

©Copyright 2016

Ya-Shin Chen



# Graphical User Interface for Cleft Severity Analysis

Ya-Shin Chen

A thesis  
submitted in partial fulfillment of the  
requirements for the degree of

Master of Science

University of Washington

2016

Committee:

Linda G. Shapiro, Chair

Eve A. Riskin

Program Authorized to Offer Degree:  
Electrical Engineering

University of Washington

**Abstract**

Graphic User Interface for Cleft Severity Analysis

Ya-Shin Chen

Chair of the Supervisory Committee:

Dr. Linda G. Shapiro

Department of Computer Science and Engineering

3D FAST (3D Facial Analysis System and Tool) is a graphical user interface (GUI) designed for analyzing craniofacial symmetry, particularly for cleft severity. The goal is to enable an easy experience for computing the extent of symmetry in the face in batch mode, and examining and editing individual images as well. The GUI was designed to have a user-friendly interface, in order to be straightforward to operate for users with little or no technological background. The decision of using common components, such as drop-down menus that are familiar to people, was made during early design phase. All of the components of the GUI were built with JAVA Swing and the VTK Library.

# Table of Contents

Chapter 1 INTRODUCTION.....	9
1.1 Motivation.....	9
1.2 Background.....	10
1.3 Goals .....	11
Chapter 2 Related Work.....	12
Chapter 3 Theoretical Framework .....	17
Chapter 4 System Implementation.....	23
4.1 Graphical User Interface (GUI) .....	23
4.1.1 Workspace .....	23
4.1.2 Progress Table.....	29
4.1.3 Project Tree.....	31
4.2 Analysis Modules Execution .....	31
4.3 Mesh Cropping.....	33
4.4 System Requirements.....	35
4.5 Instruction/ Sample Use Case .....	36
4.5.1 Add a New Project and Import Images.....	36
4.5.2 Analyze: Landmark and Midplane Detection .....	38
4.5.3 Analyze: Symmetry Scores.....	42
4.5.4 Analyze: Analyze All.....	46
4.5.5 Analyze: View Results.....	46
4.5.6 Analyze: Anthropometric Analysis.....	47
4.5.7 Add/ Edit/ Remove a landmark.....	49
4.5.8 Import/ Export a landmark file .....	49
4.5.9 Calculate Midplane by Manually Landmark Selection .....	50
4.5.10 Mesh Removal: Eraser.....	50
4.5.11 Mesh Removal: Rectangular Crop.....	51

Chapter 5 Future Work and Conclusions.....	52
5.1 Future Work .....	52
5.2 Conclusions.....	54

## Table of Figures

Figure 1. Khoros user interface.....	13
Figure 2. VisTrails user interface.....	13
Figure 3. NeatVision user interface .....	14
Figure 4. CranioGUI user interface .....	15
Figure 5. 3D FAST user interface.....	16
Figure 6. Drafted GUI for usability test.....	19
Figure 7. 3D FAST user interface.....	21
Figure 8. Image windows within the workspace .....	25
Figure 9. Three tabbed panes for (a) landmarks, (b) mid-facial plane and (c) symmetry scores .	26
Figure 10. Example of adding a landmark though ‘Add a landmark’ button.....	28
Figure 11. Mid-facial plane before and after changing the center point.....	28
Figure 12. Symmetry scores based on point-based and grid-based methods .....	29
Figure 13. Analysis progress table panel .....	30
Figure 14. Project tree.....	30
Figure 15. Module execution using Swingworker in Java.....	32
Figure 16. Mesh cropping: (a) eraser button and popup menu, (b) mesh removal demonstration. .....	33
Figure 17. 3D rectangular cropping.....	35
Figure 18. Creating a new project.....	36
Figure 19. Importing images to a project.....	37
Figure 20. Importing .ply file.....	37

Figure 21. Steps for landmark and mid-facial plane detection: (a) argument setting, (b) table panel before the analysis starts, (c) table panel after the analysis completed.....	40
Figure 22. Landmark data after landmark detection.....	41
Figure 23. Mid-facial plane data.....	42
Figure 24. Dialog for choosing a project before the symmetry score calculation .....	43
Figure 25. Table panel for symmetry scores calculation .....	43
Figure 26. Symmetry scores display .....	45
Figure 27. Steps to view the results of symmetry scores .....	46
Figure 28. Comparison of the symmetry scores .....	47
Figure 29. Steps to export CSV data.....	47
Figure 30. Dialog for exporting data.....	48
Figure 31. Demonstration for point distance data.....	48
Figure 32. Demonstration for symmetry scores data .....	49
Figure 33. Mid-facial plane evaluation by manual landmark selection.....	50
Figure 34. Mesh removal using Eraser: (a) the popup menu listing the options for eraser sizes. (b) the resulting mesh when using the eraser. ....	51
Figure 35. Raw data of texture image and 3D mesh.....	52
Figure 36. Procedure of the auto-cleaning module.....	53
Figure 37. Undo the auto-cleaning.....	54

# Chapter 1

## INTRODUCTION

### 1.1 Motivation

3D imaging software applications are widely used in healthcare. These applications assist doctors and medical professionals with long-term treatment and are helpful for medical research. This is also one of the goals of the GUI developed in this thesis. Though 3D imaging tools allow us to perform many tasks, such as examining and editing images, most of them do not come with analysis functions. The GUI we developed not only has the tools for basic image operations, but also combines algorithms that enable users to do specific analysis tasks for cleft severity.

The program for cleft severity analysis consists of many small algorithms. Without a graphical user interface (GUI), a user would have to run each of the algorithms manually. Moreover, it would be difficult to run for users who are not familiar with batch computing. Therefore, we aimed to design a GUI that offers a better experience: (1) any user can start his or her analysis procedure with easy visual interactions, (2) the analysis tasks are automatic without the need of complicated settings and (3) high integration of computer vision algorithms customized for cleft analysis produces fast and systematic results.

In order for all users to be able to use the GUI, the Java programming language is adopted, and thus the system can run on all platforms for which there exists a JVM (Java Virtual Machine). Also, the mesh-visual part was implemented with Java-wrapped VTK that provides smooth mesh-related display and analysis.

## 1.2 Background

3D FAST is a software application running computer vision algorithms for cleft severity analysis. The source code of the algorithm was developed by Jia Wu and Ezgi Mercan at the University of Washington Graphics and Imaging Lab.

The GUI was developed to enhance the connection among the algorithms and enable users to perform tasks effortlessly. The existing computing algorithms were created at different times and by different authors, so the GUI provides an integration.

To implement the GUI efficiently, the author followed Jakob Nielsen's model [1]:

- **Know the User:** In the initiation phase there are many aspects needed to be considered, such as the characteristics of the target users and the goals that these users want to achieve through this software application. Since this GUI is customized for cleft severity analysis, the most potential users are doctors and professionals who are interested in craniofacial symmetry. Thus, the doctors and researchers at Seattle Children's Hospital were involved in the design process.
- **Goal Setting:** What an application should contain and what users might expect from it are important questions to be answered. The function of each algorithm is very clear, but the way to show and run them has to be designed. The design goals for the GUI were set by discussions with Dr. Raymond Tse, who is a surgeon at Seattle Children's Hospital, and Prof. Linda Shapiro.
- **Guidelines:** Many guidelines to develop user interfaces have been developed for the HCI field in recent years. The author followed the guidelines from [1].
- **Prototyping:** A prototype GUI helps the developer to obtain more details of users' expectations. Before starting the actual coding, a scratch GUI was constructed for a

usability test. Then the feedback from users using the scratch GUI was collected to improve the design.

- Iterative Design: In the design phase, there were countless changes made based on the evaluation results from prototyping and feedback whenever a functionality was completed. Early changes were mostly to the wireframe of the GUI; once it was finalized, the remaining changes were about functionality.

### 1.3 Goals

Even though clefts have been studied for centuries, there was a lack of useful software applications to help medical researchers to perform a systematic analysis. Our goal is to develop a GUI that allows doctors and medical professionals to objectively and efficiently analyze cleft severity. Furthermore, they will be able to run the algorithms in batch mode without writing complex scripts, and evaluate facial symmetry of each patient.

## Chapter 2

### Related Work

The goal of this research is to build an interactive GUI that has a short learning curve. Different types of visual programming environments have been investigated, such as pipelining and dataflow models; however, it was found that the most familiar one to users is a system that uses popup buttons with a pull-down menu to issue commands. The systems that feature pipelining and dataflow will be discussed in the following sections and will be compared to the system featuring pull-down menus.

Many signal processing software applications provide a workflow interface that allows users to chain modules onto a workspace. The workflow user interface is flexible so that users are able to graphically create different sequences according to their needs. *Khoros* [2] and *VisTrails* [3] are two well-known examples of workflow systems for scientific visualization. The main difference between them is the layout of available modules. As shown in Figure 1 and Figure 2, *Khoros* has modules displayed as buttons at the top of the interface, while *VisTrails* enumerates modules as a list on the left side. *NeatVision* [4] is another visual-programming-based image analysis development environment for medical imaging applications. Similar to *VisTrails*, it has a list on the left side of the interface, but instead of a list of available modules, it is a hierarchical list of folders and modules as shown in Figure 3. Additionally, the workspace can contain many windows. For instance, one window may contain workflows of modules, while others may contain visualizations.

