WITH...
-----
    THE EYE CLINIC IN THE EASTERN HOSPITAL
    GOTHENBURG - SWEDEN

PERIOD
-----
    SPRING/SUMMER 1998

*****
*/

////////////////////////////////////
// IMPORTS
////////////////////////////////////
import java.awt.*;
import java.awt.image.*;
import java.applet.*;
import java.net.*;
import java.util.*;
import java.io.*;

////////////////////////////////////
// APPLET STARTS
////////////////////////////////////
public class measure extends Applet
{
    //see PRCNT_ERR_MRG01 in Vessel Class

```

```

boolean dspImgMsg=false;
final int NONE=0;
final int CNT1=1;
final int CNT2=2;
final int CNT3=3;
final int XPNT=4;
final int VSSL=5;
final int GEN1=6;
final int GEN2=7;
final int GEN3=8;
final String[] modeNames={"NONE",
    "CNT1 (Optic Disk)",
    "CNT2 (Excavation)",
    "CNT3 (Crescent)",
    "XPNT (Exit Point)",
    "VSSL (Vessels)",
    "GEN1 (Generation1 Vessels)",
    "GEN2 (Generation2 Vessels)",
    "GEN3 (Generation3 Vessels)"};

final Color CLR_CNT1=Color.yellow;
final Color CLR_CNT2=Color.orange;
final Color CLR_CNT3=Color.green;
final Color CLR_XPNT=Color.lightGray;
final Color CLR_CRCLin=Color.gray;
final Color CLR_CRCLut=Color.darkGray;
final Color CLR_VSSL_V=Color.blue;
final Color CLR_VSSL_A=Color.red;
final Color CLR_VSSL_U=Color.magenta;
final Color CLR_GEN1_V=Color.blue;
final Color CLR_GEN1_A=Color.red;
final Color CLR_GEN1_U=Color.magenta;
final Color CLR_GEN2_V=Color.blue;
final Color CLR_GEN2_A=Color.red;
final Color CLR_GEN2_U=Color.magenta;
final Color CLR_GEN3_V=Color.blue;
final Color CLR_GEN3_A=Color.red;
final Color CLR_GEN3_U=Color.magenta;

////////////////////////////////////
//NEED TO SAVE AND LOAD //
////////////////////////////////////

Contour contour1=new Contour();//Optic Disk //NEED TO SAVE
Contour contour2=new Contour();//Excavation //NEED TO SAVE
Contour contour3=new Contour();//Crescent //NEED TO SAVE
Vector vessels=new Vector(); //NEED TO SAVE
Vector gen1s=new Vector(); //NEED TO SAVE
Vector gen2s=new Vector(); //NEED TO SAVE
Vector gen3s=new Vector(); //NEED TO SAVE

Point xPnt=new Point(-1,-1); //NEED TO SAVE

int pxl_per_mm_x=50; //NEED TO SAVE
int pxl_per_mm_y=50; //NEED TO SAVE

double mmDin=1.0; //NEED TO SAVE
double mmDut=3.0; //NEED TO SAVE

////////////////////////////////////

int pxlDin_x=(int)(mmDin*pxl_per_mm_x);
int pxlDin_y=(int)(mmDin*pxl_per_mm_y);
int pxlDut_x=(int)(mmDut*pxl_per_mm_x);
int pxlDut_y=(int)(mmDut*pxl_per_mm_y);

int mode=NONE;

Image Img=null;
int ImgX,ImgY;

String aboutStr=new String("Applet done by Ghassan Hamarneh in Collaboration with the Eye Clinic in the
Eastern Hospital");

////////////////////////////////////
Panel imageLoadPanel=new Panel();
////////////////////////////////////
TextField imgTF=new TextField("http://www.s2.chalmers.se/images/s2.gif");
TextField locimgTF=new TextField("img.gif",10);
Choice imgChoice=new Choice();

Button viewURL=new Button("View URL");
Button viewLoc=new Button("View Local");
Button viewChoice=new Button("View Choice");
////////////////////////////////////

////////////////////////////////////
Panel inputPanel=new Panel();
////////////////////////////////////
Button startContour1=new Button("Start Optic Disk");
Button endContour1=new Button("End Optic Disk");
Button startContour2=new Button("Start Excavation");
Button endContour2=new Button("End Excavation");
Button startContour3=new Button("Start Crescent");
Button endContour3=new Button("End Crescent");

Button setPoint=new Button("Set Exit Point");
Button endPoint=new Button("End Exit Point");

```

```

//Choice widthChoice = new Choice();
Choice vesselChoice = new Choice();
Button startVessel=new Button("Start Vessel");
Button endVessel=new Button("end Vessel");

Choice gen1Choice = new Choice();
Button startGen1=new Button("Start Gen1");
Button endGen1=new Button("End Gen1");

Choice gen2Choice = new Choice();
Button startGen2=new Button("Start Gen2");
Button endGen2=new Button("End Gen2");

Choice gen3Choice = new Choice();
Button startGen3=new Button("Start Gen3");
Button endGen3=new Button("End Gen3");

Button delLastVessel = new Button("Delete Last Vessel");
Button delLastGen1 = new Button("Delete Last Gen1");
Button delLastGen2 = new Button("Delete Last Gen2");
Button delLastGen3 = new Button("Delete Last Gen3");
////////////////////////////////////

////////////////////////////////////
Panel optionsPanel=new Panel();
////////////////////////////////////
Label diamLABELin=new Label("Inner Circle Diameter (mm)");
TextField diamTFin=new TextField(""+mmDin,5);
Button updateInner=new Button("Update");

Label diamLABELut=new Label("Outer Circle Diameter (mm)");
TextField diamTFut=new TextField(""+mmDut,5);
Button updateOuter=new Button("Update");

Label calibLABEL=new Label("Calibration Factor");
TextField calibTF=new TextField("1.0",5);
Button updateCalib=new Button("Update");

Label pxl_per_mm_xLABEL=new Label("X Pixels Per mm");
TextField pxl_per_mm_xTF=new TextField(""+pxl_per_mm_x,5);
Button updatePxl_per_mm_x=new Button("Update");

Label pxl_per_mm_yLABEL=new Label("Y Pixels Per mm");
TextField pxl_per_mm_yTF=new TextField(""+pxl_per_mm_y,5);
Button updatePxl_per_mm_y=new Button("Update");

Checkbox auto12=new Checkbox("Auto [Start Excavation] on [End Optic Disk]");
Checkbox auto23=new Checkbox("Auto [Start Crescent] on [End Excavation]");
Checkbox autoXpnt=new Checkbox("Auto [Set Exit Point] on [End Crescent]");
//Checkbox autoVessel=new Checkbox("Auto [Start Vessel] on [End Exit Point]");

Panel p21=new Panel();
Panel p211=new Panel();
Panel p212=new Panel();
Panel p213=new Panel();
Panel p214=new Panel();
Panel p215=new Panel();

Panel p221=new Panel();
Panel p222=new Panel();
Panel p223=new Panel();
Panel p224=new Panel();
Panel p225=new Panel();

////////////////////////////////////

////////////////////////////////////
Panel infoPanel=new Panel();
////////////////////////////////////
Label lnameLABEL=new Label("Last Name");
TextField lnameTF=new TextField("",5);

Label fnameLABEL=new Label("First Name");
TextField fnameTF=new TextField("",5);

Choice lrChoice = new Choice();
Choice sexChoice=new Choice();

Label ageLABEL=new Label("Age");
TextField ageTF=new TextField("",5);

Label weeksLABEL=new Label("Gest. Weeks");
TextField weeksTF=new TextField("",5);

Label diagLABEL=new Label("Diagnosis");
TextField diagTF=new TextField("",10);

Label VAR_LABEL=new Label("VA-R");
TextField VAR_TF=new TextField("",5);

Label VAL_LABEL=new Label("VA-L");
TextField VAL_TF=new TextField("",5);

Label refrLABEL=new Label("Ref-R");
TextField refrTF=new TextField("",5);

Label reflLABEL=new Label("Ref-L");
TextField reflTF=new TextField("",5);

```

```

Label camLABEL=new Label("Camera");
TextField camTF=new TextField("",5);

Label magLABEL=new Label("Magnification");
TextField magTF=new TextField("",5);

Label emailLABEL=new Label("Email");
TextField emailTF=new TextField("",5);

Label addLABEL=new Label("Additional");
TextField addTF=new TextField("",5);

Label fileLABEL=new Label("File Name");
TextField fileTF=new TextField("info.dat",5);

Label caseLABEL=new Label("Case");
TextField caseTF=new TextField("",10);

Label idLABEL=new Label("ID Number");
TextField idTF=new TextField("",10);

Label dateLABEL=new Label("Date");
TextField dateTF=new TextField("",6);

Label opLABEL=new Label("Operator");
TextField opTF=new TextField("",10);

Button loadInfo=new Button("Load");
Button clearInfo=new Button("Clear");
////////////////////////////////////

////////////////////////////////////
Panel outputPanel=new Panel();
////////////////////////////////////
TextArea opAREA = new TextArea("Output >>",20,30);
Button clearOp=new Button("Clear Output Area");
Button insertSep=new Button("Insert Seperator");
Button startCalc = new Button("Calculate && Generate Output");
Button opInfo = new Button("Generate Information");
Panel p31=new Panel();
Panel p32=new Panel();
Panel p33=new Panel();
////////////////////////////////////

////////////////////////////////////
Panel imagePanel=new Panel();
////////////////////////////////////
Panel imgp=new Panel();
////////////////////////////////////

////////////////////////////////////
Panel statusPanel=new Panel();
////////////////////////////////////
Label display = new Label("aboutStr",Label.CENTER);
Label dspMode = new Label("Current Mode = "+mode+" : "+modeNames[mode],Label.LEFT);
////////////////////////////////////

////////////////////////////////////
Panel extraPanel=new Panel();
////////////////////////////////////
Button about=new Button("About");
Button help=new Button("Help");
Label xtraLB1=new Label("Done by Ghassan Hamarneh",Label.CENTER);
Label xtraLB2=new Label("Imaging && Image Analysis Group",Label.CENTER);
Label xtraLB3=new Label("Department of Signals && Systems",Label.CENTER);
Label xtraLB4=new Label("Gothenburg - Sweden",Label.CENTER);
Panel p9=new Panel();

////////////////////////////////////
// INIT
////////////////////////////////////
public void init()
{
    //////////////////////////////////////
    //imageLoadPanel
    imgChoice.addItem("Sample Image 1");
    imgChoice.addItem("Sample Image 2");
    imgChoice.addItem("Sample Image 3");
    imgChoice.addItem("Sample Image 4");

    imageLoadPanel.setLayout(new GridLayout(2,3,4,4));
    imageLoadPanel.add(imgTF);          imageLoadPanel.add(locimgTF);
    imageLoadPanel.add(imgChoice);
    imageLoadPanel.add(viewURL);  imageLoadPanel.add(viewLoc);  imageLoadPanel.add(viewChoice);