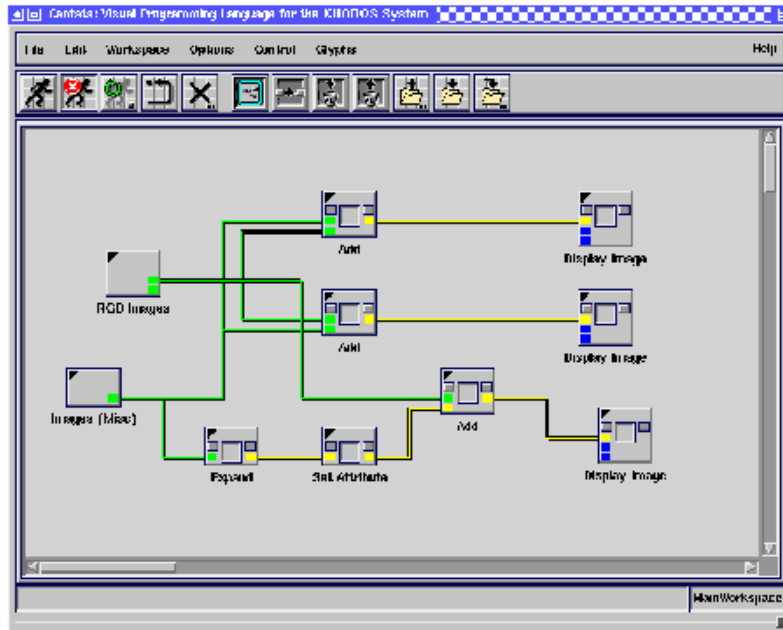


Figure 1. Khoros user interface

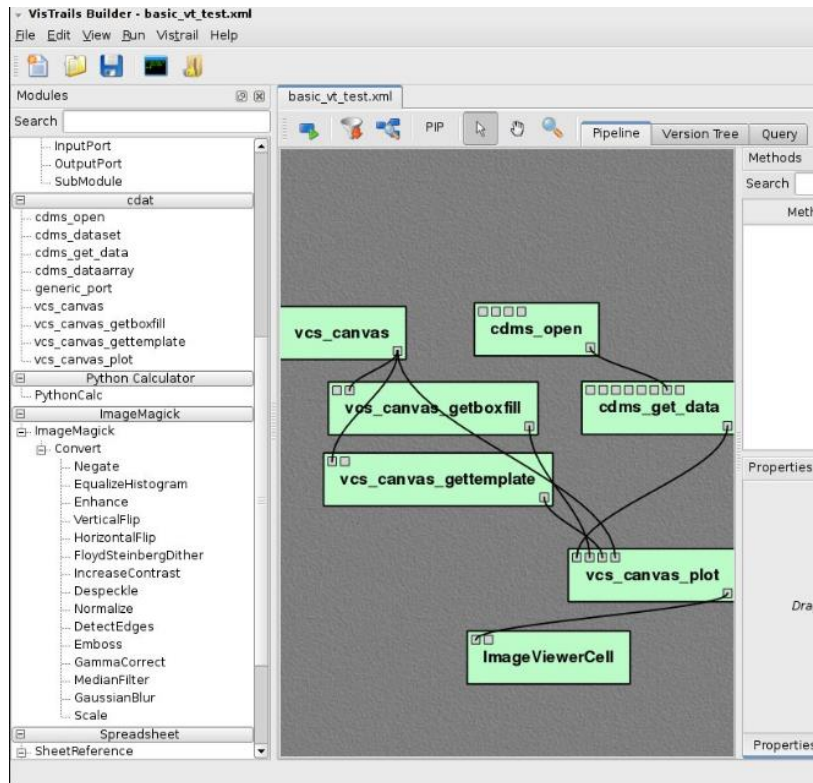


Figure 2. VisTrails user interface

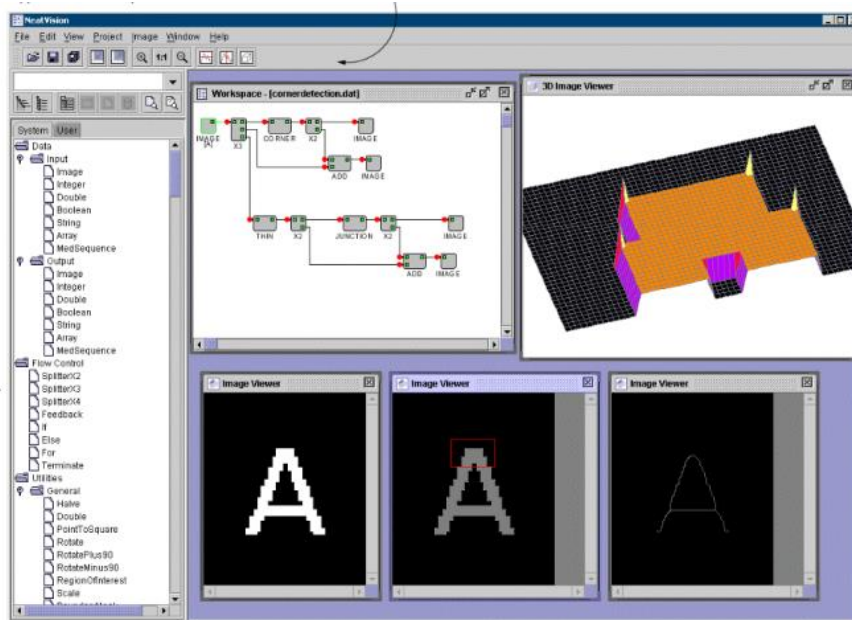


Figure 3. NeatVision user interface

CranioGUI [5] is a graphical user interface specifically designed for 3D craniofacial data analysis. It combines the advantages of the *Khoros* and *NeatVision* systems so its flexibility allows users to chain modules onto the workspace and view the analysis results displayed in windows within the workspace. However, it is incapable of batch mode analysis, and users had trouble setting the arguments of the modules according to the user's feedback. The CranioGUI interface is shown in Figure 4.

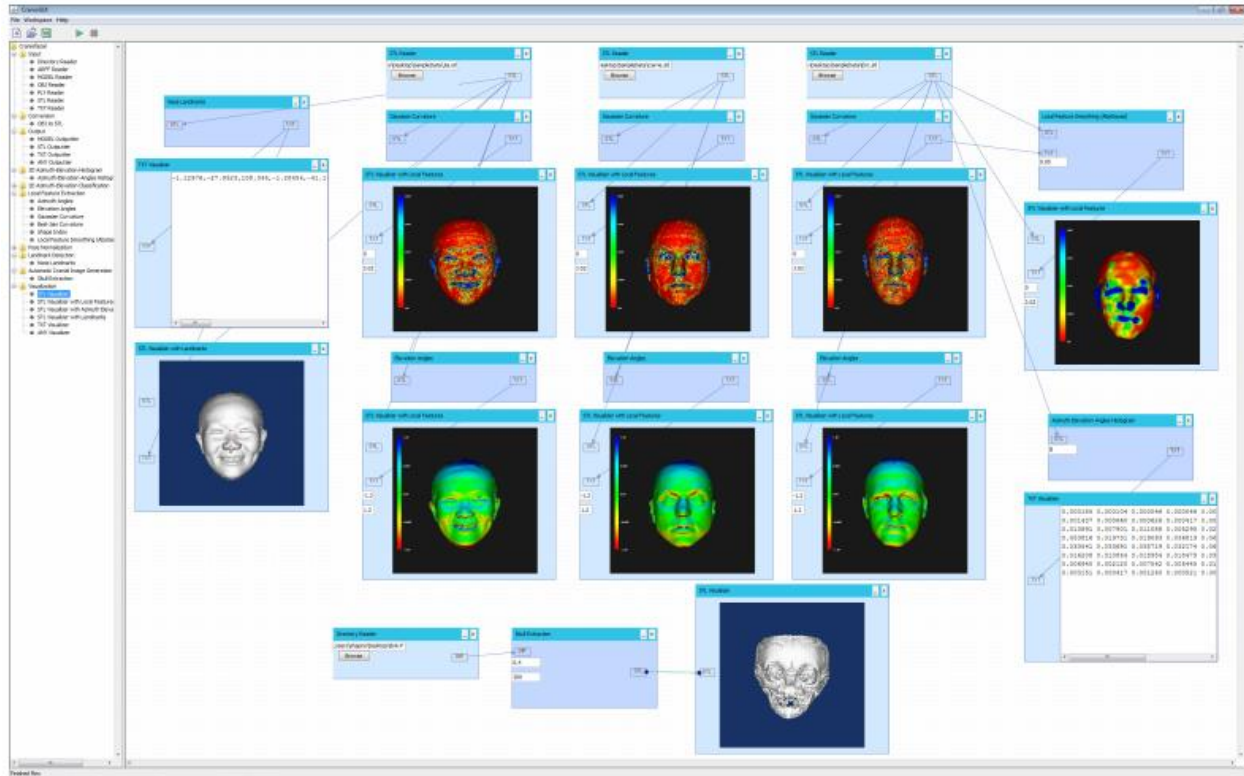


Figure 4. CranioGUI user interface

Despite the flexibility of workflow-type user interface, it might be hard to use for users who have little knowledge about the procedures of image and signal processing, since they must understand the relationship between inputs, outputs and modules in order to use the interface. To improve the usefulness of the interface, we adopted the software environment similar to *NeatVision*, but rather than chaining modules for analysis, buttons and pull-down menus are used to issues commands as shown in Figure 5. The main benefit of replacing the workflow interface with buttons and pull-down menus is that users can run the analysis modules without memorizing the functions and the relationships among them. Since the modules are hidden behind the GUI, users only need to know what analysis they want to perform. Thus, this

simplifies the use of the system. The practical implementation will be discussed in the next chapter.

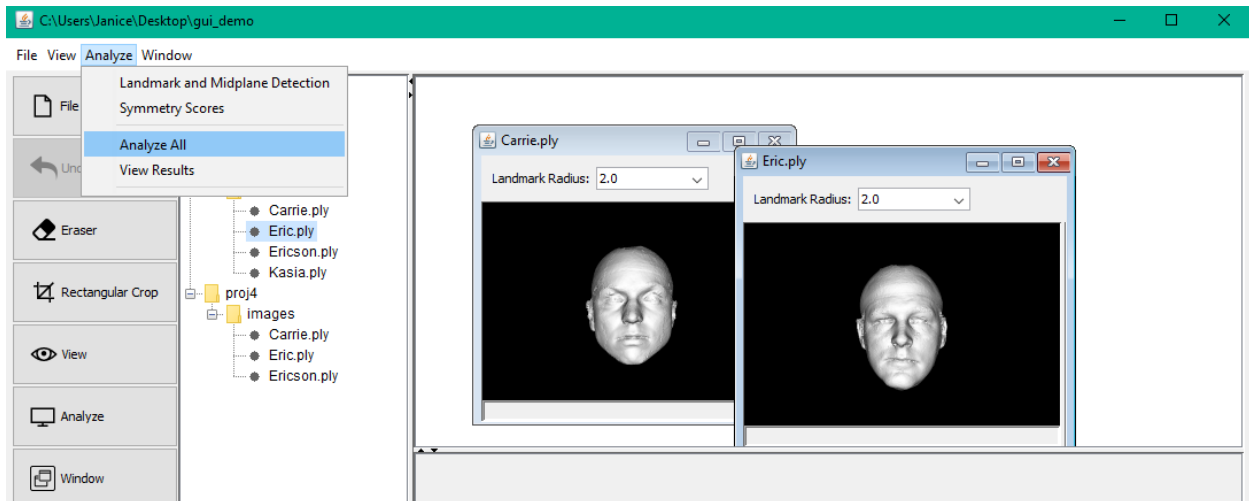
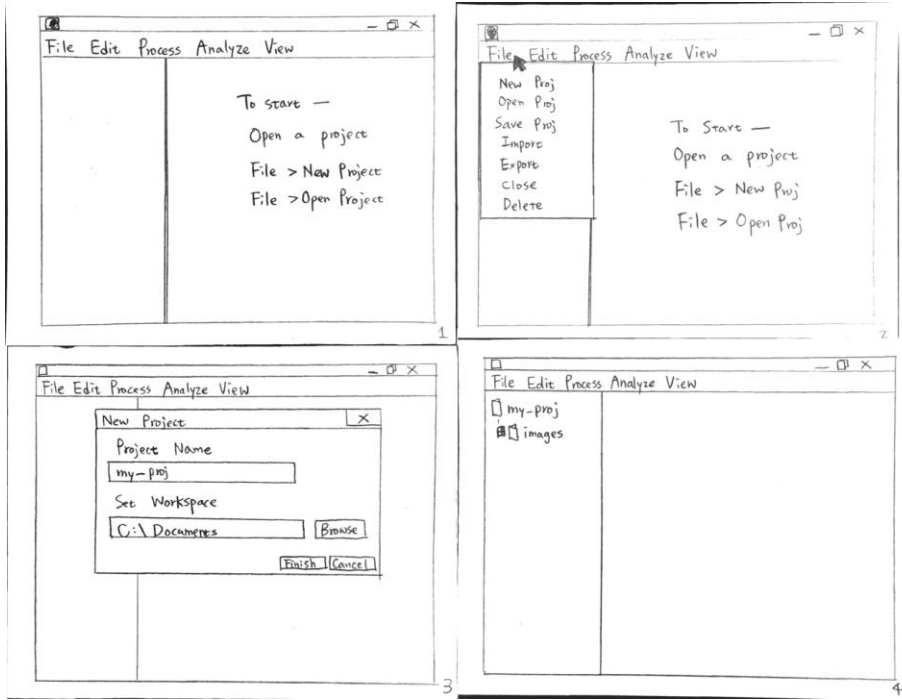