    //////////////////////////////////////
    //inputPanel
    vesselChoice.addItem("Artery");
    vesselChoice.addItem("Vein");
    vesselChoice.addItem("Unspecified");

    gen1Choice.addItem("Artery");
    gen1Choice.addItem("Vein");
    gen1Choice.addItem("Unspecified");

    gen2Choice.addItem("Artery");
    gen2Choice.addItem("Vein");
}

```

```

gen2Choice.addItem("Unspecified");

gen3Choice.addItem("Artery");
gen3Choice.addItem("Vein");
gen3Choice.addItem("Unspecified");

inputPanel.setLayout(new GridLayout(12,2,4,4));
inputPanel.add(startContour1);          inputPanel.add(endContour1);
inputPanel.add(startContour2);          inputPanel.add(endContour2);
inputPanel.add(startContour3);          inputPanel.add(endContour3);
inputPanel.add(setPoint);               inputPanel.add(endPoint);
inputPanel.add(startVessel);             inputPanel.add(vesselChoice);
inputPanel.add(endVessel);               inputPanel.add(delLastVessel);
inputPanel.add(startGen1);               inputPanel.add(gen1Choice);
inputPanel.add(endGen1);                 inputPanel.add(delLastGen1);
inputPanel.add(startGen2);               inputPanel.add(gen2Choice);
inputPanel.add(endGen2);                 inputPanel.add(delLastGen2);
inputPanel.add(startGen3);               inputPanel.add(gen3Choice);
inputPanel.add(endGen3);                 inputPanel.add(delLastGen3);

endContour1.disable();
endContour2.disable();
endContour3.disable();
endPoint.disable();
endVessel.disable();
endGen1.disable();
endGen2.disable();
endGen3.disable();
delLastVessel.disable();
delLastGen1.disable();
delLastGen2.disable();
delLastGen3.disable();

////////////////////////////////////
//optionsPanel
p211.setLayout(new GridLayout(1,2,2,2));
p211.add(diamTFin); p211.add(updateInner);

p212.setLayout(new GridLayout(1,2,2,2));
p212.add(diamTFut); p212.add(updateOuter);

p213.setLayout(new GridLayout(1,2,2,2));
p213.add(px1_per_mm_xTF); p213.add(updatePx1_per_mm_x);

p214.setLayout(new GridLayout(1,2,2,2));
p214.add(px1_per_mm_yTF); p214.add(updatePx1_per_mm_y);

p215.setLayout(new GridLayout(1,2,2,2));
p215.add(calibTF); p215.add(updateCalib);

Panel p221=new Panel();p221.setLayout(new GridLayout(1,2,4,4));
p221.add(diamLABELin); p221.add(p211);

Panel p222=new Panel();p222.setLayout(new GridLayout(1,2,4,4));
p222.add(diamLABELut); p222.add(p212);

Panel p223=new Panel();p223.setLayout(new GridLayout(1,2,4,4));
p223.add(px1_per_mm_xLABEL); p223.add(p213);

Panel p224=new Panel();p224.setLayout(new GridLayout(1,2,4,4));
p224.add(px1_per_mm_yLABEL); p224.add(p214);

Panel p225=new Panel();p225.setLayout(new GridLayout(1,2,4,4));
p225.add(calibLABEL); p225.add(p215);

optionsPanel.setLayout(new GridLayout(8,1,4,4));
optionsPanel.add(p221);
optionsPanel.add(p222);
optionsPanel.add(p223);
optionsPanel.add(p224);
optionsPanel.add(p225);
optionsPanel.add(auto12);
optionsPanel.add(auto23);
optionsPanel.add(autoXpnt);

//autoVessel.hide();//ignoring the autoVessel

//widthChoice.disable();
vesselChoice.disable();
gen1Choice.disable();
gen2Choice.disable();
gen3Choice.disable();

////////////////////////////////////
//infoPanel
sexChoice.addItem("Male");
sexChoice.addItem("Female");

lrChoice.addItem("Right Eye");
lrChoice.addItem("Left Eye");

infoPanel.setLayout(new GridLayout(10,4,4,4));

infoPanel.add(lrChoice); infoPanel.add(sexChoice); infoPanel.add(ageLABEL);
infoPanel.add(ageTF);

```



```

        infoPanel.add(lnameLABEL);    infoPanel.add(lnameTF);
infoPanel.add(fnameLABEL);    infoPanel.add(fnameTF);
        infoPanel.add(weeksLABEL);    infoPanel.add(weeksTF);    infoPanel.add(diagLABEL);
infoPanel.add(diagTF);
        infoPanel.add(VAR_LABEL);    infoPanel.add(VAR_TF);    infoPanel.add(VAL_LABEL);
infoPanel.add(VAL_TF);
        infoPanel.add(refrLABEL);    infoPanel.add(refrTF);    infoPanel.add(reflLABEL);
infoPanel.add(reflTF);
        infoPanel.add(camLABEL);    infoPanel.add(camTF);    infoPanel.add(magLABEL);
infoPanel.add(magTF);
        infoPanel.add(emailLABEL);    infoPanel.add(emailTF);    infoPanel.add(addLABEL);
infoPanel.add(addTF);

        infoPanel.add(fileLABEL);    infoPanel.add(fileTF);    infoPanel.add(caseLABEL);
infoPanel.add(caseTF);
        infoPanel.add(idLABEL);    infoPanel.add(idTF);
infoPanel.add(dateLABEL);    infoPanel.add(dateTF);
        infoPanel.add(opLABEL);    infoPanel.add(opTF);
infoPanel.add(loadInfo);    infoPanel.add(clearInfo);

        Date theDate =new Date();
        if (theDate.getMonth()>9)
            dateTF.setText(""+theDate.getYear()+"0"+(theDate.getMonth()+1)+theDate.getDate());
        else
            dateTF.setText(""+theDate.getYear()+"0"+(theDate.getMonth()+1)+theDate.getDate());
        ////////////////////////////////////////////////////
        //outputPanel
        outputPanel.setLayout(new BorderLayout(4,4));
        outputPanel.add("Center",opAREA);
        p31.setLayout(new GridLayout(1,2,4,4));
        p31.add(clearOp);    p31.add(insertSep);
        p32.setLayout(new GridLayout(1,2,4,4));
        p32.add(startCalc);    p32.add(opInfo);
        p33.setLayout(new GridLayout(2,1,4,4));
        p33.add(p31);
        p33.add(p32);
        outputPanel.add("South",p33);
        ////////////////////////////////////////////////////
        //imagePanel
        imagePanel.setBackground(Color.white);
        imagePanel.setLayout(new BorderLayout(5,5));
        imagePanel.add("Center",imgp);
        ////////////////////////////////////////////////////
        //statusPanel
        statusPanel.setLayout(new GridLayout(2,1));
        statusPanel.add(display);
        statusPanel.add(dspMode);
        ////////////////////////////////////////////////////
        //extraPanel
        p9.setLayout(new GridLayout(1,2,4,4));
        p9.add(about);    p9.add(help);
        extraPanel.setLayout(new GridLayout(5,1,4,4));
        extraPanel.add(p9);
        extraPanel.add(xtraLB1);
        extraPanel.add(xtraLB2);
        extraPanel.add(xtraLB3);
        extraPanel.add(xtraLB4);
        ////////////////////////////////////////////////////
        // START LAYOUT -GRIDBAG-
        ////////////////////////////////////////////////////
        Panel pg1 = new Panel();
        pg1.setLayout(new BorderLayout(4,4));
        pg1.add("North",extraPanel);
        pg1.add("Center",imageLoadPanel);
        pg1.add("South",outputPanel);

        Panel pg2=new Panel();
        pg2.setLayout(new BorderLayout(4,4));
        pg2.add("North",optionsPanel);
        pg2.add("Center",infoPanel);

        GridBagLayout gridbag = new GridBagLayout();
        GridBagConstraints c = new GridBagConstraints();
        setLayout(gridbag);
        c.insets=new Insets(4,4,4,4);
        c.anchor=GridBagConstraints.NORTH;

        c.fill = GridBagConstraints.HORIZONTAL;
        c.gridwidth=10;
        c.weightx=0.7;
        gridbag.setConstraints(pg1, c);
        add(pg1);

        c.gridwidth=GridBagConstraints.REMAINDER;//last in row
        c.weightx=0.3;
        gridbag.setConstraints(pg2, c);
        add(pg2);

        gridbag.setConstraints(statusPanel, c);
        add(statusPanel);

        c.weighty=1;
        c.weightx=0;

```

```

c.gridwidth=1;
c.fill = GridBagConstraints.VERTICAL;
gridbag.setConstraints(inputPanel, c);
add(inputPanel);

//c.weightx=1;
c.gridwidth=GridBagConstraints.REMAINDER;//last in row
c.fill = GridBagConstraints.BOTH;
gridbag.setConstraints(imagePanel, c);
add(imagePanel);
}

//////////////////////////////////////
// ACTION
//////////////////////////////////////
public boolean action(Event e, Object o)
{
    //COMMON
    //Any start disables: All starts buttons & All auto checkboxes
    if(e.target==startContour1 || e.target==startContour2 ||
       e.target==startContour3 || e.target==setPoint ||
       e.target==startVessel || e.target==startGen1 ||
       e.target==startGen2 || e.target==startGen3)
    {
        startContour1.disable();
        startContour2.disable();
        startContour3.disable();
        setPoint.disable();
        startVessel.disable();
        startGen1.disable();
        startGen2.disable();
        startGen3.disable();

        auto12.disable();
        auto23.disable();
        autoXpnt.disable();
        delLastVessel.disable();
        delLastGen1.disable();
        delLastGen2.disable();
        delLastGen3.disable();
        //autoVessel.disable();
    }
    //Any end enables: All start buttons (if auto allows) & All auto checkboxes
    if(e.target==endContour1 || e.target==endContour2 ||
       e.target==endContour3 || e.target==endPoint ||
       e.target==endVessel || e.target==endGen1 ||
       e.target==endGen2 || e.target==endGen3)
    {
        if(!(auto12.getState() || auto23.getState() || autoXpnt.getState() /*||
autoVessel.getState()*/))
            //if all checkboxes are off=false
            {
                auto12.enable();
                auto23.enable();
                autoXpnt.enable();
                //autoVessel.enable();
            }
        if( (auto12.getState()      && (e.target==endContour1))
           ||(auto23.getState()    && (e.target==endContour2))
           ||(autoXpnt.getState()  && (e.target==endContour3))
           /*||(autoVessel.getState())&& (e.target==endPoint)*/)
        {
            startContour1.disable();
            startContour2.disable();
            startContour3.disable();
            setPoint.disable();
            startVessel.disable();
            startGen1.disable();
            startGen2.disable();
            startGen3.disable();
        }
        else
        {
            startContour1.enable();
            if(!auto12.getState())startContour2.enable();
            if(!auto23.getState())startContour3.enable();
            if(!autoXpnt.getState())setPoint.enable();
            //if(!autoVessel.getState())startVessel.enable();
            startVessel.enable();
            startGen1.enable();
            startGen2.enable();
            startGen3.enable();

            if(vessels.size()>=1 &&
            (((Vessel)vessels.lastElement()).nPoints()>=2))delLastVessel.enable();
            if(gen1s.size()>=1 && (((Vessel)gen1s.lastElement()).nPoints()
            >=2))delLastGen1.enable();
            if(gen2s.size()>=1 && (((Vessel)gen2s.lastElement()).nPoints()
            >=2))delLastGen2.enable();
            if(gen3s.size()>=1 && (((Vessel)gen3s.lastElement()).nPoints()
            >=2))delLastGen3.enable();
        }
        repaint();
        display.setText(aboutStr);
    }
}