Figure 5. 3D FAST user interface

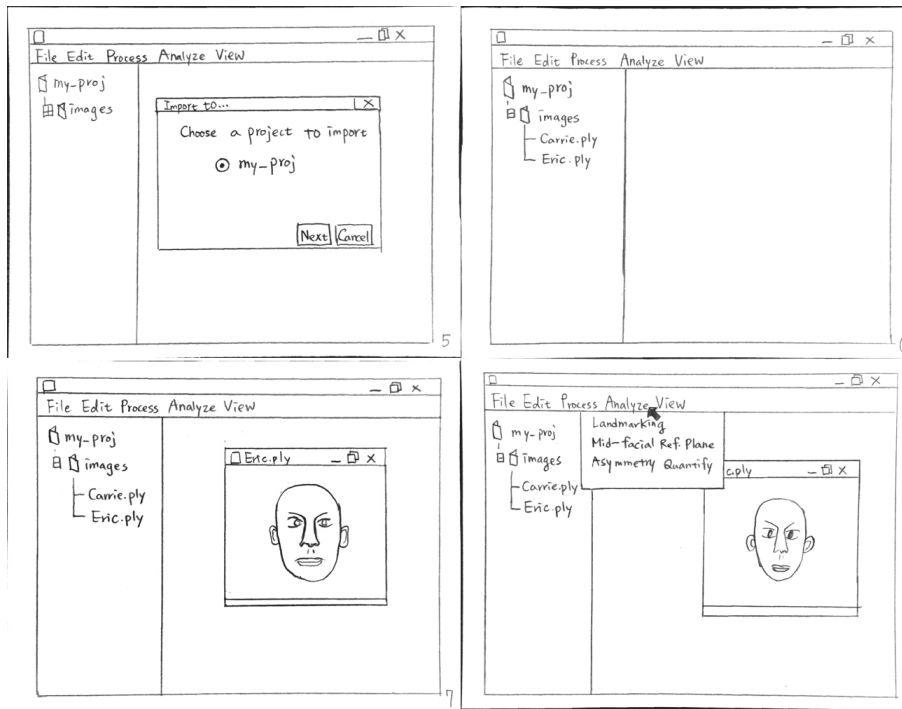
## Chapter 3

### Theoretical Framework

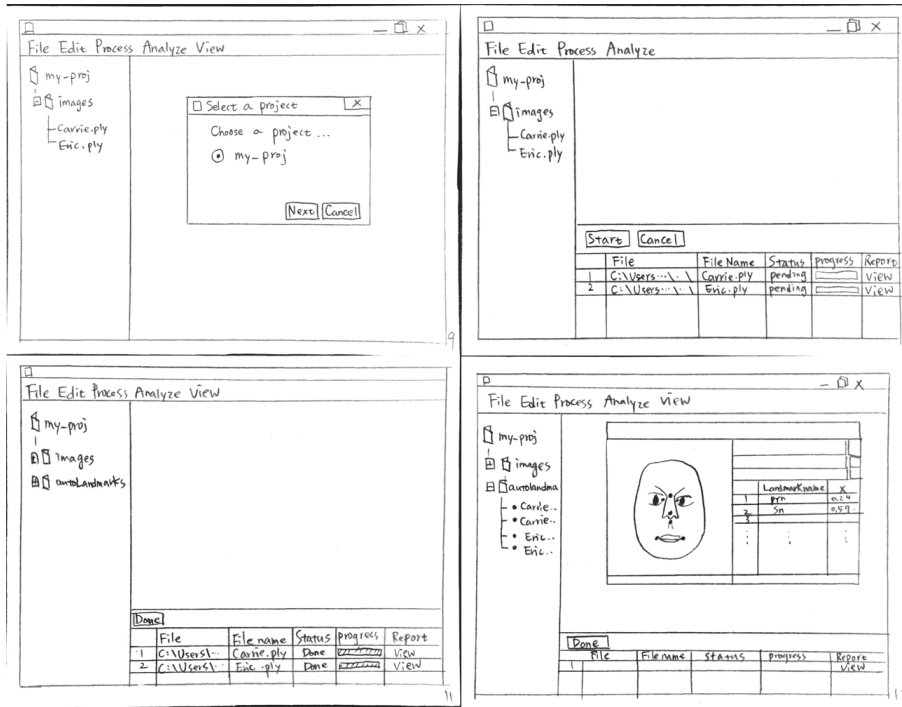
In order to offer an easier experience in the craniofacial symmetry analysis, we had discussions in the early design phase and did a usability test before starting the implementation. We invited the main user of the system, a surgeon at Seattle Children's Hospital, to experiment with the drafted user interface shown in Figure 6 (a) (b) and (c). Figure 6 (a) depicts the scenario of creating a new project; this was to test if the user could successfully create a new project with the designed interface. Figure 6 (b) is the scenario of importing and opening an image that was to test if the user could intuitively perform the task. The interface shown in Figure 6 (c) was a test for running an analysis. In short, the user was asked to perform three tasks: (1) creating a new project, (2) importing and opening an image in that project, and (3) running an analysis. The user's interaction with the GUI was observed and recorded for improving the design. After the user was successful in creating a new project, a hierarchical file tree was added to the interface. It was noted that the user expected to take actions from the tree nodes that represented image files. For example, he expected to import images by right-clicking on a tree node. Therefore, a popup menu was added to the file tree. This will be discussed in the next section. Also, we realized that popup buttons and pull-down menus are the most familiar user interface for users. Therefore, users will access data and issue commands through buttons and proceed to view results via internal windows contained in a workspace.



(a)



(b)



(c)

Figure 6. Drafted GUI for usability test

The following sections explain each component of the interface of 3D FAST. According to the functions, the interface can be divided into five regions, which are shown in Figure 7. The GUI is designed according to the following criteria:

### Workspace

Our design for the workspace is an extension of *NeatVision*. One workspace window in the application containing multiple internal windows for visualization organizes the interface. The difference is that 3D FAST does not have a workflow window.

The main purpose of 3D FAST is image processing, and thus image viewing is the most essential operation. The workspace works as a desktop that allows users to open images in the workspace

to perform examination and editing jobs. It has a convenient ‘drag and drop’ feature so that users are able to move images from the image tree nodes at the navigation panel. Furthermore, it is clearly separated from other components so that users will intuitively identify the main working region. Thus we decided to implement a workspace because it is easier to focus on.

## **Navigation Panel**

Working on multiple projects is possible for some users. The navigation panel contains a tree structure that lists users’ projects. By walking through the tree, users are able to manage projects, browse images in the projects, and open or delete images via a popup menu.

Recall that there is a navigation panel at the left side of CranioGUI allowing users to browse items in the hierarchical tree. 3D FAST uses a similar design, but it provides project folders instead of a tree of available modules. The benefit is that it is easier to understand and more intuitive for users with little technical background.

## **Toolbars**

3D FAST provides many functions for users to edit and analyze images, and those can be accessed through buttons arranged in a toolbar. The toolbar offers an organized view so that users can find any function they need to perform.

### *Top toolbar*

We place a toolbar at the top of the GUI as in conventional user interfaces for users who are more likely to search for functions there.

### *Side toolbar*

Another toolbar is placed at the left side of the GUI. The buttons in this toolbar are designed to be like shortcuts, which have larger buttons for better navigation.

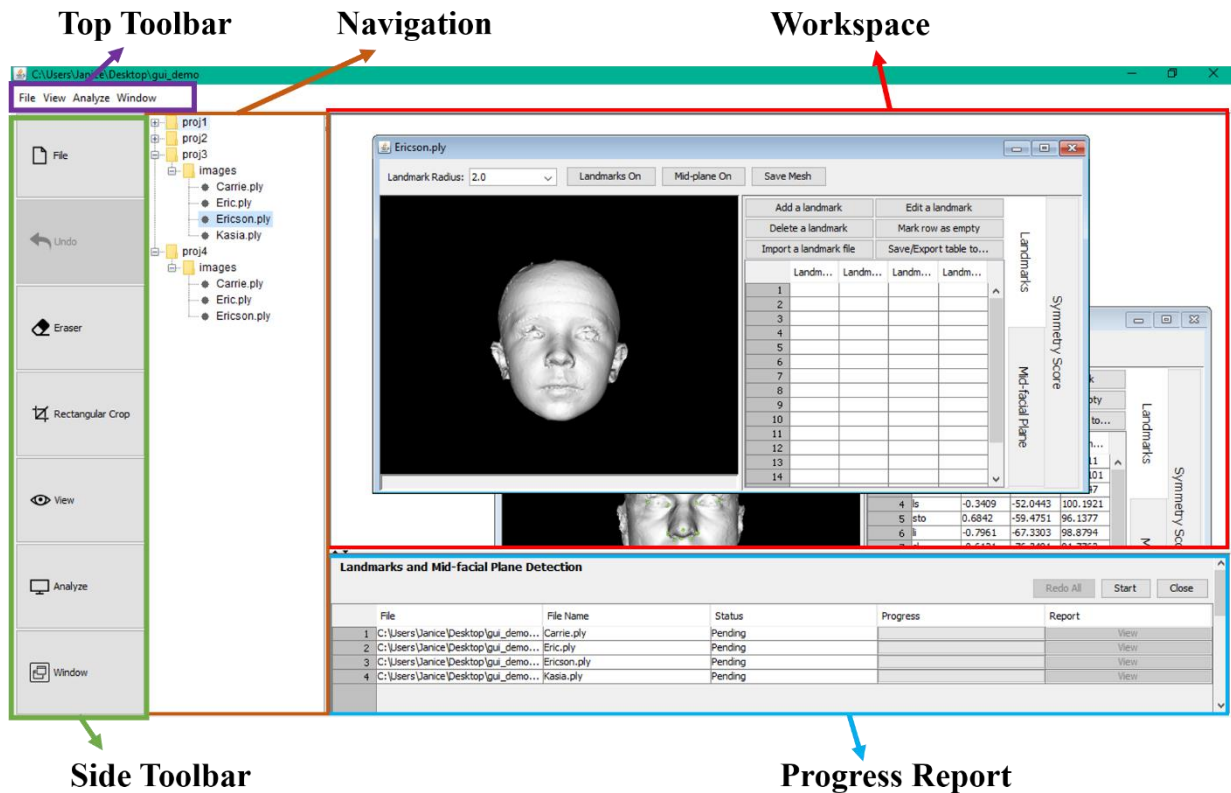


Figure 7. 3D FAST user interface

### Real-time Progress Report

Displaying the progress of an analysis is important, because it allows users to monitor the analysis process. Without an interface showing the analysis process, users have to wait until the analysis is completed to check the correctness of the result. An interface of running progress assists users in deciding whether to cancel the process due to a poor result. Furthermore, it also helps estimate the time it takes to finish the process. Thus, 3D FAST provides a panel showing the progress for reporting status during the analysis.

A table is added to this panel when users choose to do image analysis, and then the progress bar of each image displays the progress after starting the analysis. If there is any problem with the

analysis, the progress bar and status field in the table will show the reasons so that users know how to fix it.

## Summary

After conducting the literature review and exploring CranioGUI, we created a friendlier and more interactive system which has the following features:

- Single workspace: like *NeatVision* in that the workspace contains multiple windows, but 3D FAST does not adopt a workflow strategy.
- Popup button and pull-down menu: no workflow used in 3D FAST, instead, users issue commands via buttons and pull-down menus. This enables an easier experience, since users can perform analysis without knowing the input/output of modules.
- Project navigation: instead of a list of available modules in the reviewed literature, 3D FAST displays a hierarchical file tree that allows users to take different actions.
- Toolbar shortcuts: instead of small shortcut icons, large buttons with texts and icons allow users to easily access all of the functions and reduce the learning curve.
- Progress report: unlike the discussed literature and CranioGUI, which does not show any progress during the analysis, the status field and progress bar of the analysis table in 3D FAST reports the progress for users.

## Chapter 4

### System Implementation

This chapter describes how we implemented 3D FAST. We decided on the Java programming language, since it is cross platform. The graphical user interface is built with Swing, which is a GUI widget toolkit for Java. Visualization Tool Kit (VTK), an open-source software system for 3D computer graphics, image processing and visualization, is used for 3D mesh rendering. The details of the 3D FAST implementation are discussed in the following paragraphs.

#### 4.1 Graphical User Interface (GUI)

This system uses the Java Swing library as a standard method for developing its GUI. Swing has all kinds of classes for developing a GUI, such as JButton, JMenu, JPanel, JTable, etc. For example, as discussed in Chapter 3, the workspace works as a desktop, and it was implemented with the JDesktopPane class. JButton class creates buttons that are used for users to issue commands. Buttons at the toolbars are popup buttons, which means a popup menu will show up when the user clicks on one of the buttons. The components used in this GUI will be illustrated in the following sections.

##### 4.1.1 Workspace

The workspace is the only place where users can operate on the meshes and acquire the information about images. We additionally added a ‘drag and drop’ feature so that users can open images by dragging the polygon files (.ply) and dropping it onto the workspace.

Information about one image, such as facial landmarks, mid-facial plane and symmetry scores, can be read from image windows, which were implemented with JInternalframe class. The image windows are added to the workspace when users double click on or drag and drop the image tree nodes. As shown in Figure 8, an image window consists of a toolbar, a mesh pane at the left side and landmark data at the right side. The actions on the toolbar allow users to control the landmarks and the mid-facial plane, and the last button is a shortcut for users to save the current mesh. In fact, an image window contains three tabbed panes at the right side. As displayed in Figure 9, the right pane changes its contents according to the chosen tabbed pane. The mesh pane and the contents of the three tabbed panes are as follows:

### **Image Mesh**

The mesh pane of an image window is responsible for displaying 3D mesh, landmarks and mid-facial plane, and the visualization is rendered by the VTK library. If a landmark file exists, the landmarks will be shown once the image is opened. Adding, moving and deleting landmarks are achievable through the operation of a keyboard and a mouse. For instance, the Shift key and left mouse click will add a new landmark. The Ctrl key and left mouse click will delete an existing landmark. Right clicking on a landmark and right clicking again on another mesh position will move the landmark to the new position. Also, landmarks can be turned on or off by pushing ‘Landmarks On/Off’ at the toolbar of the image window.

The mid-facial plane can be displayed as long as there exists a mid-facial plane file, and it can be turned on or off as well through pressing the ‘Mid-plane On/Off’ button on the toolbar above.

Users can also perform mesh editing in this pane and save the new mesh after the edit completion. The mesh editing function will be further discussed later.