//////////////////////////////////////
//ABOUT+HELP
//////////////////////////////////////

```

```

if (e.target==help)
{
    AppletContext theAppContext = getAppletContext();

    try
    {
        theAppContext.showDocument(new
URL((getCodeBase().toString()+"help.htm"), "_blank");
    }
    catch (MalformedURLException ex)
    {
        display.setText("MalformedURLException thrown. Unable to display the Help
information");
        return true;
    }
}
else if (e.target==about)
{
    myDialog aboutDialog;
    Frame aboutParent=getTopLevelParent((Component)about);
    if (aboutParent!=null)
    {
        aboutDialog=new myDialog(aboutParent,"About...",
Technology, Sweden. 1998");
        aboutDialog.show();
    }
    else
        display.setText("Unable to display the About dialog");
}
////////////////////////////////////
//IMAGE VIEW
////////////////////////////////////
else if (e.target==viewURL)
{
    URL ImgURL;
    String s=imgTF.getText();
    try{ImgURL=new URL(s);}
    catch (MalformedURLException ex)
        {display.setText("Bad URL");return true;}
    Img=getImage(ImgURL);

    if (!prepareImage(Img, this))
    {
        display.setText("Error preparing URL image");
        return true;
    }
    dspImgMsg=true;
    repaint();
}
else if (e.target==viewLoc)
{
    Img=getImage(getCodeBase(), locimgTF.getText());
    if (!prepareImage(Img, this))
    {
        display.setText("Error preparing local image");
        return true;
    }
    dspImgMsg=true;
    repaint();
}
else if (e.target==viewChoice)
{
    switch (imgChoice.getSelectedIndex())
    {
        case 0:Img=getImage(getCodeBase(),"sampleImg1.gif");break;
        case 1:Img=getImage(getCodeBase(),"sampleImg2.gif");break;
        case 2:Img=getImage(getCodeBase(),"sampleImg3.gif");break;
        case 3:Img=getImage(getCodeBase(),"sampleImg4.gif");break;
        default:Img=getImage(getCodeBase(),"sampleImg1.gif");break;
    }
    if (!prepareImage(Img, this))
    {
        display.setText("Error preparing choice image");
        return true;
    }
    dspImgMsg=true;
    repaint();
}
////////////////////////////////////
//INNER & OUTER CIRCLES
////////////////////////////////////
else if (e.target==diamTFin)
{
    try
    {
        Double tD=new Double(diamTFin.getText());
        mmDin=tD.doubleValue();
        if (mmDin<=0)throw new NumberFormatException();
        display.setText("Inner diameter updated");
    }
    catch (NumberFormatException ex)
    {
        display.setText("Type a valid number. Inner diameter must be a real
positive number");
    }
    pxlDin_x=(int)(mmDin*pxl_per_mm_x);
    pxlDin_y=(int)(mmDin*pxl_per_mm_y);
    repaint();
}
else if (e.target==updateInner)
{
    //same as above
}

```

```

        try
        {
            Double tD=new Double(diamTFin.getText());
            mmDin=tD.doubleValue();
            if(mmDin<=0)throw new NumberFormatException();
            display.setText("Inner diameter updated");
        }
        catch(NumberFormatException ex)
        {
            display.setText("Type a valid number. Inner diameter must be a real
positive number");
        }
        pxlDin_x=(int)(mmDin*pxl_per_mm_x);
        pxlDin_y=(int)(mmDin*pxl_per_mm_y);
        repaint();
    }
    else if(e.target==diamTFut)
    {
        try
        {
            Double tD=new Double(diamTFut.getText());
            mmDut=tD.doubleValue();
            if(mmDut<=0)throw new NumberFormatException();
            display.setText("Outer diameter updated");
        }
        catch(NumberFormatException ex)
        {
            display.setText("Type a valid number. Outer diameter must be a real
positive number");
        }

        pxlDut_x=(int)(mmDut*pxl_per_mm_x);
        pxlDut_y=(int)(mmDut*pxl_per_mm_y);
        repaint();
    }
    else if(e.target==updateOuter)
    {
        //same as above
        try
        {
            Double tD=new Double(diamTFut.getText());
            mmDut=tD.doubleValue();
            if(mmDut<=0)throw new NumberFormatException();
            display.setText("Outer diameter updated");
        }
        catch(NumberFormatException ex)
        {
            display.setText("Type a valid number. Outer diameter must be a real
positive number");
        }

        pxlDut_x=(int)(mmDut*pxl_per_mm_x);
        pxlDut_y=(int)(mmDut*pxl_per_mm_y);
        repaint();
    }
    else if(e.target==pxl_per_mm_xTF)
    {
        try
        {
            Integer tD = new Integer(pxl_per_mm_xTF.getText());
            pxl_per_mm_x=tD.intValue();
            if(pxl_per_mm_x<=0)throw new NumberFormatException();
            display.setText("X Pixels per mm updated");
        }
        catch(NumberFormatException ex)
        {
            display.setText("Type a valid number. X Pixels per mm must be a positive
integer");
        }
    }
    else if(e.target==updatePxl_per_mm_x)
    {
        //same as above
        try
        {
            Integer tD = new Integer(pxl_per_mm_xTF.getText());
            pxl_per_mm_x=tD.intValue();
            if(pxl_per_mm_x<=0)throw new NumberFormatException();
            display.setText("X Pixels per mm updated");
        }
        catch(NumberFormatException ex)
        {
            display.setText("Type a valid number. X Pixels per mm must be a positive
integer");
        }
    }
    else if(e.target==pxl_per_mm_yTF)
    {
        try
        {
            Integer tD = new Integer(pxl_per_mm_yTF.getText());
            pxl_per_mm_y=tD.intValue();
            if(pxl_per_mm_y<=0)throw new NumberFormatException();
            display.setText("Y Pixels per mm updated");
        }
        catch(NumberFormatException ex)
        {
            display.setText("Type a valid number. Y Pixels per mm must be a positive
integer");
        }
    }
    else if(e.target==updatePxl_per_mm_y)
    {
        //same as above
        try

```

```

        {
            Integer tD = new Integer(px1_per_mm_yTF.getText());
            px1_per_mm_y=tD.intValue();
            if(px1_per_mm_y<=0)throw new NumberFormatException();
            display.setText("Y Pixels per mm updated");
        }
        catch(NumberFormatException ex)
        {
            display.setText("Type a valid number. Y Pixels per mm must be a positive
integer");
        }
    }
    else if(e.target==updateCalib)
    {
        ///////////////////////////////////////////////////////////////////
        //OPTIONS
        ///////////////////////////////////////////////////////////////////
    else if(e.target==auto12)
    {
        if(auto12.getState())startContour2.disable();
        else startContour2.enable();
    }
    else if(e.target==auto23)
    {
        if(auto23.getState())startContour3.disable();
        else startContour3.enable();
    }
    else if(e.target==autoXpnt)
    {
        if(autoXpnt.getState())setPoint.disable();
        else setPoint.enable();
    }
    /*else if(e.target==autoVessel)
    {
        if(autoVessel.getState())startVessel.disable();
        else startVessel.enable();
    }*/

    ///////////////////////////////////////////////////////////////////
    // INFORMATION
    ///////////////////////////////////////////////////////////////////
    else if(e.target==loadInfo)
    {
        String LNAME="NA";
        String FNAME="NA";
        String RL="NA";
        String MF="NA";
        String AGE="NA";
        String WEEKS="NA";
        String DIAG="NA";
        String VAR="NA";
        String VAL="NA";
        String REFR="NA";
        String REFL="NA";
        String CAM="NA";
        String MAG="NA";
        String EMAIL="NA";
        String ADD="NA";
        String CASE="NA";
        String ID="NA";
        String DATE="NA";
        String OPER="NA";

        boolean err_read=false;
        try
        {
            String infoFile=(getCodeBase()).toString()+fileTF.getText();
            URL url = new URL(infoFile);
            DataInputStream input = new DataInputStream(url.openStream());

            LNAME=input.readLine();
            FNAME=input.readLine();
            MF=input.readLine();
            AGE=input.readLine();
            WEEKS=input.readLine();
            DIAG=input.readLine();
            VAR=input.readLine();
            VAL=input.readLine();
            REFR=input.readLine();
            REFL=input.readLine();
            CAM=input.readLine();
            MAG=input.readLine();
            EMAIL=input.readLine();
            ADD=input.readLine();
            display.setText("Information read from Info File");
        }
        catch (IOException ex)
        {
            //display.setText(ex.getMessage());
            display.setText("Error reading from Info File - IOException");
            err_read=true;
        }
        if(!err_read)
        {
            try{
                //refTF.setText(Ref);
                //nameTF.setText(Name);
                //idTF.setText(ID);
                //caseTF.setText(CaseNo);

                if(RL.charAt(0)=='r' || RL.charAt(0)=='R')lrChoice.select(0);else

```

```

        if(MF.charAt(0)=='m' ||
MF.charAt(0)=='M')sexChoice.select(0);else sexChoice.select(1);
        lnameTF.setText(LNAME);
        fnameTF.setText(FNAME);
        ageTF.setText(AGE);
        weeksTF.setText(WEEKS);
        diagTF.setText(DIAG);
        VAR_TF.setText(VAR);
        VAL_TF.setText(VAL);
        refrTF.setText(REFR);
        reflTF.setText(REFL);
        camTF.setText(CAM);
        magTF.setText(MAG);
        emailTF.setText(EMAIL);
        addTF.setText(ADD);

        //infoPanel.add(fileLABEL); infoPanel.add(fileTF
        //infoPanel.add(caseLABEL); infoPanel.add(caseTF
        //infoPanel.add(idLABEL); infoPanel.add(idTF
        //infoPanel.add(dateLABEL); infoPanel.add(dateTF
        //infoPanel.add(opLABEL); infoPanel.add(opTF

        display.setText("Information read and interpreted from Info
File");
    }
    catch(Exception ex)
    {
        display.setText("Error interpreting data from Info File");
    }
}
else if(e.target==clearInfo)
{
    //lrChoice.select(0);
    //sexChoice.select(0);
    //refrTF.setText("");
    //nameTF.setText("");
    //caseTF.setText("");
    //idTF.setText("");
    //diagTF.setText("");
    //opTF.setText("");
    lrChoice.select(0);
    sexChoice.select(0);
    lnameTF.setText("");
    fnameTF.setText("");
    ageTF.setText("");
    weeksTF.setText("");
    diagTF.setText("");
    VAR_TF.setText("");
    VAL_TF.setText("");
    refrTF.setText("");
    reflTF.setText("");
    camTF.setText("");
    magTF.setText("");
    emailTF.setText("");
    addTF.setText("");
}
//loadInfo
////////////////////////////////////
//INPUT START
////////////////////////////////////
else if(e.target==startContour1)
{
    mode=CNT1;
    contour1.clear();
    endContour1.enable();
    display.setText("Click the points for "+modeNames[1]+" then End");
}
else if(e.target==startContour2)
{
    mode=CNT2;
    contour2.clear();
    endContour2.enable();
    display.setText("Click the points for "+modeNames[2]+" then End");
}
else if(e.target==startContour3)
{
    mode=CNT3;
    contour3.clear();
    endContour3.enable();
    display.setText("Click the points for "+modeNames[3]+" then End");
}
else if(e.target==setPoint)
{
    mode=XPNT;
    endPoint.enable();
    display.setText("Click the "+modeNames[4]+" then End");
}
else if(e.target==startVessel)
{
    mode=VSSL;
    endVessel.enable();
    vesselChoice.enable();
    //widthChoice.enable();
    vessels.addElement(new Vessel());
    ((Vessel)vessels.lastElement()).setType(vesselChoice.getSelectedIndex());
    ((Vessel)vessels.lastElement()).setWidth(widthChoice.getSelectedIndex());
    display.setText("Click the points for one of the "+modeNames[5]+" then End");
}
else if(e.target==startGen1)
{
    mode=GEN1;

```