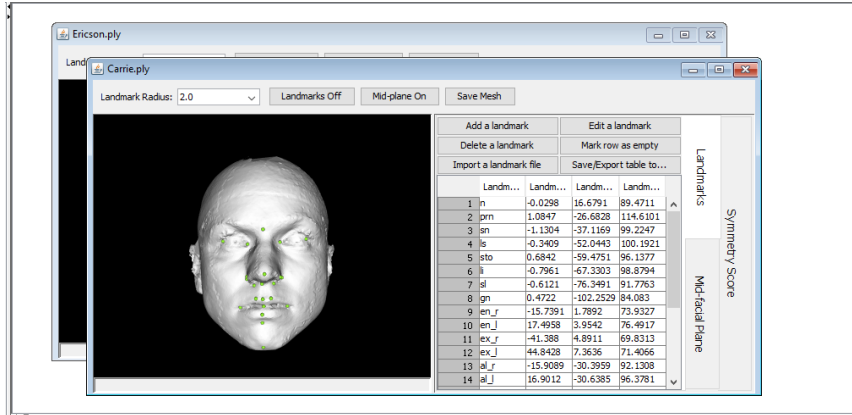
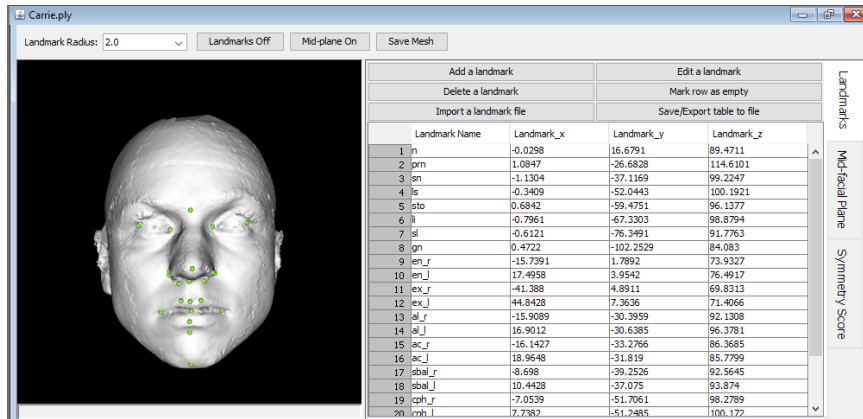
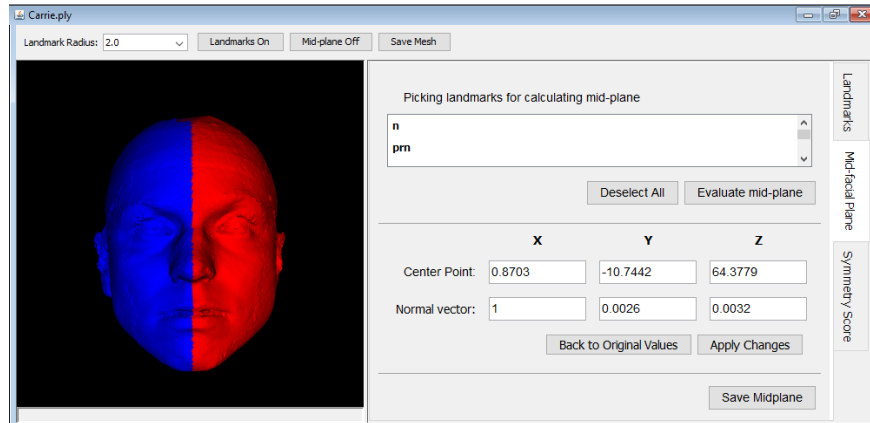


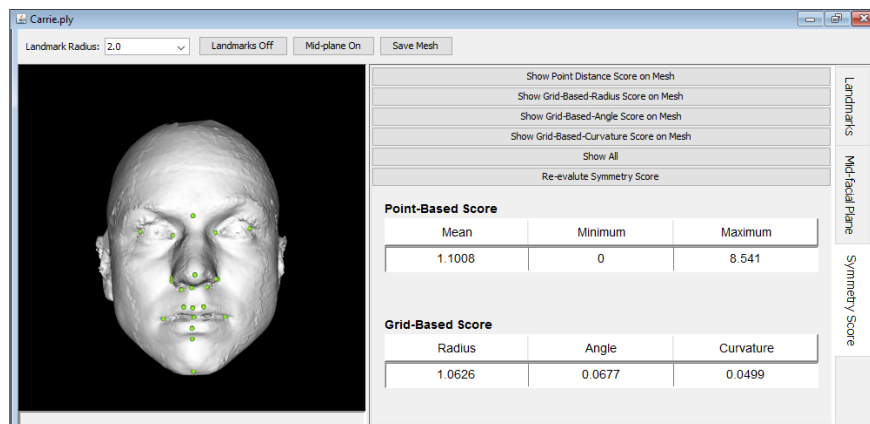
Figure 8. Image windows within the workspace



(a)



(b)



(c)

Figure 9. Three tabbed panes for (a) landmarks, (b) mid-facial plane and (c) symmetry scores

## Landmarks

Figure 9 (a) illustrates the landmark tabbed pane, which shows all landmark coordinates with a table and also allows users to edit the landmark data. For example, users can add/edit/delete landmarks through the buttons above the table. Figure 10 shows a picture of adding a landmark

in which a small window showed up when the user clicked on the ‘Add a landmark’ button. A new landmark will be added to the mesh and the table, after the user enters the coordinates in the small window. Moreover, users are able to import landmark data from a landmark file or export data from the table. Additionally, a landmark on the mesh will be highlighted when the user selects one row of the table, and vice versa. This feature is useful for users to quickly find a landmark.

### **Mid-facial Plane**

In Figure 9 (b), the mid-facial plane shows the center point and normal vector coordinates at the lower right part. The fields for coordinates can be edited, and then the mid-facial plane will be recalculated and reapplied to the mesh. Figure 11 shows the mid-plane before and after changing the x coordinate of the center point from 0.8703 to 18.703.

The default landmarks for mid-plane calculation are n, en, ex, ch and t. The upper part of this tabbed pane allows users to calculate mid-facial plane by selecting landmarks other than default. Finally, the mid-plane can be saved by pressing the ‘Save Midplane’ button.

### **Symmetry Score**

Figure 9 (c) shows the tabbed pane for symmetry scores and buttons at the top. By pressing the buttons, users can check facial symmetry on the mesh as shown in Figure 12. When users change the landmark data or the mid-facial plane, the symmetry scores can be re-evaluated through the ‘Re-evaluate Symmetry Score’ button. In Figure 12, the upper left, upper right, lower left and

lower right windows are showing facial symmetry according to the results of the point-distance method, the grid-based curvature difference method, the grid-based radius difference and the grid-based angle difference method, respectively.

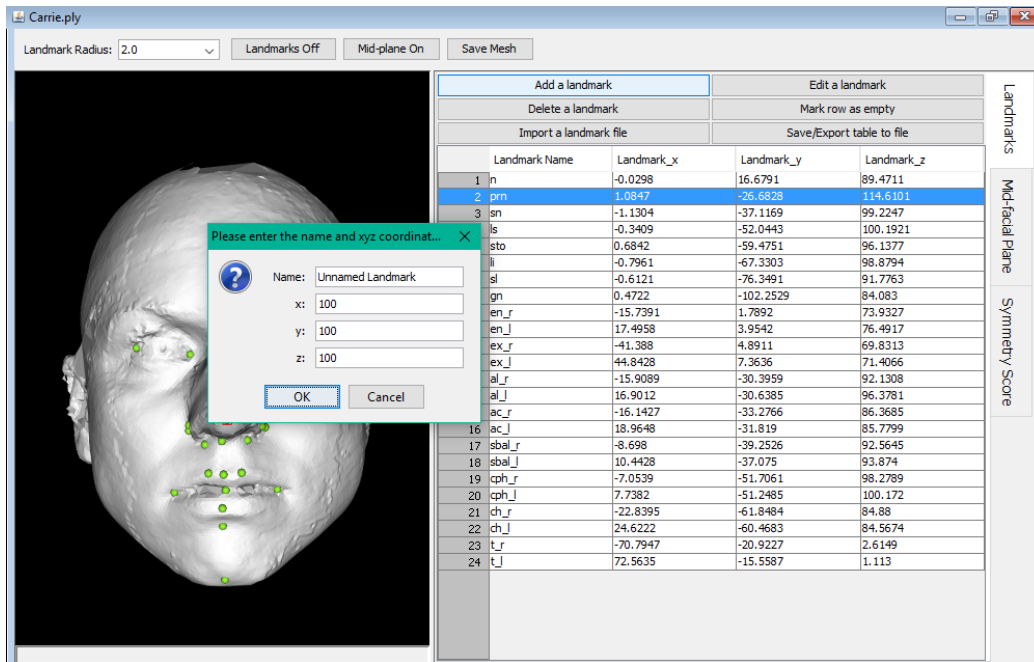


Figure 10. Example of adding a landmark through 'Add a landmark' button

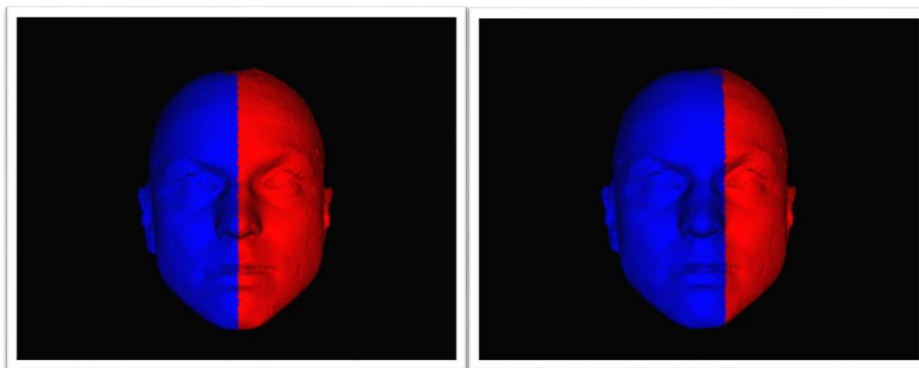


Figure 11. Mid-facial plane before and after changing the center point.

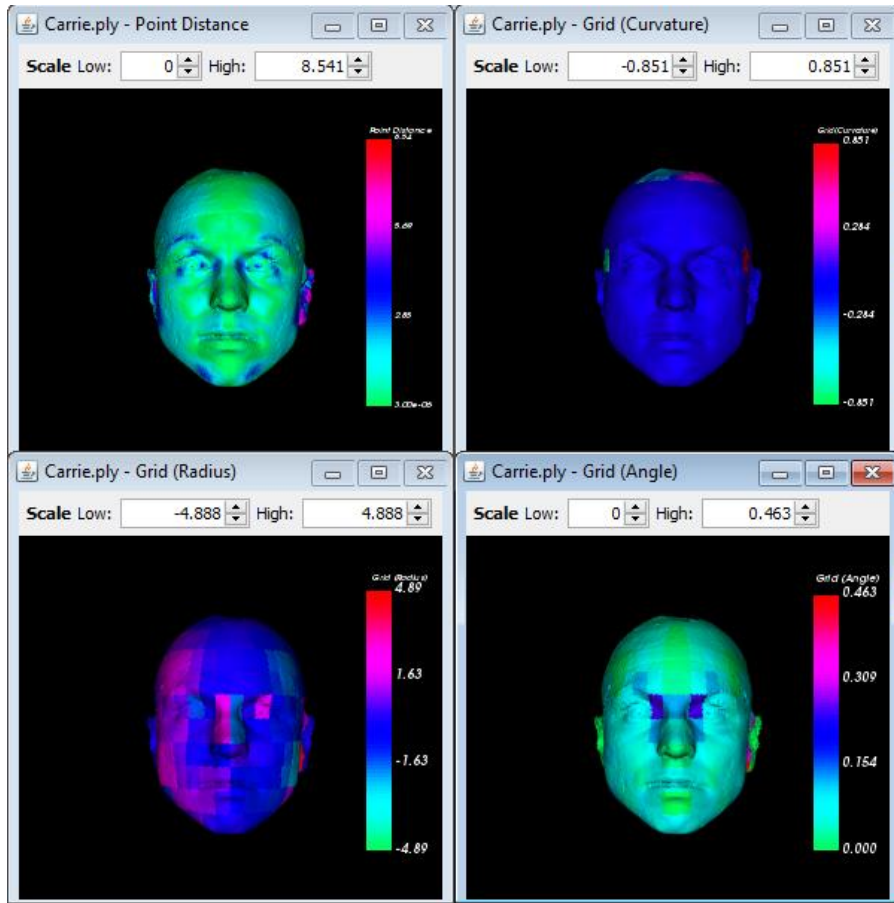


Figure 12. Symmetry scores based on point-based and grid-based methods

#### 4.1.2 Progress Table

A progress table is added to the panel below the workspace when users plan to do an analysis. Figure 13 shows an example of a progress table. The text in bold at the top of the panel is the name of the analysis that is going to start, and it is ‘Landmarks and Mid-facial Plane Detection’ in this example. The table lists file path, file name, status, progress and view button for each image. The status shows ‘Pending’ before the analysis, ‘Processing’ during the analysis, and ‘Done’ after completing the analysis. The green progress bar shows the progress of image

processing; when it reaches the end the processing is 100% done, and then the view button is enabled so that users can view the results by pressing the button.

During an analysis, users are allowed to cancel the process via the ‘Cancel’ button. This increases the flexibility to control the analysis, as users can stop the process anytime once poor results have been obtained.

File	File Name	Status	Progress	Report
1	C:\Users\Janice\Desktop\gui_demo\pr... Carrie.ply	Done	<div style="width: 100%; height: 10px; background-color: green;"></div>	View
2	C:\Users\Janice\Desktop\gui_demo\pr... Eric.ply	Processing	<div style="width: 50%; height: 10px; background-color: green;"></div>	View
3	C:\Users\Janice\Desktop\gui_demo\pr... Ericson.ply	Pending		View
4	C:\Users\Janice\Desktop\gui_demo\pr... Kasia.ply	Pending		View

Figure 13. Analysis progress table panel

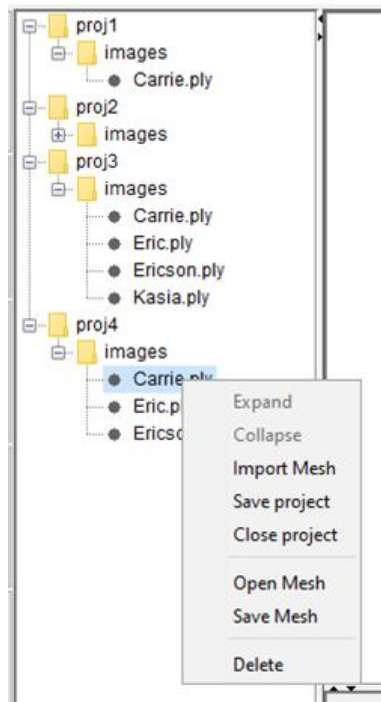


Figure 14. Project tree

### 4.1.3 Project Tree

As shown in Figure 14, all of the opened projects in this GUI are added to the tree. The root of the tree is the directory of the workspace, which is hidden from the tree, because all of the projects are created under the workspace. The tree structure allows users to perform many actions, such as opening images by double-click or drag-and-drop. A popup menu shows up when a tree node is clicked with the right mouse key. Users can perform the actions listed in the menu. These actions are like shortcuts that are helpful for increasing work efficiency.

## 4.2 Analysis Modules Execution

3D FAST is specifically designed for cleft severity analysis, and there are three main modules for analyzing facial symmetry: landmark detection, mid-facial plane detection and symmetry score calculation. The original code was written in C++, so the ability to call the modules from Java is the key factor to making a successful GUI.

In Java, the class ‘ProcessBuilder’ is used to create operating system processes. Constructing an object from this class and passing in appropriate arguments to the constructor enable us to run the original command line code through the GUI. ProcessBuilder creates a Process object when it uses the start() method. The standard output from the modules can be accessed by the Process object, and it then was used to update the progress for users.

Depending on the number of images in a project, normally running a module takes a long period of time for a batch job. To avoid interfering with the interaction between users and the GUI when executing time intensive computing, the modules will be running in the background. The class ‘Swingworker’ in Java was used to perform lengthy tasks in a background thread. Also, it helps

provide updates to the GUI during the executing period. For example, the landmark data in an opened image window will be updated automatically once the detection for the image is done.

In summary, to execute an analysis module, we used 'ProcessBuilder' for execution and 'Swingworker' for running in the background. Swingworker class updates the progress bar according to the standard output read from Process object, and then updates the view button and all of the image information when a process is completed. Figure 15 describes the module execution.

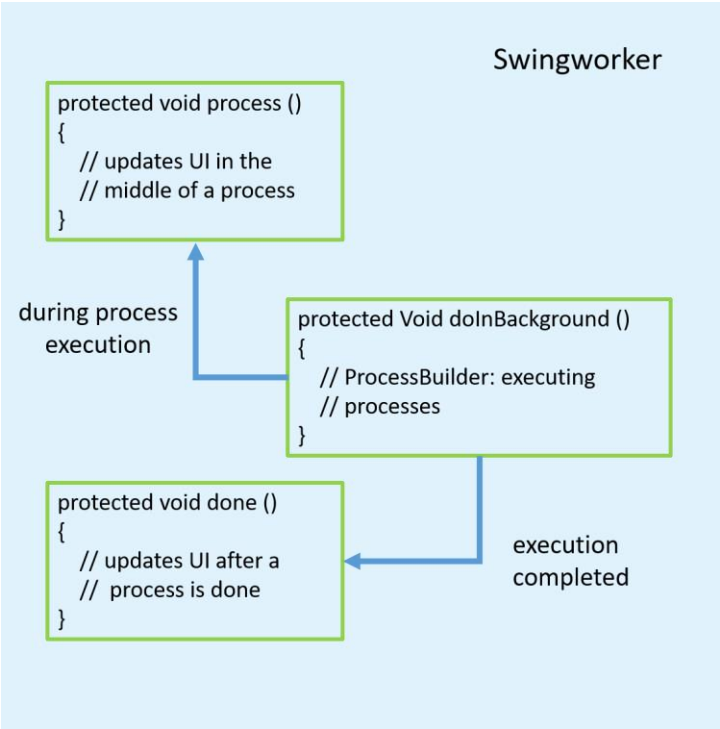


Figure 15. Module execution using Swingworker in Java

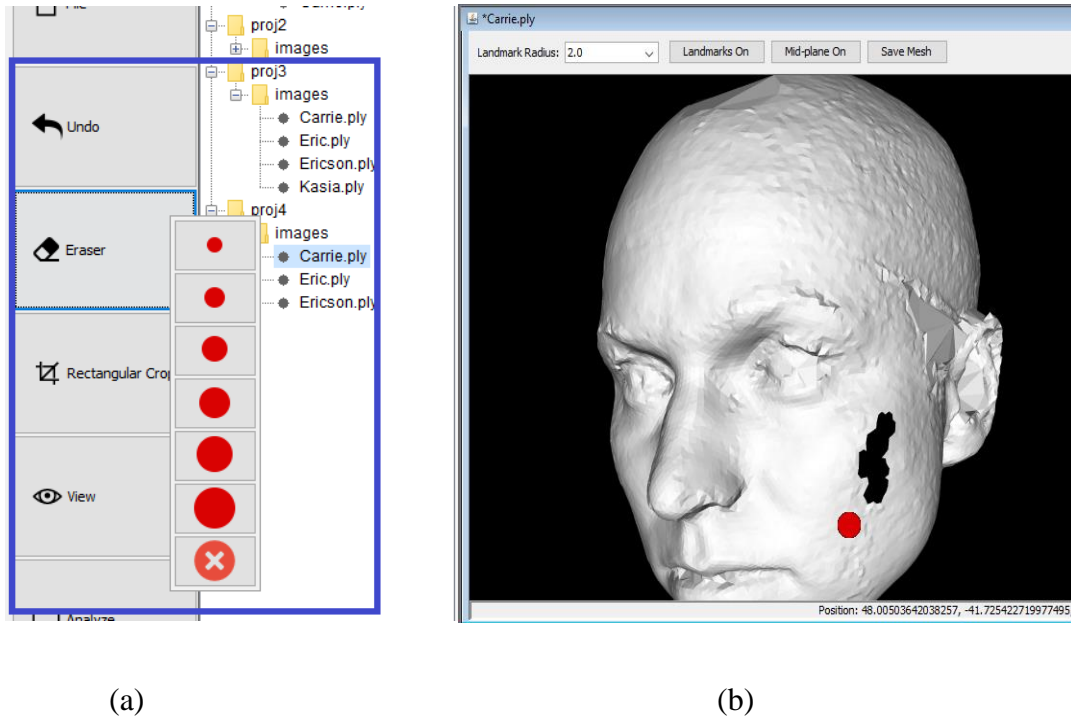


Figure 16. Mesh cropping: (a) eraser button and popup menu, (b) mesh removal demonstration.

### 4.3 Mesh Cropping

Two types of mesh cropping were implemented in this system: 2D and 3D cropping. Two-dimensional cropping works as an eraser, which enables users to remove meshes by dragging a mouse. As shown in the blue rectangle in Figure 16 (a), a popup menu of eraser sizes shows up after the Eraser button is clicked. Users may choose their preferred eraser size to perform mesh removal. Figure 16 (b) demonstrates mesh removal using the eraser, which is shown by the red circle in the picture.

Three-dimensional mesh cropping is to crop the mesh through a cuboid. Users may drag a mouse from a point and draw a two-dimensional rectangle over the mesh that is going to be removed. According to the drawn rectangle, the eraser function creates a cuboid that is long enough to

cover the mesh in depth (z-direction), and then cut the mesh with it. As displayed in Figure 17 in the left picture, a two-dimensional rectangle was drawn, resulting in the mesh shown in the right picture.

Both the eraser and rectangular cropping were implemented with the VTK library. VTK supports mesh removal mostly in three dimensions, so rectangular cropping can perform faster. The eraser consists of more steps, and thus the implementation is rather complicated. There are five steps to complete the eraser action:

1. Find the mouse position and use it as the center of the circle.
2. Locate all points within the radius of the circle.
3. Extract the resulting mesh and erase this mesh separately.
4. Update the display.
5. Store the erased mesh in the undo list.

Step 3 takes longer to complete, since there are more filter actions working with VTK.

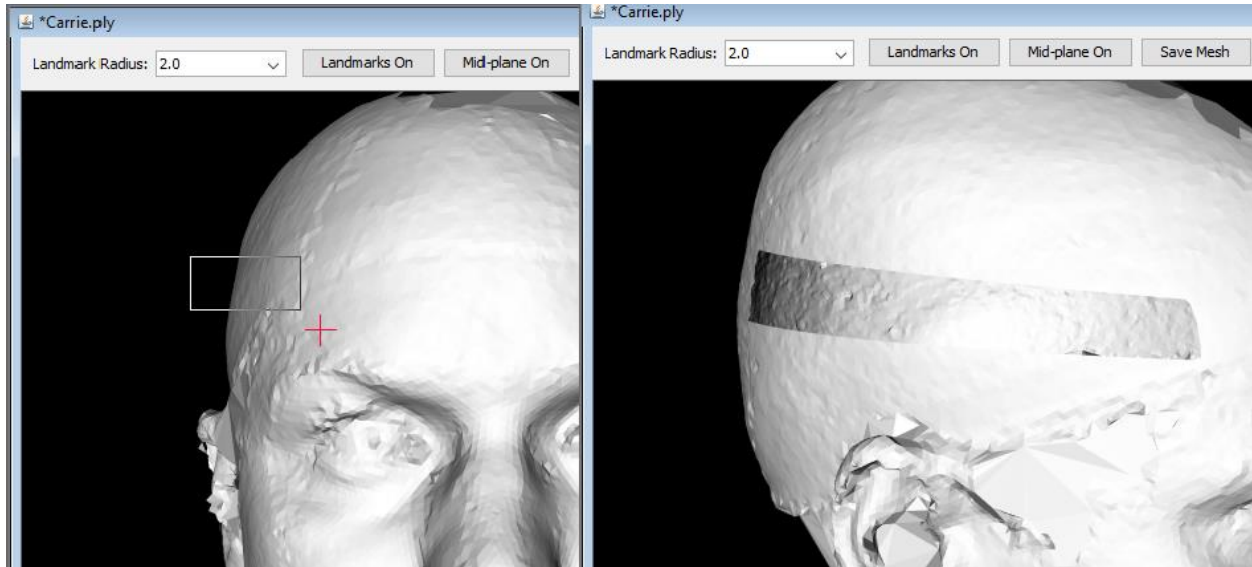


Figure 17. 3D rectangular cropping

The Undo feature is also implemented for the mesh editing functions. Edits of eraser and rectangular cropping are combined into an undo list, so the removal action is undone in the sequence of edits made by eraser and rectangular cropping.

#### 4.4 System Requirements

3D FAST was implemented in the Java environment in which the VTK library was used to handle all of the mesh-related jobs. The modules in C++ must be compiled separately. Therefore, in order to successfully run the system, the host computer must install:

- Java Development Kit (JDK). 3D FAST was developed using jdk1.8.0\_66.
- Java Runtime Environment (JRE). (JDK comes with a JRE, so there is no need to download a separate JRE if JDK already exists.)

- A C++ compiler. 3D FAST is developed using Microsoft Visual Studio 2015's C++ compiler.
- VTK library. 3D FAST is developed using VTK 6.3.

## 4.5 Instruction/ Sample Use Case

### 4.5.1 Add a New Project and Import Images

**Create a project:** File > New Project

To create a new project click 'File' on the toolbar and then click 'New Project'. Enter a project name in the dialog window.

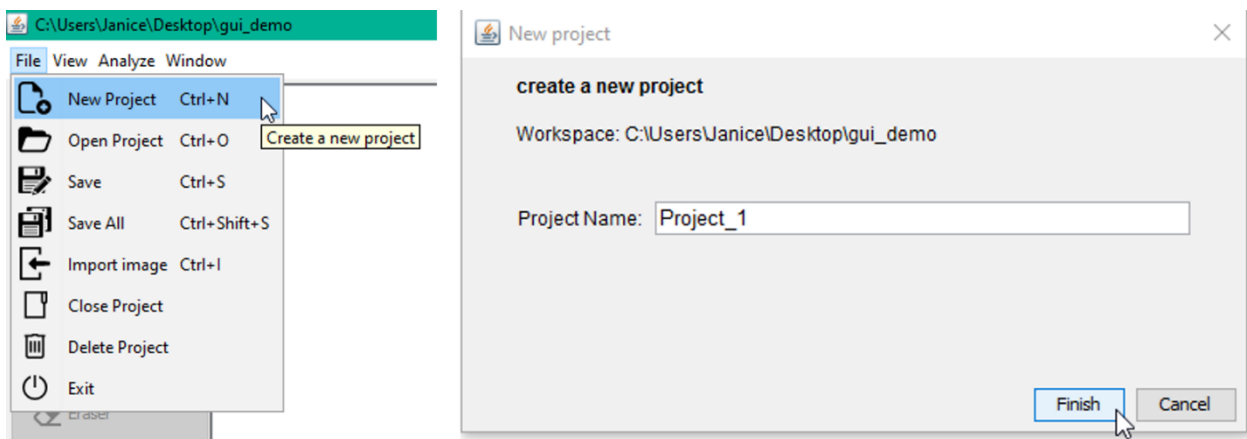


Figure 18. Creating a new project

**Import image files:** File > Import Image

To import images click 'File' on the toolbar and then click 'Import Image' as shown in Figure 19. Only .ply files are supported in this system. Figure 20 shows a file format filter, which will show .ply files only. Multiple selection is allowed.