```

        endGen1.enable();
        gen1Choice.enable();
        //widthChoice.enable();
        gen1s.addElement(new Vessel());
        ((Vessel)gen1s.lastElement()).setType(gen1Choice.getSelectedIndex());
        //((Vessel)vessels.lastElement()).setWidth(widthChoice.getSelectedIndex());
        display.setText("Click the points for one of the "+modeNames[6]+" then End");
    }
    else if(e.target==startGen2)
    {
        mode=GEN2;
        endGen2.enable();
        gen2Choice.enable();
        //widthChoice.enable();
        gen2s.addElement(new Vessel());
        ((Vessel)gen2s.lastElement()).setType(gen2Choice.getSelectedIndex());
        //((Vessel)vessels.lastElement()).setWidth(widthChoice.getSelectedIndex());
        display.setText("Click the points for one of the "+modeNames[7]+" then End");
    }
    else if(e.target==startGen3)
    {
        mode=GEN3;
        endGen3.enable();
        gen3Choice.enable();
        //widthChoice.enable();
        gen3s.addElement(new Vessel());
        ((Vessel)gen3s.lastElement()).setType(gen3Choice.getSelectedIndex());
        //((Vessel)vessels.lastElement()).setWidth(widthChoice.getSelectedIndex());
        display.setText("Click the points for one of the "+modeNames[8]+" then End");
    }
    ////////////////////////////////////////////////////
    //INPUT END
    ////////////////////////////////////////////////////
    else if(e.target==endContour1)
    {
        endContour1.disable();
        if(auto12.getState())
        {
            mode=CNT2;
            contour2.clear();
            endContour2.enable();
            display.setText("Click the points for "+modeNames[2]+" then End");
        }
        else
            mode=NONE;
    }
    else if(e.target==endContour2)
    {
        endContour2.disable();
        if(auto23.getState())
        {
            mode=CNT3;
            contour3.clear();
            endContour3.enable();
            display.setText("Click the points for "+modeNames[3]+" then End");
        }
        else
            mode=NONE;
    }
    else if(e.target==endContour3)
    {
        endContour3.disable();
        if(autoXpnt.getState())
        {
            mode=XPNT;
            endPoint.enable();
            endPoint.enable();
            display.setText("Click the "+modeNames[4]+" then End");
        }
        else
            mode=NONE;
    }
    else if(e.target==endPoint)
    {
        endPoint.disable();
        mode=NONE;
    }
    else if(e.target==endVessel)
    {
        endVessel.disable();
        vesselChoice.disable();
        //widthChoice.disable();
        mode=NONE;
        ((Vessel)vessels.lastElement()).setType(vesselChoice.getSelectedIndex());
        //((Vessel)vessels.lastElement()).setWidth(widthChoice.getSelectedIndex());
        if(((Vessel)vessels.lastElement()).nPoints()<2)
        {
            vessels.removeElementAt(vessels.size()-1);
            display.setText("Vessel ignored, contains less than 2 points");
        }
    }
    else if(e.target==endGen1)
    {
        endGen1.disable();
        gen1Choice.disable();
        //widthChoice.disable();
        mode=NONE;
        ((Vessel)gen1s.lastElement()).setType(gen1Choice.getSelectedIndex());
        //((Vessel)vessels.lastElement()).setWidth(widthChoice.getSelectedIndex());
        if(((Vessel)gen1s.lastElement()).nPoints()<2)
        {

```

```

        gen1s.removeElementAt(gen1s.size()-1);
        display.setText("Generation1 Vessel ignored, contains less than 2 points");
    }
}
else if(e.target==endGen2)
{
    endGen2.disable();
    gen2Choice.disable();
    //widthChoice.disable();
    mode=NONE;
    ((Vessel)gen2s.lastElement()).setType(gen2Choice.getSelectedIndex());
    //((Vessel)vessels.lastElement()).setWidth(widthChoice.getSelectedIndex());
    if(((Vessel)gen2s.lastElement()).nPoints()<2)
    {
        gen2s.removeElementAt(gen2s.size()-1);
        display.setText("Generation2 Vessel ignored, contains less than 2 points");
    }
}
else if(e.target==endGen3)
{
    endGen3.disable();
    gen3Choice.disable();
    //widthChoice.disable();
    mode=NONE;
    ((Vessel)gen3s.lastElement()).setType(gen3Choice.getSelectedIndex());
    //((Vessel)vessels.lastElement()).setWidth(widthChoice.getSelectedIndex());
    if(((Vessel)gen3s.lastElement()).nPoints()<2)
    {
        gen3s.removeElementAt(gen3s.size()-1);
        display.setText("Generation3 Vessel ignored, contains less than 2 points");
    }
}
}
//INPUT DEL LAST
//INPUT DEL LAST
else if(e.target==delLastVessel)
{
    vessels.removeElementAt(vessels.size()-1);
    if(vessels.size()>=1) delLastVessel.enable(); else delLastVessel.disable();
    repaint();
}
else if(e.target==delLastGen1)
{
    gen1s.removeElementAt(gen1s.size()-1);
    if(gen1s.size()>=1) delLastGen1.enable(); else delLastGen1.disable();
    repaint();
}
else if(e.target==delLastGen2)
{
    gen2s.removeElementAt(gen2s.size()-1);
    if(gen2s.size()>=1) delLastGen2.enable(); else delLastGen2.disable();
    repaint();
}
else if(e.target==delLastGen3)
{
    gen3s.removeElementAt(gen3s.size()-1);
    if(gen3s.size()>=1) delLastGen3.enable(); else delLastGen3.disable();
    repaint();
}
//VESSEL TYPE CHOICE
//VESSEL TYPE CHOICE
else if(e.target==vesselChoice)
{
    ((Vessel)vessels.lastElement()).setType(vesselChoice.getSelectedIndex());
}
else if(e.target==gen1Choice)
{
    ((Vessel)gen1s.lastElement()).setType(gen1Choice.getSelectedIndex());
}
else if(e.target==gen2Choice)
{
    ((Vessel)gen2s.lastElement()).setType(gen2Choice.getSelectedIndex());
}
else if(e.target==gen3Choice)
{
    ((Vessel)gen3s.lastElement()).setType(gen3Choice.getSelectedIndex());
}
}
//INPUT TO PRODUCE OUTPUT
//INPUT TO PRODUCE OUTPUT
//calculations
else if(e.target==startCalc)
{
    opAREA.appendText("\n-----");

    //Branshing points
    //Total Branshing points
    int nBP = 0;
    //Branshing Points for Arteries
    int nBP4A=0; //gen1_arteries + gen2_arteries + gen3_arteries
    int nBP4A_1=0; //gen1_arteries
    int nBP4A_2=0; //gen2_arteries
    int nBP4A_3=0; //gen3_arteries
    //Total Branshing Points for Veins
    int nBP4V=0; //gen1_veins + gen2_veins + gen3_veins
}

```



```

int nBP4V_1=0; //gen1_veins
int nBP4V_2=0; //gen2_veins
int nBP4V_3=0; //gen3_veins
//Total Branching Points for Unspecified
int nBP4U=0; //gen1_unspec + gen2_unspec + gen3_unspec
int nBP4U_1=0; //gen1_unspec
int nBP4U_2=0; //gen2_unspec
int nBP4U_3=0; //gen3_unspec

int k;

k=gen1s.size();
for(int i=0;i<k;i++)
{
    if(((Vessel)gen1s.elementAt(i)).getType()==0)nBP4A_1++;
    else if(((Vessel)gen1s.elementAt(i)).getType()==1) nBP4V_1++;
    else if(((Vessel)gen1s.elementAt(i)).getType()==2) nBP4U_1++;
}

//GEN2(S)
k=gen2s.size();
for(int i=0;i<k;i++)
{
    if(((Vessel)gen2s.elementAt(i)).getType()==0) nBP4A_2++;
    else if(((Vessel)gen2s.elementAt(i)).getType()==1) nBP4V_2++;
    else if(((Vessel)gen2s.elementAt(i)).getType()==2) nBP4U_2++;
}

//GEN3(S)
k=gen3s.size();
for(int i=0;i<k;i++)
{
    if(((Vessel)gen3s.elementAt(i)).getType()==0) nBP4A_3++;
    else if(((Vessel)gen3s.elementAt(i)).getType()==1) nBP4V_3++;
    else if(((Vessel)gen3s.elementAt(i)).getType()==2) nBP4U_3++;
}

nBP4A=nBP4A_1 + nBP4A_2 + nBP4A_3;
nBP4V=nBP4V_1 + nBP4V_2 + nBP4V_3;
nBP4U=nBP4U_1 + nBP4U_2 + nBP4U_3;
nBP=nBP4A + nBP4V + nBP4U;

opAREA.appendText("\nTotal number of Branching Points = "+nBP);
opAREA.appendText("\n    Number of Branching Points for Arteries    = "+nBP4A);
opAREA.appendText("\n    Gen1 = "+nBP4A_1+", Gen2 = "+nBP4A_2+", Gen3 = "+nBP4A_3);
opAREA.appendText("\n    Number of Branching Points for Veins    = "+nBP4V);
opAREA.appendText("\n    Gen1 = "+nBP4V_1+", Gen2 = "+nBP4V_2+", Gen3 = "+nBP4V_3);
opAREA.appendText("\n    Number of Branching Points for Unspecified = "+nBP4U);
opAREA.appendText("\n    Gen1 = "+nBP4U_1+", Gen2 = "+nBP4U_2+", Gen3 = "+nBP4U_3);

//////////
//Indices of Tortosity
//////////

double sumTotA_v=0.0; //total curved length for arteries_vessels
double sumLinA_v=0.0; //total line length for arteries_vessels
double torIndA_v=-1.0; //tortosity index for arteries_vessels

double sumTotA_1=0.0; //total curved length for arteries_g1
double sumLinA_1=0.0; //total line length for arteries_g1
double torIndA_1=-1.0; //tortosity index for arteries_g1

double sumTotA_2=0.0; //total curved length for arteries_g2
double sumLinA_2=0.0; //total line length for arteries_g2
double torIndA_2=-1.0; //tortosity index for arteries_g2

double sumTotA_3=0.0; //total curved length for arteries_g3
double sumLinA_3=0.0; //total line length for arteries_g3
double torIndA_3=-1.0; //tortosity index for arteries_g3

//////////

double sumTotV_v=0.0; //total curved length for veins_vessels
double sumLinV_v=0.0; //total line length for veins_vessels
double torIndV_v=-1.0; //tortosity index for veins_vessels

double sumTotV_1=0.0; //total curved length for veins_g1
double sumLinV_1=0.0; //total line length for veins_g1
double torIndV_1=-1.0; //tortosity index for veins_g1

double sumTotV_2=0.0; //total curved length for veins_g2
double sumLinV_2=0.0; //total line length for veins_g2
double torIndV_2=-1.0; //tortosity index for veins_g2

double sumTotV_3=0.0; //total curved length for veins_g3
double sumLinV_3=0.0; //total line length for veins_g3
double torIndV_3=-1.0; //tortosity index for veins_g3

//////////

double sumTotU_v=0.0; //total curved length for unspecified_vessels
double sumLinU_v=0.0; //total line length for unspecified_vessels
double torIndU_v=-1.0; //tortosity index for unspecified_vessels

double sumTotU_1=0.0; //total curved length for unspecified_g1
double sumLinU_1=0.0; //total line length for unspecified_g1
double torIndU_1=-1.0; //tortosity index for unspecified_g1

double sumTotU_2=0.0; //total curved length for unspecified_g2

```

```

double sumLinU_2=0.0; //total line length for unspecified_g2
double torIndU_2=-1.0; //tortousity index for unspecified_g2

double sumTotU_3=0.0; //total curved length for unspecified_g2
double sumLinU_3=0.0; //total line length for unspecified_g2
double torIndU_3=-1.0; //tortousity index for unspecified_g2

// derived variables

double sumTot=0.0; //total curved length for everything
double sumLin=0.0; //total line length for everything
double torInd=-1.0; //tortousity index for everything

double sumTotA=0.0; //total curved length for arteries
double sumLinA=0.0; //total line length for arteries
double torIndA=-1.0; //tortousity index for arteries

double sumTotV=0.0; //total curved length for veins
double sumLinV=0.0; //total line length for veins
double torIndV=-1.0; //tortousity index for veins

double sumTotU=0.0; //total curved length for unspecified
double sumLinU=0.0; //total line length for unspecified
double torIndU=-1.0; //tortousity index for unspecified

double sumTotVssl=0.0; //total curved length for vessels
double sumLinVssl=0.0; //total line length for vessels
double torIndVssl=-1.0; //tortousity index for vessels

double sumTotGen1=0.0; //total curved length for gen1s
double sumLinGen1=0.0; //total line length for gen1s
double torIndGen1=-1.0; //tortousity index for gen1s

double sumTotGen2=0.0; //total curved length for gen2s
double sumLinGen2=0.0; //total line length for gen2s
double torIndGen2=-1.0; //tortousity index for gen2s

double sumTotGen3=0.0; //total curved length for gen3s
double sumLinGen3=0.0; //total line length for gen3s
double torIndGen3=-1.0; //tortousity index for gen3s