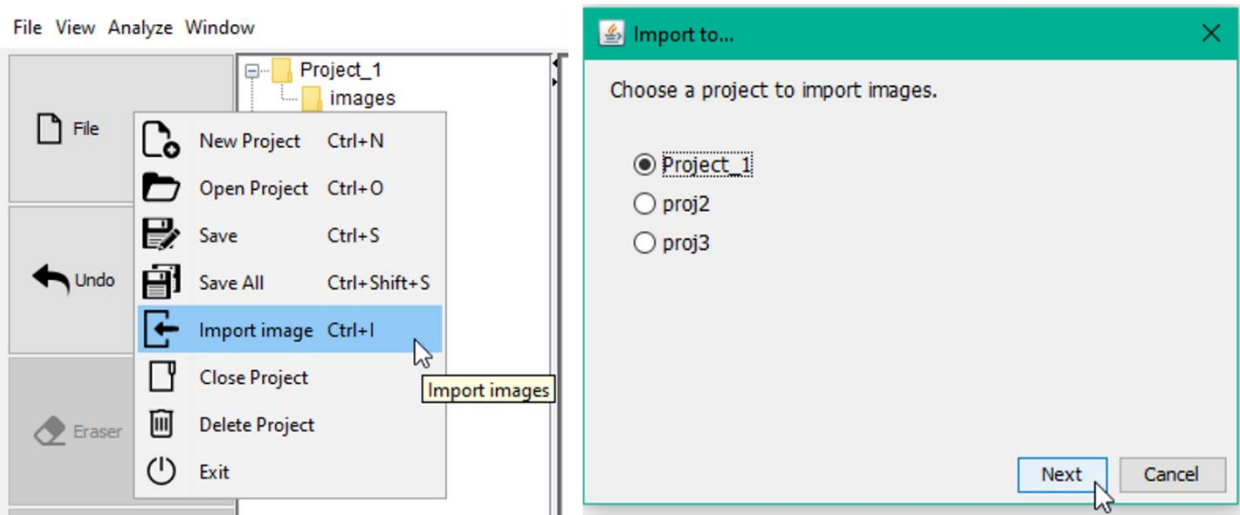


Figure 19. Importing images to a project

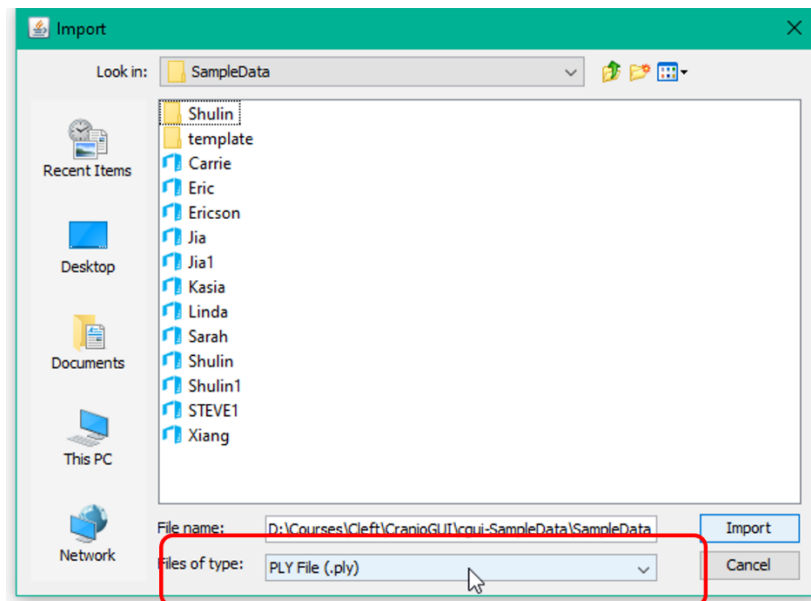


Figure 20. Importing .ply file

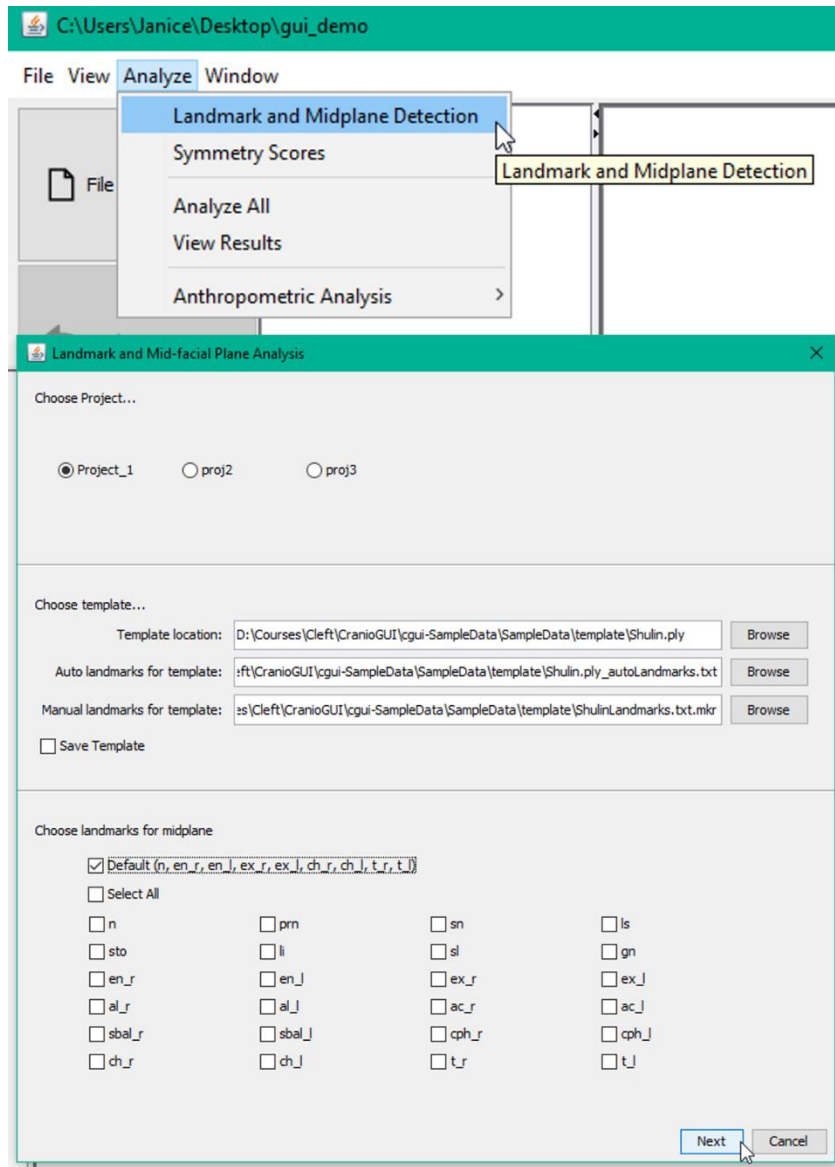
#### 4.5.2 Analyze: Landmark and Midplane Detection

Actions to start landmark and mid-facial plane detection: ‘Analyze’ > ‘Landmarks and Midplane Detection’. This is described in Figure 21 (a).

In the dialog window, enter the following information:

1. Choose the appropriate project.
2. Provide file paths for the template image file (.ply), the auto landmark file (.txt) and the manual defined landmark file (.mkr). Check ‘Save Template’ if the template will be frequently used.
3. Check any landmarks that will be used to calculate the mid-facial plane.

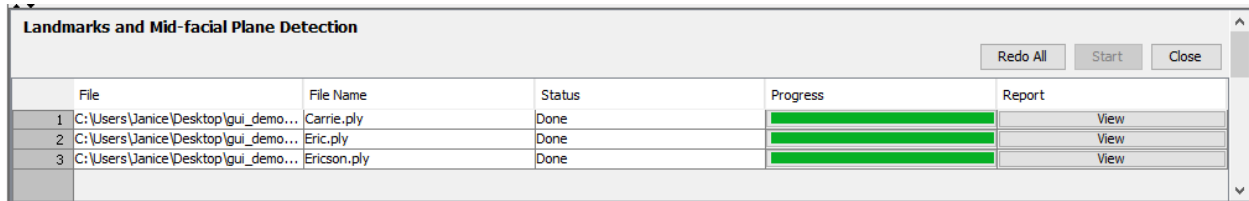
Finally, click ‘Next’ to continue. A table listing the images in the chosen project will show up at the bottom of the GUI. The title of the analysis will be at the top of the panel, and users should check if it is the right analysis. If the landmark files and the mid-facial files exist for some .ply files, the Status field of those files will show ‘Done’; otherwise, it will be ‘Pending’. Click ‘Start’ to begin the analysis. Figure 21 (b) and (c) describe the table panel before and after the analysis.



(a)

Landmarks and Mid-facial Plane Detection					
File	File Name	Status	Progress	Report	
1	C:\Users\Janice\Desktop\gui_demo... Carrie.ply	Pending			View
2	C:\Users\Janice\Desktop\gui_demo... Eric.ply	Pending			View
3	C:\Users\Janice\Desktop\gui_demo... Ericson.ply	Pending			View

(b)



	File	File Name	Status	Progress	Report
1	C:\Users\Janice\Desktop\gui_demo...	Carrie.ply	Done	<div style="width: 100%; background-color: green;"></div>	View
2	C:\Users\Janice\Desktop\gui_demo...	Eric.ply	Done	<div style="width: 100%; background-color: green;"></div>	View
3	C:\Users\Janice\Desktop\gui_demo...	Ericson.ply	Done	<div style="width: 100%; background-color: green;"></div>	View

(c)

Figure 21. Steps for landmark and mid-facial plane detection: (a) argument setting, (b) table panel before the analysis starts, (c) table panel after the analysis completed.

Clicking on the ‘View’ buttons, we can check the landmarks in ‘Landmarks’ tab of the image window as shown in Figure 22. The landmarks are represented by the green dots and the size can be adjusted from the drop-down menu ‘Landmark Radius’ on the toolbar. Please refer to section 4.5.7 for the way to edit the landmarks.

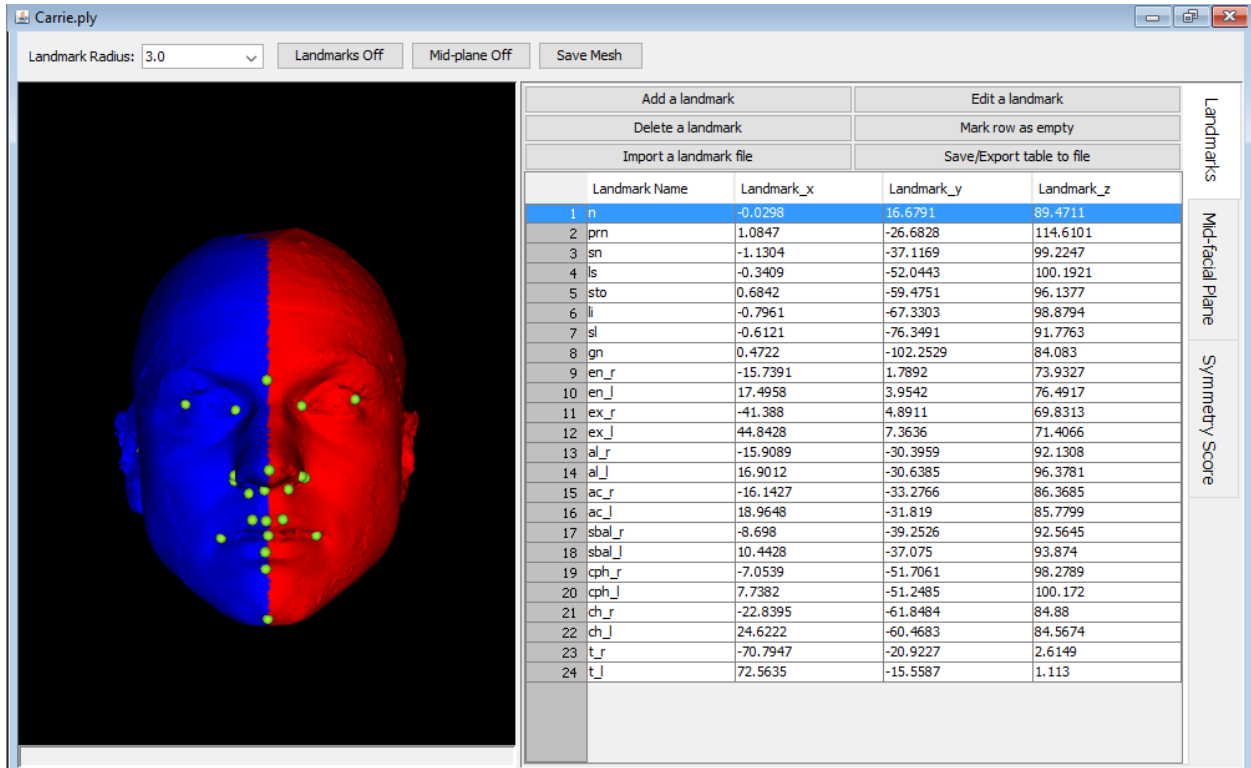


Figure 22. Landmark data after landmark detection

Figure 23 shows the mid-facial plane data in the 'Mid-facial Plane' tab. The center point and normal vector are read from the mid-facial plane file which is generated by the detection process. The way to change the mid-facial plane is described in section 4.5.9

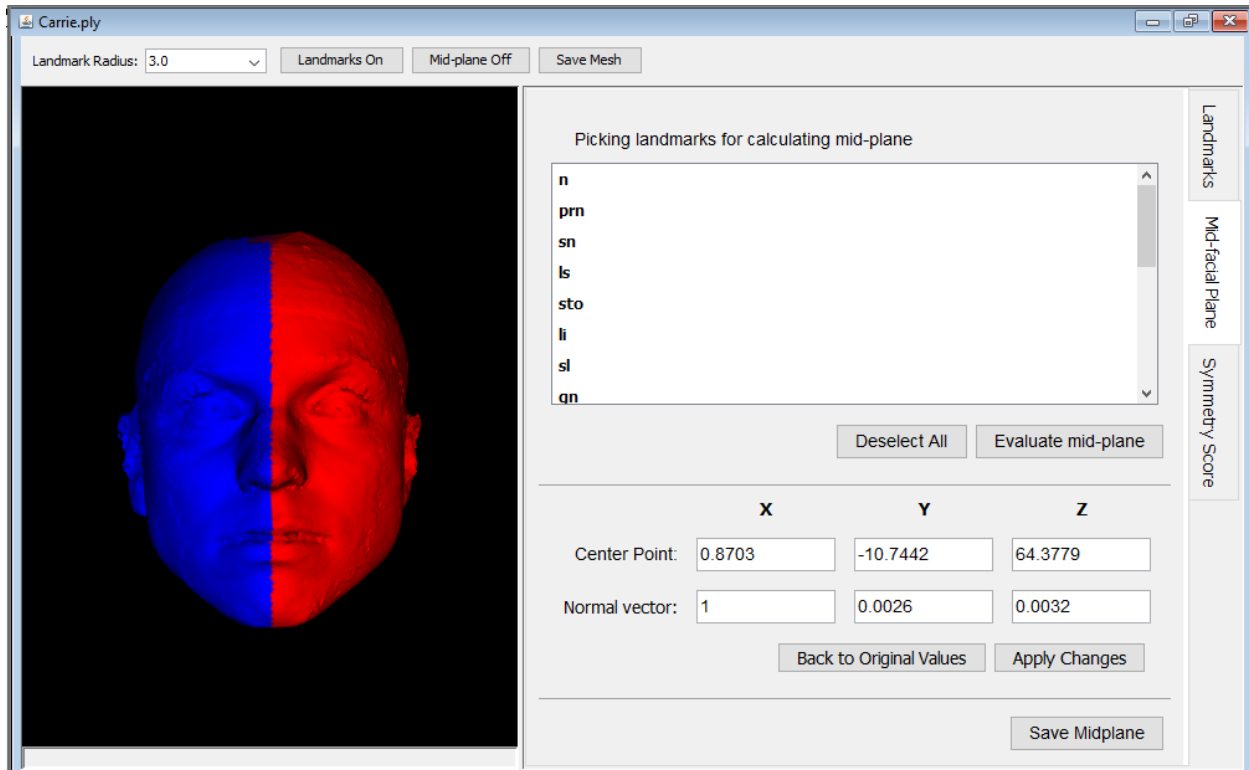


Figure 23. Mid-facial plane data

#### 4.5.3 Analyze: Symmetry Scores

Symmetry scores describe the symmetry of a face. 3D FAST provides four kinds of score based on four different methods: point-based point distance, grid-based radius distance, grid-based angle distance, and grid-based curvature distance [6].

Before the calculation, users should make sure that the mid-facial plane has been detected, which means the mid-facial plane file for each image in the project exists.<sup>1</sup>

Actions to start the analysis: ‘Analyze’ > ‘Symmetry Scores’. Then choose a project in the dialog shown in Figure 24 for the calculation and click ‘Next’. As Figure 25 shows, a table listing all

<sup>1</sup> By default, the mid-facial plane file is saved as ‘\*.ply\_mid\_plane.txt’ in the landmarks folder under the project path. ‘\*’ represents the image file name.

the .ply files will appear at the bottom of the GUI. Please make sure this is the right analysis by checking the title. Next, click ‘Start’ to begin the process.

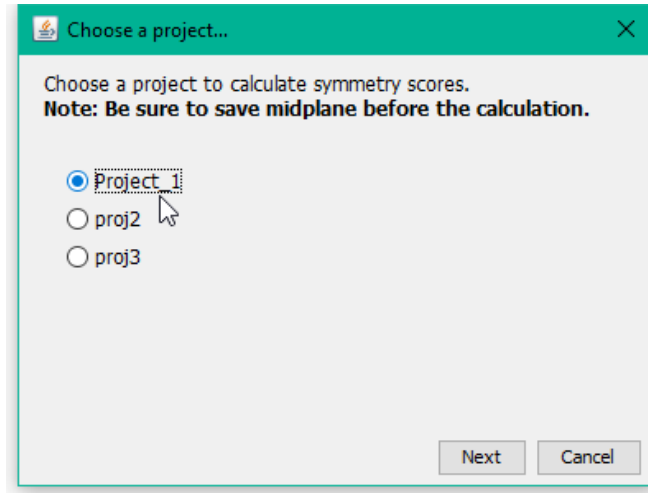


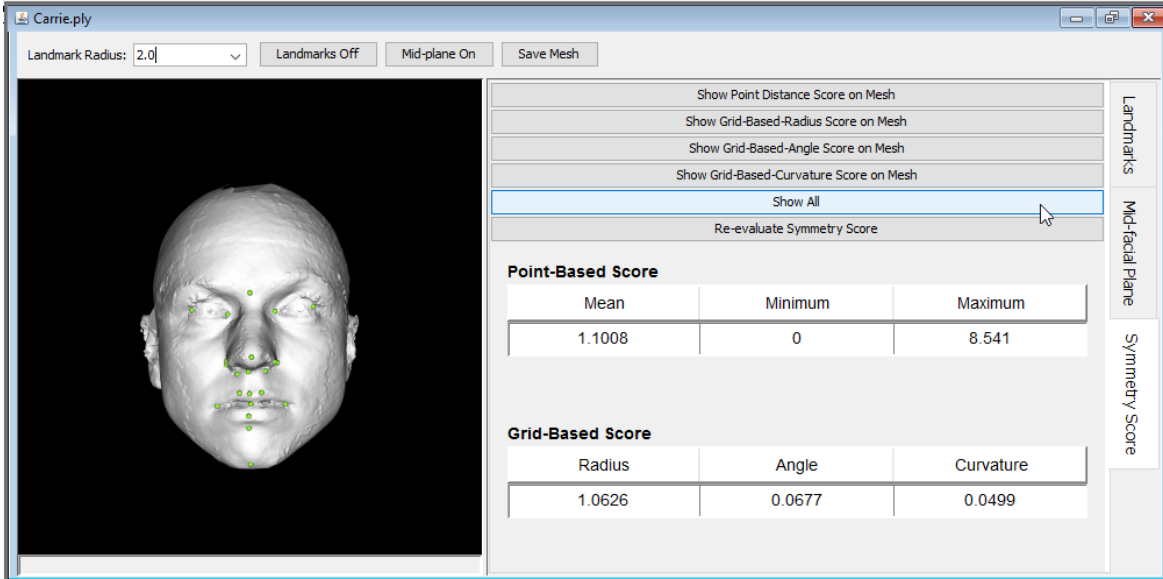
Figure 24. Dialog for choosing a project before the symmetry score calculation

Symmetry Scores Calculation					
File	File Name	Status	Progress	Report	
1	C:\Users\Janice\Desktop\gui_demo... Carrie.ply	Pending			View
2	C:\Users\Janice\Desktop\gui_demo... Eric.ply	Pending			View
3	C:\Users\Janice\Desktop\gui_demo... Ericson.ply	Pending			View

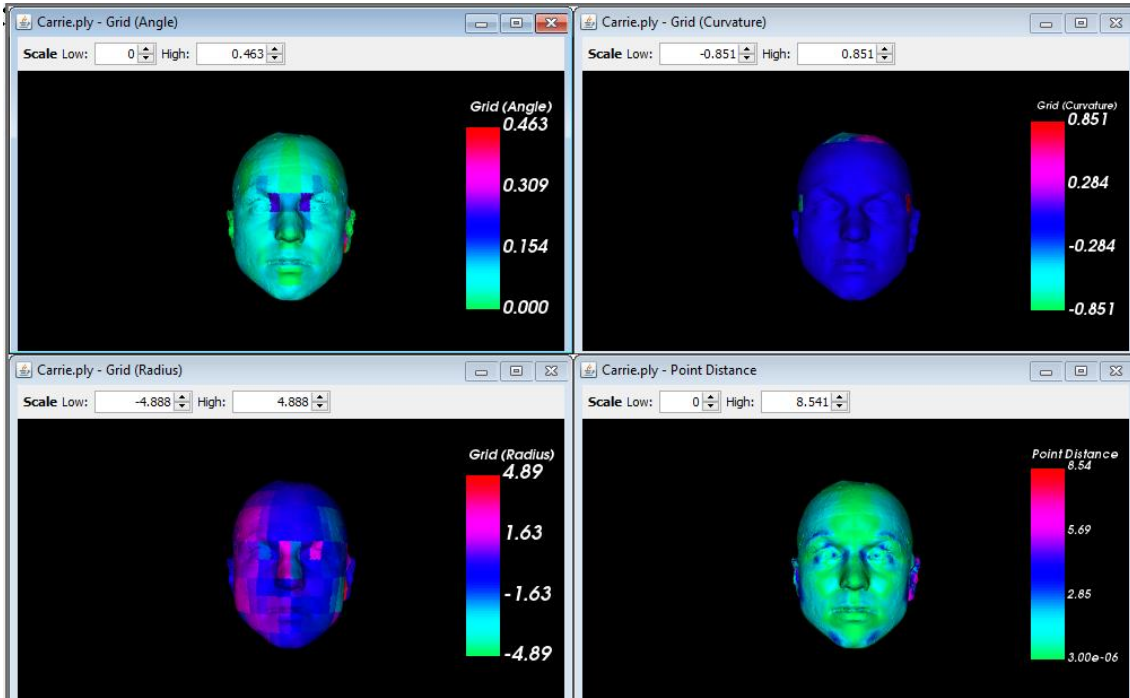
Figure 25. Table panel for symmetry scores calculation

Users may check the scores in the ‘Symmetry Score’ tab of the image window by clicking on the ‘View’ button or by double-clicking on the tree node. As in Figure 26 (a), the upper table shows the scores calculated with the point-based point distance method, and the scores in the lower table are from the grid-based methods.

Clicking on the 'Show All' button will visualize the facial symmetry in the four windows based on the four methods separately, which is shown in Figure 26 (b). On the toolbar of the four mesh windows users may change the scale for an appropriate view.



(a)



(b)

Figure 26. Symmetry scores display

#### 4.5.4 Analyze: Analyze All

Analyze All is a convenient shortcut that allows users to do all of the analysis at one time. The process includes landmark detection, mid-facial plane evaluation and symmetry scores calculation. The procedure is the same as described in sections 4.5.2 and 4.5.3.

#### 4.5.5 Analyze: View Results

The ‘View Results’ function allows users to compare the symmetry scores of the images contained in the same project. To perform this, click ‘Analyze’ and then click ‘View Results’ in the popup menu as shown in Figure 27.

The action will bring up a new window on the workspace as displayed in Figure 28. The table in the window shows the symmetry scores of each image, and it allows the user to sort the scores by clicking on the column heading such as the Point-Based Score.