//Summing Up Lengths
//vessels:AVU
for(int i=0;i<vessels.size();i++)
{
    double
t1_mm=((Vessel)vessels.elementAt(i)).getTotalLength(mmDut,xPnt,pxl_per_mm_x,pxl_per_mm_y);
    double
l1_mm=((Vessel)vessels.elementAt(i)).getLineLength(mmDut,xPnt,pxl_per_mm_x,pxl_per_mm_y);
    if(t1_mm>0 && l1_mm>0)
    {
        if(((Vessel)vessels.elementAt(i)).getType()==0)
        {
            sumTotA_v+=t1_mm;
            sumLinA_v+=l1_mm;
        }
        else if(((Vessel)vessels.elementAt(i)).getType()==1)
        {
            sumTotV_v+=t1_mm;
            sumLinV_v+=l1_mm;
        }
        else if(((Vessel)vessels.elementAt(i)).getType()==2)
        {
            sumTotU_v+=t1_mm;
            sumLinU_v+=l1_mm;
        }
    }
}

//genls:AVU
for(int i=0;i<genls.size();i++)
{
    double
t1_mm=((Vessel)genls.elementAt(i)).getTotalLength(mmDut,xPnt,pxl_per_mm_x,pxl_per_mm_y);
    double
l1_mm=((Vessel)genls.elementAt(i)).getLineLength(mmDut,xPnt,pxl_per_mm_x,pxl_per_mm_y);
    if(t1_mm>0 && l1_mm>0)
    {
        if(((Vessel)genls.elementAt(i)).getType()==0)
        {
            sumTotA_1+=t1_mm;
            sumLinA_1+=l1_mm;
        }
        else if(((Vessel)genls.elementAt(i)).getType()==1)
        {
            sumTotV_1+=t1_mm;
            sumLinV_1+=l1_mm;
        }
        else if(((Vessel)genls.elementAt(i)).getType()==2)
        {
            sumTotU_1+=t1_mm;
            sumLinU_1+=l1_mm;
        }
    }
}

```

```

//gen2s:AVU
for(int i=0;i<gen2s.size();i++)
{
    double
t1_mm=((Vessel)gen2s.elementAt(i)).getTotalLength(mmDut,xPnt,pxl_per_mm_x,pxl_per_mm_y);
    double
ll_mm=((Vessel)gen2s.elementAt(i)).getLineLength(mmDut,xPnt,pxl_per_mm_x,pxl_per_mm_y);
    if(t1_mm>0 && ll_mm>0)
    {
        if(((Vessel)gen2s.elementAt(i)).getType()==0)
        {
            sumTotA_2+=t1_mm;
            sumLinA_2+=ll_mm;
        }
        else if(((Vessel)gen2s.elementAt(i)).getType()==1)
        {
            sumTotV_2+=t1_mm;
            sumLinV_2+=ll_mm;
        }
        else if(((Vessel)gen2s.elementAt(i)).getType()==2)
        {
            sumTotU_2+=t1_mm;
            sumLinU_2+=ll_mm;
        }
    }
}

//gen3s:AVU
for(int i=0;i<gen3s.size();i++)
{
    double
t1_mm=((Vessel)gen3s.elementAt(i)).getTotalLength(mmDut,xPnt,pxl_per_mm_x,pxl_per_mm_y);
    double
ll_mm=((Vessel)gen3s.elementAt(i)).getLineLength(mmDut,xPnt,pxl_per_mm_x,pxl_per_mm_y);
    if(t1_mm>0 && ll_mm>0)
    {
        if(((Vessel)gen3s.elementAt(i)).getType()==0)
        {
            sumTotA_3+=t1_mm;
            sumLinA_3+=ll_mm;
        }
        else if(((Vessel)gen3s.elementAt(i)).getType()==1)
        {
            sumTotV_3+=t1_mm;
            sumLinV_3+=ll_mm;
        }
        else if(((Vessel)gen3s.elementAt(i)).getType()==2)
        {
            sumTotU_3+=t1_mm;
            sumLinU_3+=ll_mm;
        }
    }
}

//calculating
sumTotA=sumTotA_v+sumTotA_1+sumTotA_2+sumTotA_3;
sumLinA=sumLinA_v+sumLinA_1+sumLinA_2+sumLinA_3;

sumTotV=sumTotV_v+sumTotV_1+sumTotV_2+sumTotV_3;
sumLinV=sumLinV_v+sumLinV_1+sumLinV_2+sumLinV_3;

sumTotU=sumTotU_v+sumTotU_1+sumTotU_2+sumTotU_3;
sumLinU=sumLinU_v+sumLinU_1+sumLinU_2+sumLinU_3;

sumTotVssl=sumTotA_v+sumTotV_v+sumTotU_v;
sumLinVssl=sumLinA_v+sumLinV_v+sumLinU_v;

sumTotGen1=sumTotA_1+sumTotV_1+sumTotU_1;
sumLinGen1=sumLinA_1+sumLinV_1+sumLinU_1;

sumTotGen2=sumTotA_2+sumTotV_2+sumTotU_2;
sumLinGen2=sumLinA_2+sumLinV_2+sumLinU_2;

sumTotGen3=sumTotA_3+sumTotV_3+sumTotU_3;
sumLinGen3=sumLinA_3+sumLinV_3+sumLinU_3;

sumTot=sumTotA+sumTotV+sumTotU;
sumLin=sumLinA+sumLinV+sumLinU;

torInd=sumTot/sumLin;
torIndA=sumTotA/sumLinA;
torIndV=sumTotV/sumLinV;
torIndU=sumTotU/sumLinU;

torIndVssl=sumTotVssl/sumLinVssl;
torIndGen1=sumTotGen1/sumLinGen1;
torIndGen2=sumTotGen2/sumLinGen2;
torIndGen3=sumTotGen3/sumLinGen3;

torIndA_v=sumTotA_v/sumLinA_v;
torIndA_1=sumTotA_1/sumLinA_1;
torIndA_2=sumTotA_2/sumLinA_2;
torIndA_3=sumTotA_3/sumLinA_3;

torIndV_v=sumTotV_v/sumLinV_v;

```

```

torIndV_1=sumTotV_1/sumLinV_1;
torIndV_2=sumTotV_2/sumLinV_2;
torIndV_3=sumTotV_3/sumLinV_3;

torIndU_v=sumTotU_v/sumLinU_v;
torIndU_1=sumTotU_1/sumLinU_1;
torIndU_2=sumTotU_2/sumLinU_2;
torIndU_3=sumTotU_3/sumLinU_3;

//outputing
//Everything
opAREA.appendText("\nIndex of Tortuosity for Everything = ");
if(!(new Double(torInd)).isNaN())opAREA.appendText(""+torInd);
else opAREA.appendText("Not Available");
//Arteries
opAREA.appendText("\n Index of Tortuosity for All Arteries = ");
if(!(new Double(torIndA)).isNaN())opAREA.appendText(""+torIndA);
else opAREA.appendText("Not Available");
//Veins
opAREA.appendText(", All Veins = ");
if(!(new Double(torIndV)).isNaN())opAREA.appendText(""+torIndV);
else opAREA.appendText("Not Available");
//Unspecs
opAREA.appendText(", All Unspecified = ");
if(!(new Double(torIndU)).isNaN())opAREA.appendText(""+torIndU);
else opAREA.appendText("Not Available");
//Vessels
opAREA.appendText("\n Index of Tortuosity for All Vessels = ");
if(!(new Double(torIndVssl)).isNaN())opAREA.appendText(""+torIndVssl);
else opAREA.appendText("Not Available");
//Gen1s
opAREA.appendText(", All Gen1s = ");
if(!(new Double(torIndGen1)).isNaN())opAREA.appendText(""+torIndGen1);
else opAREA.appendText("Not Available");
//Gen2s
opAREA.appendText(", All Gen2s = ");
if(!(new Double(torIndGen2)).isNaN())opAREA.appendText(""+torIndGen2);
else opAREA.appendText("Not Available");
//Gen3s
opAREA.appendText(", All Gen3s = ");
if(!(new Double(torIndGen3)).isNaN())opAREA.appendText(""+torIndGen3);
else opAREA.appendText("Not Available");
//Arteries_Vessels
opAREA.appendText("\n Index of Tortuosity for Vessels of Arteries = ");
if(!(new Double(torIndA_v)).isNaN())opAREA.appendText(""+torIndA_v);
else opAREA.appendText("Not Available");
//Arteries_Gen1
opAREA.appendText(", Gen1s of Arteries = ");
if(!(new Double(torIndA_1)).isNaN())opAREA.appendText(""+torIndA_1);
else opAREA.appendText("Not Available");
//Arteries_Gen2
opAREA.appendText(", Gen2s of Arteries = ");
if(!(new Double(torIndA_2)).isNaN())opAREA.appendText(""+torIndA_2);
else opAREA.appendText("Not Available");
//Arteries_Gen3
opAREA.appendText(", Gen3s of Arteries = ");
if(!(new Double(torIndA_3)).isNaN())opAREA.appendText(""+torIndA_3);
else opAREA.appendText("Not Available");
//Veins_Vessels
opAREA.appendText("\n Index of Tortuosity for Vessels of Veins = ");
if(!(new Double(torIndV_v)).isNaN())opAREA.appendText(""+torIndV_v);
else opAREA.appendText("Not Available");
//Veins_Gen1
opAREA.appendText(", Gen1s of Veins = ");
if(!(new Double(torIndV_1)).isNaN())opAREA.appendText(""+torIndV_1);
else opAREA.appendText("Not Available");
//Veins_Gen2
opAREA.appendText(", Gen2s of Veins = ");
if(!(new Double(torIndV_2)).isNaN())opAREA.appendText(""+torIndV_2);
else opAREA.appendText("Not Available");
//Veins_Gen3
opAREA.appendText(", Gen3s of Veins = ");
if(!(new Double(torIndV_3)).isNaN())opAREA.appendText(""+torIndV_3);
else opAREA.appendText("Not Available");
//Unspecifieds_Vessels
opAREA.appendText("\n Index of Tortuosity for Vessels of Unspecified = ");
if(!(new Double(torIndU_v)).isNaN())opAREA.appendText(""+torIndU_v);
else opAREA.appendText("Not Available");
//Unspecifieds_Gen1
opAREA.appendText(", Gen1s of Unspecified = ");
if(!(new Double(torIndU_1)).isNaN())opAREA.appendText(""+torIndU_1);
else opAREA.appendText("Not Available");
//Unspecifieds_Gen2
opAREA.appendText(", Gen2s of Unspecified = ");
if(!(new Double(torIndU_2)).isNaN())opAREA.appendText(""+torIndU_2);
else opAREA.appendText("Not Available");
//Unspecifieds_Gen3
opAREA.appendText(", Gen3s of Unspecified = ");
if(!(new Double(torIndU_3)).isNaN())opAREA.appendText(""+torIndU_3);
else opAREA.appendText("Not Available");

////////////////////
//Areas
////////////////////

opAREA.appendText("\nArea of Optic Disk = ");
if(contour1.area(px1_per_mm_x,px1_per_mm_y) != -1.0)
opAREA.appendText(""+contour1.area(px1_per_mm_x,px1_per_mm_y));
else opAREA.appendText("Not Available");

```

```

        opAREA.appendText("\nArea of Excavation = ");
        if(contour2.area(px1_per_mm_x,px1_per_mm_y) != -1.0)
opAREA.appendText(""+contour2.area(px1_per_mm_x,px1_per_mm_y));
        else opAREA.appendText("Not Available");

        opAREA.appendText("\nArea of Crescent = ");
        if(contour3.area(px1_per_mm_x,px1_per_mm_y) != -1.0)
opAREA.appendText(""+contour3.area(px1_per_mm_x,px1_per_mm_y));
        else opAREA.appendText("Not Available");

        opAREA.appendText("\nArea of Rim = ");
        if((contour1.area(px1_per_mm_x,px1_per_mm_y) != -
1.0)&&(contour2.area(px1_per_mm_x,px1_per_mm_y) != -1.0))
            opAREA.appendText(""+ (contour1.area(px1_per_mm_x,px1_per_mm_y)-
contour2.area(px1_per_mm_x,px1_per_mm_y)));
        else opAREA.appendText("Not Available");
    }

    //information
    else if(e.target==opInfo)
    {
        opAREA.appendText("\n-----");
        //CNT1
        opAREA.appendText("\n#Pts for "+modeNames[1]+" = "+contour1.nPoints());
        opAREA.appendText(" {");
        for(int i=0;i<contour1.nPoints()-1;i++)
        {
            opAREA.appendText(""+(contour1.getPoint(i)).x+", "+(contour1.getPoint(i)).y+"");
            if(i<contour1.nPoints()-1)opAREA.appendText(",");
        }
        opAREA.appendText("}");

        //CNT2
        opAREA.appendText("\n#Pts for "+modeNames[2]+" = "+contour2.nPoints());
        opAREA.appendText(" {");
        for(int i=0;i<contour2.nPoints()-1;i++)
        {
            opAREA.appendText(""+(contour2.getPoint(i)).x+", "+(contour2.getPoint(i)).y+"");
            if(i<contour2.nPoints()-1)opAREA.appendText(",");
        }
        opAREA.appendText("}");

        //CNT3
        opAREA.appendText("\n#Pts for "+modeNames[3]+" = "+contour3.nPoints());
        opAREA.appendText(" {");
        for(int i=0;i<contour3.nPoints()-1;i++)
        {
            opAREA.appendText(""+(contour3.getPoint(i)).x+", "+(contour3.getPoint(i)).y+"");
            if(i<contour3.nPoints()-1)opAREA.appendText(",");
        }
        opAREA.appendText("}");

        //XPNT
        if(xPnt.x!=-1)opAREA.appendText("\n "+modeNames[4]+": (" +xPnt.x+", "+xPnt.y+"");
        else opAREA.appendText("\n "+modeNames[4]+" Not Available");

        //Inner & outer circles
        opAREA.appendText("\nPixels Per millimeter [X] =" +px1_per_mm_x);
        opAREA.appendText("\nPixels Per millimeter [Y] =" +px1_per_mm_y);
        opAREA.appendText("\nInner Circle Diameter = " + mmDin + " mm = "+px1Din_x + " xPixels
= "+px1Din_y + " yPixels");
        opAREA.appendText("\nOuter Circle Diameter = " + mmDut + " mm = "+px1Dut_x + " xPixels
= "+px1Dut_y + " yPixels");

        //Vessels
        opAREA.appendText("\n#" +modeNames[5]+" = "+vessels.size());
        //Each vessel
        for(int i=0;i<vessels.size();i++)
        {
            opAREA.appendText("\n Vessel "+i+": #Pts =
"+((Vessel)vessels.elementAt(i)).nPoints()+
",
"+vesselChoice.getItem(((Vessel)vessels.elementAt(i)).getType()));

            double
t1_mm=((Vessel)vessels.elementAt(i)).getTotalLength(mmDut,xPnt,px1_per_mm_x,px1_per_mm_y);
            double
l1_mm=((Vessel)vessels.elementAt(i)).getLineLength(mmDut,xPnt,px1_per_mm_x,px1_per_mm_y);
            opAREA.appendText(". Total Length = "+t1_mm+" mm");
            opAREA.appendText(". Line Length = "+l1_mm+" mm");
            opAREA.appendText("\n
"+((Vessel)vessels.elementAt(i)).getCoors());
        }

        //Generatoinl
        opAREA.appendText("\n#" +modeNames[6]+" = "+genls.size());
        //Each generationl
        for(int i=0;i<genls.size();i++)
        {
            opAREA.appendText("\n Generationl Vessel "+i+": #Pts =
"+((Vessel)genls.elementAt(i)).nPoints()+
", "+genlChoice.getItem(((Vessel)genls.elementAt(i)).getType()));

            double
t1_mm=((Vessel)genls.elementAt(i)).getTotalLength(mmDut,xPnt,px1_per_mm_x,px1_per_mm_y);
            double
l1_mm=((Vessel)genls.elementAt(i)).getLineLength(mmDut,xPnt,px1_per_mm_x,px1_per_mm_y);

```

```

        opAREA.appendText(". Total Length = "+t1_mm+" mm");
        opAREA.appendText(". Line Length = "+l1_mm+" mm");
        opAREA.appendText("\n          "+((Vessel)gen1s.elementAt(i)).getCoors());
    }

    //Generatoin2
    opAREA.appendText("\n#"+modeNames[7]+" = "+gen2s.size());
    //Each generation2
    for(int i=0;i<gen2s.size();i++)
    {
        opAREA.appendText("\n  Generation2 Vessel "+i+": #Pts =
"+((Vessel)gen2s.elementAt(i)).nPoints()+
        ", "+gen2Choice.getItem(((Vessel)gen2s.elementAt(i)).getType()));

        double
        t1_mm=((Vessel)gen2s.elementAt(i)).getTotalLength(mmDut,xPnt,pxl_per_mm_x,pxl_per_mm_y);
        double
        l1_mm=((Vessel)gen2s.elementAt(i)).getLineLength(mmDut,xPnt,pxl_per_mm_x,pxl_per_mm_y);

        opAREA.appendText(". Total Length = "+t1_mm+" mm");
        opAREA.appendText(". Line Length = "+l1_mm+" mm");
        opAREA.appendText("\n          "+((Vessel)gen2s.elementAt(i)).getCoors());
    }

    //Generatoin3
    opAREA.appendText("\n#"+modeNames[8]+" = "+gen3s.size());
    //Each generation3
    for(int i=0;i<gen3s.size();i++)
    {
        opAREA.appendText("\n  Generation3 Vessel "+i+": #Pts =
"+((Vessel)gen3s.elementAt(i)).nPoints()+
        ", "+gen3Choice.getItem(((Vessel)gen3s.elementAt(i)).getType()));

        double
        t1_mm=((Vessel)gen3s.elementAt(i)).getTotalLength(mmDut,xPnt,pxl_per_mm_x,pxl_per_mm_y);
        double
        l1_mm=((Vessel)gen3s.elementAt(i)).getLineLength(mmDut,xPnt,pxl_per_mm_x,pxl_per_mm_y);

        opAREA.appendText(". Total Length = "+t1_mm+" mm");
        opAREA.appendText(". Line Length = "+l1_mm+" mm");
        opAREA.appendText("\n          "+((Vessel)gen3s.elementAt(i)).getCoors());
    }
}

// OUTPUT
// =====
else if(e.target==insertSep)
{
    opAREA.appendText("\n=====");
}
else if(e.target==clearOp)
{
    opAREA.setText("Output >>");
}
dspMode.setText("Current Mode = "+mode+" : "+modeNames[mode]);
return true;
}

// =====
// MOUSEDOWN
// =====
public boolean mouseDown(Event evt, int x, int y)
{
    if((x-ImgX<0)||(y-ImgY<0))return true;

    switch (mode)
    {
    case CNT1:
        contour1.addPoint(x-ImgX,y-ImgY);
        break;
    case CNT2:
        contour2.addPoint(x-ImgX,y-ImgY);
        break;
    case CNT3:
        contour3.addPoint(x-ImgX,y-ImgY);
        break;
    case XPNT:
        Point tmp=new Point(x-ImgX,y-ImgY);
        if(!(tmp.x<0 || tmp.y<0))xPnt=new Point(x-ImgX,y-ImgY);
        break;
    case VSSL:
        ((Vessel)vessels.lastElement()).addPoint(x-ImgX,y-ImgY);
        break;
    case GEN1:
        ((Vessel)gen1s.lastElement()).addPoint(x-ImgX,y-ImgY);
        break;
    case GEN2:
        ((Vessel)gen2s.lastElement()).addPoint(x-ImgX,y-ImgY);
        break;
    case GEN3:
        ((Vessel)gen3s.lastElement()).addPoint(x-ImgX,y-ImgY);
        break;
    }
    repaint();
    return true;
}

// =====
// UPDATE
// =====
/*
public void update(Graphics g)
{paint(g);}

```

```

*/
////////////////////////////////////
// PAINT
////////////////////////////////////
public void paint(Graphics gr)
{
    Graphics g = imgp.getGraphics();
    ImgX=imagePanel.location().x;
    ImgY=imagePanel.location().y;

    if(Img!=null)
    {
        if(!g.drawImage(Img,0,0,imgp.bounds().width,imgp.bounds().height,Color.white,this))
            display.setText("Loading image...");
        else
        {
            if(dspImgMsg)
            {
                display.setText("Image updated");
                dspImgMsg=false;
            }
        }
    }

    //CONTOUR 1
    g.setColor(CLR_CNT1);
    contour1.draw(g,true);

    //CONTOUR 2
    g.setColor(CLR_CNT2);
    contour2.draw(g,true);

    //CONTOUR 3
    g.setColor(CLR_CNT3);
    contour3.draw(g,true);

    //XPOINT & INNER & OUTER CIRCLES
    if(!(xPnt.x<0 || xPnt.y<0))
    {
        g.setColor(CLR_XPNT);
        g.drawLine(xPnt.x-5,xPnt.y-5,xPnt.x+5,xPnt.y+5);
        g.drawLine(xPnt.x-5,xPnt.y+5,xPnt.x+5,xPnt.y-5);

        g.setColor(CLR_CRCLin);
        g.drawOval(xPnt.x-pxlDin_x/2,xPnt.y-pxlDin_y/2,pxlDin_x,pxlDin_y);

        g.setColor(CLR_CRCLut);
        g.drawOval(xPnt.x-pxlDut_x/2,xPnt.y-pxlDut_y/2,pxlDut_x,pxlDut_y);
    }

    //VESSEL(S)
    int k=vessels.size();
    for(int i=0;i<k;i++)
    {
        if(((Vessel)vessels.elementAt(i)).getType()==0) g.setColor(CLR_VSSL_A);
        else if(((Vessel)vessels.elementAt(i)).getType()==1) g.setColor(CLR_VSSL_V);
        else if(((Vessel)vessels.elementAt(i)).getType()==2) g.setColor(CLR_VSSL_U);
        ((Vessel)vessels.elementAt(i)).draw(g,false);
    }

    //GEN1(S)
    k=gen1s.size();
    for(int i=0;i<k;i++)
    {
        if(((Vessel)gen1s.elementAt(i)).getType()==0) g.setColor(CLR_GEN1_A);
        else if(((Vessel)gen1s.elementAt(i)).getType()==1) g.setColor(CLR_GEN1_V);
        else if(((Vessel)gen1s.elementAt(i)).getType()==2) g.setColor(CLR_GEN1_U);
        ((Vessel)gen1s.elementAt(i)).draw(g,false);
    }

    //GEN2(S)
    k=gen2s.size();
    for(int i=0;i<k;i++)
    {
        if(((Vessel)gen2s.elementAt(i)).getType()==0) g.setColor(CLR_GEN2_A);
        else if(((Vessel)gen2s.elementAt(i)).getType()==1) g.setColor(CLR_GEN2_V);
        else if(((Vessel)gen2s.elementAt(i)).getType()==2) g.setColor(CLR_GEN2_U);
        ((Vessel)gen2s.elementAt(i)).draw(g,false);
    }

    //GEN3(S)
    k=gen3s.size();
    for(int i=0;i<k;i++)
    {
        if(((Vessel)gen3s.elementAt(i)).getType()==0) g.setColor(CLR_GEN3_A);
        else if(((Vessel)gen3s.elementAt(i)).getType()==1) g.setColor(CLR_GEN3_V);
        else if(((Vessel)gen3s.elementAt(i)).getType()==2) g.setColor(CLR_GEN3_U);
        ((Vessel)gen3s.elementAt(i)).draw(g,false);
    }
}
////////////////////////////////////
// GET PARENT
////////////////////////////////////
/*
    . Gets the top level parent of a component.

```

```

    · @param initialComponent Any component within the same hierarchy as the applet.
*/
protected Frame getTopLevelParent(Component initialComponent)
{
    Component theComponent = initialComponent; while (theComponent.getParent() != null)
    {
        theComponent = theComponent.getParent();
    }
    if (theComponent instanceof Frame)
    {
        return (Frame)theComponent;
    }
    else
    {
        return null;
    }
}

////////////////////////////////////
// CONTOUR
////////////////////////////////////
class Contour
{
    private List listCnt;

    public Contour()
    {
        listCnt = new List();
    }
    public void addPoint(int x, int y)
    {
        listCnt.addItem(""+x);
        listCnt.addItem(""+y);
    }
    public void clear()
    {
        listCnt.clear();
    }
    public int nPoints()
    {
        return (listCnt.countItems())/2;
    }
    public Point getPoint(int i)
    {
        Point p = new Point(0,0);
        p.x=(new Integer(listCnt.getItem(i*2))).intValue();
        p.y=(new Integer(listCnt.getItem(i*2+1))).intValue();
        return p;
    }
    public void draw(Graphics g,boolean closedCnt)
    {
        int xi,yi,xf,yf;
        int c=listCnt.countItems();
        int dc=0;

        if(c==2)
        {
            g.fillOval(
                ((new Integer(listCnt.getItem(0))).intValue()-2,
                ((new Integer(listCnt.getItem(1))).intValue()-2,
                4,4);
        }
        else if(c>2)
        {
            if(!closedCnt){c-=2;dc=2;}
            g.fillOval(
                ((new Integer(listCnt.getItem(0))).intValue()-2,
                ((new Integer(listCnt.getItem(1))).intValue()-2,
                4,4);

            for(int i=0;i<=c-2;i+=2)
            {
                xi=(new Integer(listCnt.getItem(i))).intValue();
                yi=(new Integer(listCnt.getItem(i+1))).intValue();
                xf=(new Integer(listCnt.getItem((i+2)%c))).intValue();
                yf=(new Integer(listCnt.getItem((i+3)%c))).intValue();

                g.drawLine(xi,yi,xf,yf);
            }
        }
    }
    public double area(double pxl_per_mm_x, double pxl_per_mm_y)
    {
        int xi,yi,xf,yf;
        double xi_mm,yi_mm,xf_mm,yf_mm;
        int c=listCnt.countItems();
        double area=0.0;

        if(c>2)//more than one point (x1,y1)...(x1,y1) l=2,3,...
        {
            for(int i=0;i<=c-2;i+=2)
            {
                xi=(new Integer(listCnt.getItem(i))).intValue();
                yi=(new Integer(listCnt.getItem(i+1))).intValue();
                xf=(new Integer(listCnt.getItem((i+2)%c))).intValue();
                yf=(new Integer(listCnt.getItem((i+3)%c))).intValue();

                xi_mm=xi/pxl_per_mm_x;
                yi_mm=yi/pxl_per_mm_y;
            }
        }
    }
}

```



```

        xf_mm=xf/pxl_per_mm_x;
        yf_mm=yf/pxl_per_mm_y;

        area+=(xf_mm - xi_mm)*0.5*(yi_mm + yf_mm);
    }

    if(area<0)
        return -1.0*area;
    else
        return area;
}
else
{
    return -1.0;
}
}

}

////////////////////////////////////
// VESSEL
////////////////////////////////////
class Vessel
{
    private Polygon poly;
    private int type;
    private int width;
    private double totalLength;
    private double lineLength;
    final double PRCNT_ERR_MRG01=5.0; // 5.0% (the last point can be a little bit outside the circle)

    public Vessel()
    {
        poly = new Polygon();
    }

    public void addPoint(int x,int y)
    {
        poly.addPoint(x,y);
    }
    public void setType(int n){type=n;}
    public void setWidth(int n){width=n;}
    public int getType(){return type;}
    public int getWidth(){return width;}

    public void clear()
    {
        poly = new Polygon();
    }

    public int nPoints()
    {
        return poly.npoints;
    }

    public String getCoors()
    {
        String s=new String("");
        int c=poly.npoints;

        for(int i=0;i<=c-1;i++)
        {
            s=s.concat(""+poly.xpoints[i]+","+poly.ypoints[i]+"");
            if(i<c-1)s.concat(",");
        }
        s=s.concat("");
        return s;
    }

    public void draw(Graphics g,boolean closedCnt)
    {
        int xi,yi,xf,yf;
        int c=poly.npoints;
        int dc=0;
        Integer k;

        if(c==1)
            g.fillOval(poly.xpoints[0]-2,poly.ypoints[0]-2,4,4);
        else if(c>1)
        {
            g.fillOval(poly.xpoints[0]-2,poly.ypoints[0]-2,4,4);
            if(!closedCnt){c-=1;dc=1;}
            for(int i=0;i<=c-1;i++)
            {
                g.drawLine(
                    poly.xpoints[i],
                    poly.ypoints[i],
                    poly.xpoints[(i+1)%(c+dc)],
                    poly.ypoints[(i+1)%(c+dc)]);
            }
        }
    }

    public double getTotalLength(double mmDut, Point xPnt, double pxl_per_mm_x, double pxl_per_mm_y)
    {
        Point p1,p2;

```

```

int c=poly.npoints;
double totalLength=0.0;
//int last=getLastPntIndex(mmDut,xPnt,pxl_per_mm_x,pxl_per_mm_y);
if (c<2 || (xPnt.x<0 || xPnt.y<0) )
{
    totalLength=-1.0;
}
else
{
    for(int i=0;i<=c-2;i++)
    {
        p1=new Point(poly.xpoints[i] , poly.ypoints[i] );
        p2=new Point(poly.xpoints[i+1], poly.ypoints[i+1]);

        if( circleContainsPnt(mmDut, p1, xPnt, p1.x,p1.y,p2.x,p2.y) &&
            circleContainsPnt(mmDut, p2, xPnt, p1.x,p1.y,p2.x,p2.y))
        {
            segment seg = new segment(p1.x,p1.y,p2.x,p2.y);
            totalLength+=seg.length(pxl_per_mm_x,pxl_per_mm_y);
        }
    }
}
return totalLength;
}

public double getLineLength(double mmDut, Point xPnt, double p1_per_mm_x, double p1_per_mm_y)
{
    double lineLength;
    int c=poly.npoints;
    int last=getLastPntIndex(mmDut,xPnt,pxl_per_mm_x,pxl_per_mm_y);
    if((last==0 || c<2) || (xPnt.x<0 || xPnt.y<0))
    {
        lineLength=-1.0;
    }
    else
    {
        segment seg = new segment(poly.xpoints[0], poly.ypoints[0], poly.xpoints[last],
poly.ypoints[last]);
        lineLength=seg.length(pxl_per_mm_x,pxl_per_mm_y);
    }
    return lineLength;
}

private boolean circleContainsPnt(double mmD, Point c , Point p, double p1_per_mm_x, double
pxl_per_mm_y)
/*circleContainsPoint(diameter in mm, circle center in pixels, point in pixels, p1_per_mm_x,
pxl_per_mm_y)*/
{
    double dx=(p.x-c.x)/pxl_per_mm_x;
    double dy=(p.y-c.y)/pxl_per_mm_y;
    double r=(mmD/2.0)*(PRCNT_ERR_MRGNO1/100.0+1); //increase the radius a little bit
    if ((dx*dx+dy*dy) <= r*r ) return true;
    else return false;
}

private int getLastPntIndex(double mmDut, Point xPnt, double p1_per_mm_x, double p1_per_mm_y)
{
    int lastPntIndex = -1;
    int c=poly.npoints;
    for(int i=0;i<=c-1;i++)
    {
        segment seg=new segment(xPnt.x, xPnt.y, poly.xpoints[i], poly.ypoints[i]);
        double dd = seg.length(pxl_per_mm_x,pxl_per_mm_y);
        if(dd <= (PRCNT_ERR_MRGNO1/100.0+1)*(mmDut/2.0) ) lastPntIndex = i;
    }
    return lastPntIndex;
}
}

////////////////////////////////////
// MY DIALOG
////////////////////////////////////
class myDialog extends Dialog
{
    Button ok=new Button("OK");

    public myDialog(Frame parent, String title, String msg1, String msg2)
    {
        super(parent,true);
        setBackground(Color.white);
        setResizable(false);

        Panel p1=new Panel();
        Panel p2=new Panel();

        p1.add(new Label(msg1, Label.CENTER));
        p1.add(new Label(msg2, Label.CENTER));
        p2.add(ok);

        add("Center",p1);
        add("South",p2);

        reshape(20,20,500,150);
    }

    public boolean action(Event e, Object o)
    {
        if(e.target==ok) dispose();
        return true;
    }
}

```

```

}
////////////////////////////////////
// SEGMENT
////////////////////////////////////
class segment
{
    private Point p1;
    private Point p2;

    public segment(Point pnt1, Point pnt2)
    {
        p1=pnt1;
        p2=pnt2;
    }
    public segment(int x1, int y1,int x2, int y2)
    {
        p1=new Point(x1,y1);
        p2=new Point(x2,y2);
    }

    public double length(double px1_per_mm_x, double px1_per_mm_y)
    {
        double dx=(p1.x-p2.x)/px1_per_mm_x;
        double dy=(p1.y-p2.y)/px1_per_mm_y;

        return java.lang.Math.sqrt( dx*dx+ dy*dy );
    }
    public double length()
    {
        return length(1.0,1.0);
    }
}

```

Help Information

The following is the help information as presented on the project web-page and can be obtained by pressing the 'Help' button.



Digital Image Analysis of Fundus Photographs on the WWW

DONE BY...

**GHASSAN HAMARNEH
IMAGING & IMAGE ANALYSIS GROUP
DEPT. OF SIGNALS AND SYSTEMS
CHALMERS UNIVERSITY OF TECHNOLOGY
GOTHENBURG - SWEDEN**

IN COLLABORATION WITH...

**THE EYE CLINIC IN THE EASTERN HOSPITAL
GOTHENBURG - SWEDEN**



HELP

About and help	Output information	Setting options	Status information and online help
Loading images	Setting parameters	Loading information	Regions and measurements



About and Help

Done by Ghassan Hamarneh
Imaging && Image Analysis Group
Department of Signals && Systems
Gothenburg - Sweden

Press the about button to see about information
Press the help button to view this help page

[Top of Page](#)

Loading Images

<input type="text" value="http://www.s2.chalmers.se/images/s2.c"/>	<input type="text" value="img.gif"/>	<input type="text" value="Sample Image 1"/>
<input type="button" value="View URL"/>	<input type="button" value="View Local"/>	<input type="button" value="View Choice"/>

Type an image URL then press View URL
Type the name of a local image file then press View Local
Choose one of the sample images and press View Choice

[Top of Page](#)

Output Information

Output >>	
<input type="button" value="Clear Output Area"/>	<input type="button" value="Insert Seperator"/>
<input type="button" value="Calculate && Generate Output"/>	<input type="button" value="Generate Information"/>

Press Clear Output Area to clear the output area
Press Insert Seperator to insert a seperator in the output area
Press Calculate & Generate Output to calculate measurements and display results
Press Generate Information to display information about the progress of measurements

[Top of Page](#)

Setting Parameters

Inner Circle Diameter (mm)	<input type="text" value="1.0"/>	<input type="button" value="Update"/>
Outer Circle Diameter (mm)	<input type="text" value="3.0"/>	<input type="button" value="Update"/>
X Pixels Per mm	<input type="text" value="50"/>	<input type="button" value="Update"/>
Y Pixels Per mm	<input type="text" value="50"/>	<input type="button" value="Update"/>
Calibration Factor	<input type="text" value="1.0"/>	<input type="button" value="Update"/>

Type the inner circle diameter and press return or update

Type the output circle diameter and press return or update

Type the number of pixels per mm in the X direction and press return or update

Type the number of pixels per mm in the Y direction and press return or update

Type the Calibration Factor and press return or update

[Top of Page](#)

Setting Options

- Auto [Start Excavation] on [End Optic Disk]
- Auto [Start Crescent] on [End Excavation]
- Auto [Set Exit Point] on [End Crescent]

Check the first box if you want to automatically start specifying the Excavation region once you finished the Optic Disk

Check the second box if you want to automatically start specifying the Crescent region once you finished the Excavation

Check the third box if you want to automatically start specifying the Exit Point once you finished the Crescent

[Top of Page](#)

Loading Information

Right Eye	<input type="text" value="Male"/>	Age	<input type="text"/>
Last Name	<input type="text"/>	First Name	<input type="text"/>
Gest. Weeks	<input type="text"/>	Diagnosis	<input type="text"/>
VA-R	<input type="text"/>	VA-L	<input type="text"/>
Ref-R	<input type="text"/>	Ref-L	<input type="text"/>
Camera	<input type="text"/>	Magnification	<input type="text"/>
Email	<input type="text"/>	Additional	<input type="text"/>
File Name	<input type="text" value="info.dat"/>	Case	<input type="text"/>
ID Number	<input type="text"/>	Date	<input type="text" value="9801116"/>
Operator	<input type="text"/>	<input type="button" value="Load"/>	<input type="button" value="Clear"/>

Type in the file name that contains the user information in File Name

Press the Load button to load the user information

Press Clear to reset all the controls

[Top of Page](#)

Status Information and Online Help

Click the points for one of the VSSL (Vessels) then End

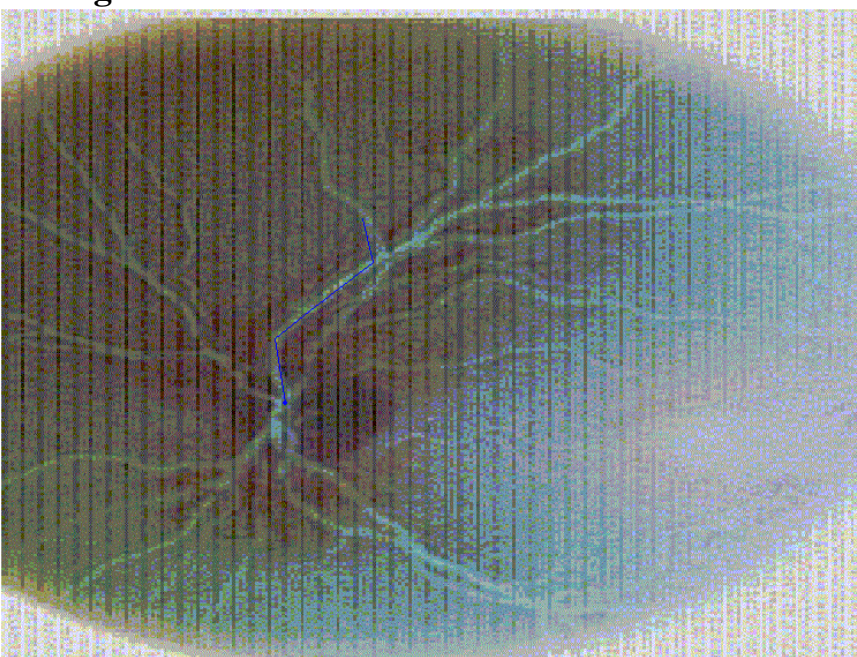
Current Mode = 5 : VSSL (Vessels)

**On the left side you see information about the mode you are currently in
In the middle you see on-line help and tips that guide you while using the applet**

[Top of Page](#)

Regions and measurements

Start Optic Disk	End Optic Disk
Start Excavation	End Excavation
Start Crescent	End Crescent
Set Exit Point	End Exit Point
Start Vessel	Vein
end Vessel	Delete Last Vessel
Start Gen1	Artery
End Gen1	Delete Last Gen1
Start Gen2	Artery
End Gen2	Delete Last Gen2
Start Gen3	Artery
End Gen3	Delete Last Gen3



**Press Start 'some region' and then start specifying the region
Press End 'som region' when you have finished specifying the region
Specify the type of vessel before or while you are marking the vessel
Use Delete Last Vessel/Gen1/2/3 to delete the last vessel/Gen1/2/3 specified**

[Top of Pag](#)

References & Related WWW links (URLs)

References

- Morphometry of the optic nerve and retinal vessels in children by computer-assisted image analysis of fundus photographs. Kerstin Strömberg, Ann Hellström, Tomas Gustavsson. Graefes Arch Clin Exp Ophthalmol (1995) 233:150-153. Springer-Verlag 1995.
- Programming.java, An Introduction to Programming Using Java, Decker & Hirshfield, PWS Publishing 1998

URLs

- The Pediatric Ophthalmology- Digital Image Analysis of Fundus Photographs on the WWW
<http://www.oftalmologi.gu.se/fundus/index.html>
- The homepage of the Java course
http://www.s2.chalmers.se/~jessi/phd/java_course/index.htm
- Demonstration Page at Informatik.gu.se
<http://internet.informatik.gu.se/~jessi/>
- Help file for using the Applet
http://www.s2.chalmers.se/~jessi/phd/java_course/project/help.htm
- Java Links (see table)

http://www.davecentral.com/java.html	DaveCentral Java
http://java.sun.com/jdc/	Developer Connection
http://www.developer.com/directories/pages/dir.java.html	Gamelan
http://www.jars.com	JARS.COM
http://www.sun.com/java/	Java Computing
http://sunsite.utk.edu/winners_circle/	Java Cup
http://java.sun.com	Java Home Page
http://www.ibm.com/java	Java IBM
http://www.sys-con.com/java	Java Journal
http://java.sun.com/products/java-media/index.html	Java Media
http://java.sun.com:81/support/moresupport.html	Java Tech Resources
http://www.jsworld.com/	JavaScript World
http://java.sun.com/docs/books/tutorial	Java Tutorial
http://www.javaworld.com	JavaWorld
http://www.microsoft.com/java/default.htm	Microsoft - Java
http://www.microsoft.com/visualj/	Microsoft Visual J++
http://www.sun.com/	Sun Microsystems
http://www.javalobby.org	The Java Lobby
http://javaboutique.internet.com	The Java(TM) Boutique

- Java and Imaging Links

http://java.sun.com/products/java-media/jai/	Sun: Java Advanced Imaging API
http://java.sun.com:80/products/java-media/index.html	Sun: Java Media
http://ocelot.cat.syr.edu/~shrideep/furm/	Image Editor
http://www.icg.tu-graz.ac.at/~kienast/java/Java.html	Gert Kienast: Image processing with JAVA
http://cobb.ee.psu.edu/users/greg/	Greg Simon & Manu Chatterjee JAVA Image Processing
http://www.npac.syr.edu/projects/vishuman/VisibleHuman.html	The NPAC Visible Human Viewer
http://www.nlm.nih.gov/research/visible	The Visible Human Project
http://rsb.info.nih.gov/ij	ImageJ
http://www.emory.edu/CRL/abb/	Andrew Barclay's Medical Imaging Pages
http://www.persci.com/~schulze/	Mark Schulze
http://www.davecentral.com/javaimgp.html	DaveCentral Java - Image Processing
http://www.eeng.dcu.ie/~whelanp/jvision/jvision_help.html	JVision (Image Analysis Development Environment)

http://rsb.info.nih.gov/IJ/default.html	ImageJ
http://www.uwm.edu/~linbo/jip/jip.html	JIP Java Image Processing

Future

Here are some suggestions on possible future work:

- More widths & types of the lines used for drawing vessels depending on their type, for example 1st generation vessels should be thicker than 2nd generation.
- Each client should have a web-page. That page should be private to the client, and all the results - including numbers, graphs and tables- should be put on this page.
- The client should be able to browse for the image file and submit it to the server from within a web-form, at the server there should be a CGI script to deal with this.
- As much of the functionality should be automated. For example, the saving and loading of images, building private pages for the clients, putting the results on those pages.
- Building Databases and interacting with the database. The database should include information on clients, images, image-information, users allowed to access the private pages and their passwords, etc.
- Calibration of the system. The current program accounts for calibration but the only thing missing is the values for Pixels/mm in both the spatial directions.

Glossary

CGI

common gateway interface

The CGI standard lays down the rules for running external programs in a Web HTTP server. External programs are called *gateways* because they open up an outside world of information to the server.

client

The customer side of a client/server setup. To confuse matters, when you log on to a server, the word *client* can refer to you, to your computer, or to the software running on your computer. For example, to download something from an ftp site, you use ftp client software.

GUI

graphical user interface

A graphical user interface lets users interact with their computer via icons and a pointer instead of by typing in text at a command line. Popular GUIs, such as Sun Microsystem's OpenWindows, Microsoft's Windows, and Apple's Mac OS, have freed many users from the command-line interfaces like MS-DOS and Unix.

HTML

Hypertext Markup Language

As its name suggests, HTML is a collection of formatting commands that create hypertext documents--Web pages, to be exact. When you point your Web browser to a URL, the browser interprets the HTML commands embedded in the page and uses them to format the page's text and graphic elements. HTML commands cover many types of text formatting (**bold** and *italic* text, lists, headline fonts in various sizes, and so on), and also have the ability to include graphics and other nontext elements.

Development and maintenance of [HTML standards](#) is coordinated by the [World Wide Web Consortium](#)

Java

Sun Microsystems' Java is a programming language for adding animation and other action to Web sites. The small applications (called applets) that Java creates can play back on any graphical system that's Web-ready, but your Web browser has to be [Java-capable](#) for you to see it. According to Sun's description, Java is a "simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high-performance, multithreaded, dynamic, buzzword-compliant, general-purpose programming language." And Sun should know

Perl

practical extraction and report language

The programming language of choice for writing Web server applications, [Perl](#) is used for creating interactive forms and a slew of other CGI programs. This free-licensed language comes in versions for Windows NT, Novell NetWare, and Unix. Perl scripts are available free of charge all over the Internet.

server

The business end of a client/server setup, a server is usually a computer that provides the information, files, Web pages, and other services to the client that logs on to it. (The word *server* is also used to describe the software and operating system designed to run server hardware.) The client/server setup is analogous to a restaurant with waiters and customers. Some Internet servers take this analogy to extremes and become inattentive, or even refuse to serve you.

URL

uniform resource locator - universal resource locator

URLs are the Internet equivalent of addresses. How do they work? Like other types of addresses, they move from the general to the specific (from zip code to recipient, so to speak). Take this URL, for example:

<http://www.cnet.com/Resources/index.html>

First you have the protocol:

http:/

then the server address or domain:

/www.cnet.com

and finally the directory:

/Resources/

in which the file index.html resides.

Two debates rage: first, does the *U* stand for *uniform* or *universal*? *Universal* was the original definition of choice but was deemed by most to be too ambitious, and the more frequently used *uniform* was instated by the now-defunct URI Working Group.

Second, is URL pronounced "you are ell," or does it rhyme with hurl? Both pronunciations are widely used