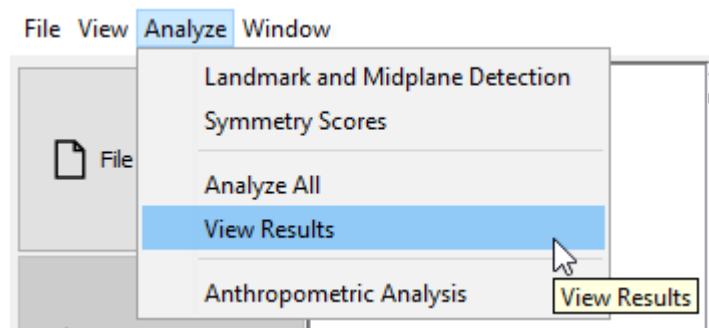


Figure 27. Steps to view the results of symmetry scores

File Name	Point-Based Score	Grid-Based Radius Score	Grid-Based Angle Score	Grid-Based Curvature Sc...
Carrie.ply	1.100757	1.062609	0.067678	0.049947
Ericson.ply	1.104189	1.282343	0.068427	11.366029
Eric.ply	3.200431	1.703174	0.080794	0.023656

Figure 28. Comparison of the symmetry scores

#### 4.5.6 Analyze: Anthropometric Analysis

Anthropometric analysis outputs comma-separated values (CSV) files that can be read with Microsoft Excel. Two types of data can be exported:

1. Distance between landmark points
2. Symmetry scores

Actions to output the data: Analyze > Anthropometric Analysis > Export Point Distance Data/ Export Symmetry Scores. This is illustrated in Figure 29.

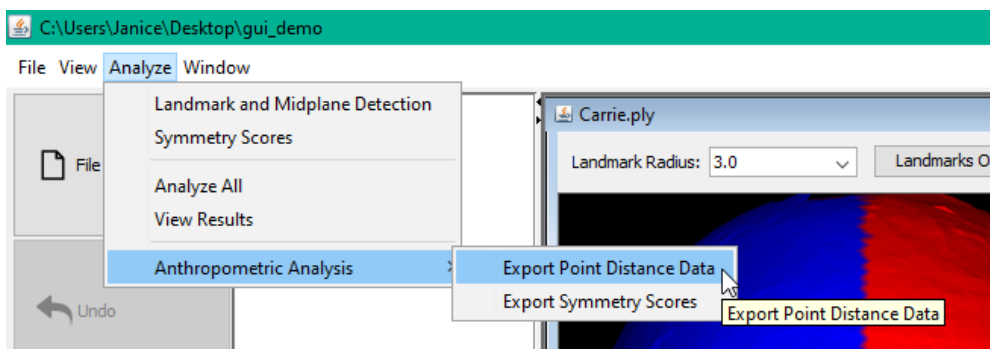


Figure 29. Steps to export CSV data

In the dialog (Figure 30), choose the project for exporting data. Then save the exported file and check it with the appropriate software, such as Microsoft Excel and a text editor. Figure 31 and Figure 32 are the demonstrations for the two types of data.

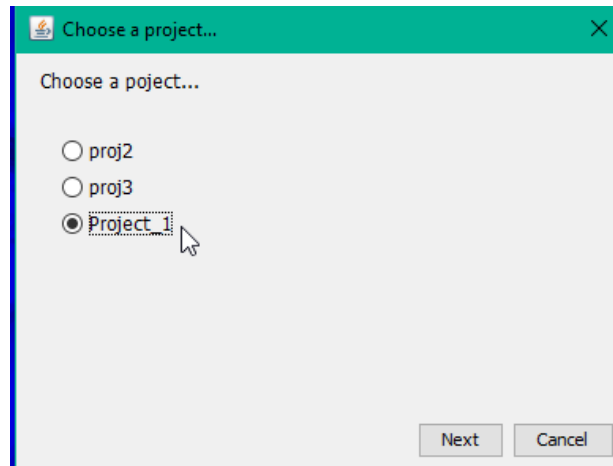


Figure 30. Dialog for exporting data

	A	B	C	D	E	F	G	H	I	J	K
1	C:\Users\Janice\Desktop\gui_demo\Project_1\images\Carrie.ply										
2		n	prn	sn	ls	sto	li	sl	gn	en_r	en_l
3	n	0	50.13434	54.68401	69.5552	76.44867	84.53804	93.05858	119.0551	26.64466	25.24939
4	prn	50.13434	0	18.72132	29.20817	37.63941	43.62591	54.69011	81.50542	52.42466	51.58449
5	sn	54.68401	18.72132	0	14.97953	22.64313	30.21732	39.93644	66.89209	48.64956	50.50305
6	ls	69.5552	29.20817	14.97953	0	8.52677	15.34911	25.72209	52.73593	61.84416	63.36945
7	sto	76.44867	37.63941	22.64313	8.52677	0	8.450669	17.47674	44.44445	67.20189	68.49722
8	li	84.53804	43.62591	30.21732	15.34911	8.450669	0	11.48152	37.94905	74.98762	76.92394
9	sl	93.05858	54.69011	39.93644	25.72209	17.47674	11.48152	0	27.04387	81.56485	83.72665
10	gn	119.0551	81.50542	66.89209	52.73593	44.44445	37.94905	27.04387	0	105.7857	107.8304
11	en_r	26.64466	52.42466	48.64956	61.84416	67.20189	74.98762	81.56485	105.7857	0	33.40351
12	en_l	25.24939	51.58449	50.50305	63.36945	68.49722	76.92394	83.72665	107.8304	33.40351	0

Figure 31. Demonstration for the point distance data

	A	B	C	D	E	F	G
1		Point-Base	Grid-Based	Grid-Based	Grid-Based	Curvature Score	
2	Carrie	1.1008	1.0626	0.0677	0.0499		
3	Eric	3.2004	1.7032	0.0808	0.0237		
4	Ericson	1.1042	1.2823	0.0684	11.366		

Figure 32. Demonstration for the symmetry scores data

#### 4.5.7 Add/ Edit/ Remove a landmark

To change the landmark position, users may click on the buttons above the landmark table and enter the preferred x-y-z coordinates, or directly operate on the mesh. For editing landmarks on the mesh:

- Moving a landmark: right-click on a landmark, and the landmark color will become red when it is selected. Then move the mouse to a new position, and right-click again on the new position.
- Adding a landmark: Shift key + left-click on the mesh where the new landmark will be.
- Deleting a landmark: Ctrl key + left-click on a landmark which will be removed.

#### 4.5.8 Import/ Export a landmark file

3D FAST allows users to import a landmark file (.mkr) or a landmark template with landmark names and no landmark coordinates. To import an .mkr file click 'Import a landmark file' button and choose an .mkr file. To export or save the current landmark data, click 'Save/Export table to

file’, enter a file name and click on ‘Save’. Note that the file extension of a landmark file should be mkr.<sup>2</sup>

#### 4.5.9 Calculate Midplane by Manually Landmark Selection

Users may change the plane in two ways. First, as Figure 33 shows, the mid-facial plane can be evaluated by selecting landmarks in the list at the upper part of the tabbed pane. Second, change the coordinates of center point and normal vector at the lower part and apply changes. Be sure to save the mid-facial plane once it is finalized.

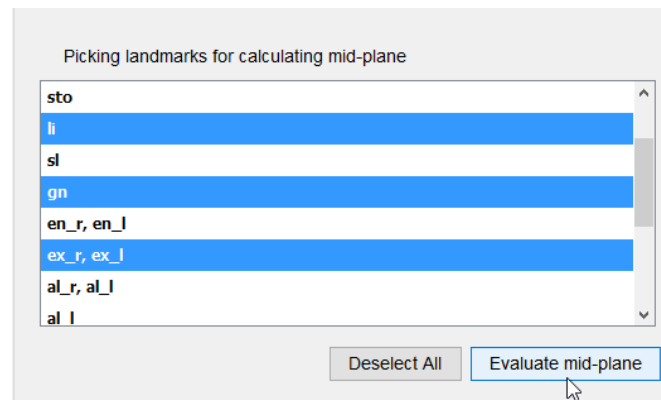


Figure 33. Mid-facial plane evaluation by manual landmark selection

#### 4.5.10 Mesh Removal: Eraser

Eraser is a 2D mesh remover. Actions to use an eraser: Eraser > choose an appropriate size > drag the mouse on the mesh. Figure 34 (a) shows the options for eraser size, and the last one with a cross means exiting the eraser mode. Moving the mouse cursor onto the mesh and dragging will remove the mesh under the red circle. To exit the eraser mode, click on the button with a cross or press the ESC key.

---

<sup>2</sup> The default landmark file is generated through landmark detection. The file name is ‘\*\_deform.ply\_tansLMK.mkr’. ‘\*’ is the PLY file name.

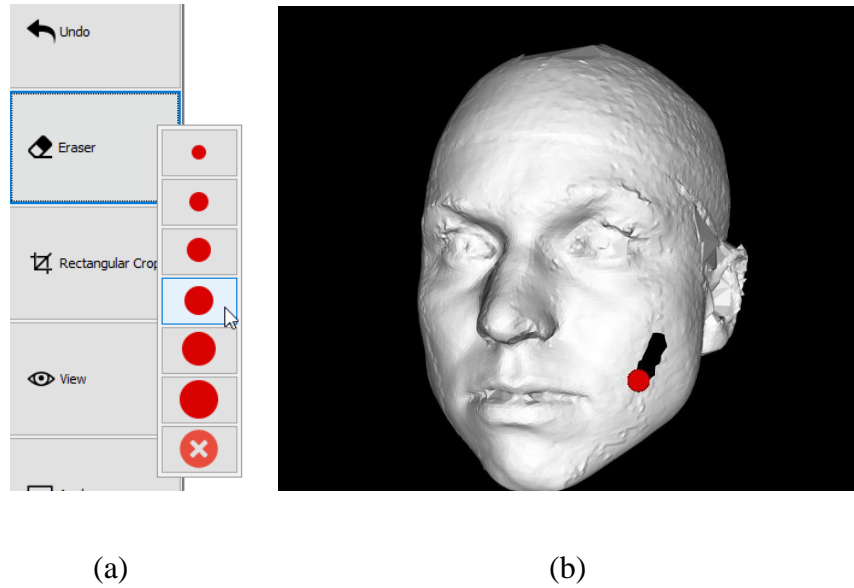


Figure 34. Mesh removal using Eraser: (a) the popup menu listing the options for eraser sizes. (b) the resulting mesh when using the eraser.

#### 4.5.11 Mesh Removal: Rectangular Crop

Rectangular Crop is a 3D mesh remover. To use it, click on 'Rectangular Crop' button and then the mouse cursor will become a cross when moving onto the mesh panel. By dragging the cursor a 2D rectangle can be drawn on the mesh, which can be viewed as a cross section of a cuboid. Next, an invisible cuboid that the length covers the whole mesh in depth (z-direction) is generated. The function then uses the cuboid to crop the mesh. This is shown in Figure 17. To exit, click 'Rectangular Crop' again or press the ESC key.

## Chapter 5

### Future Work and Conclusions

#### 5.1 Future Work

There are a few features waiting to be added once they are completed. These features will make the system more useful and friendlier, and increase efficiency for cleft severity analysis.

##### **Auto-cleaning Module**

3D FAST integrated the algorithms for facial symmetry evaluation and the mesh operation tools. It is easy to start with a cleaned polygon file (.ply file), because it can be directly fed into analysis modules. However, the system could be more useful if it is capable of importing a raw 3D texture image which has a lot of undesired details, such as cloth and other accessories as shown in Figure 35.



Figure 35. Raw data of texture image and 3D mesh

A module named auto-cleaning will be beneficial for users. It saves time in manually scripting mesh cleaning if it is added into the system. Figure 36 illustrates the procedure of auto-cleaning. The raw texture image and 3D mesh like Figure 35 are the inputs to the module, and the output will be a cleaned polygon mesh like Figure 36 (e).

The original code was written in a combination of Matlab and C++, and currently most of the Matlab part was converted to C++, however, it is still in testing. Once this is done it will be integrated into 3D FAST.

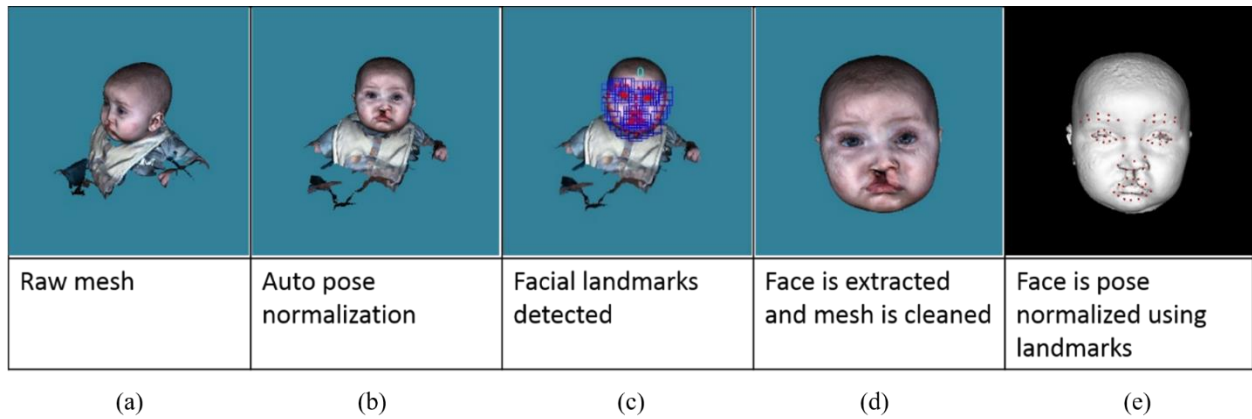


Figure 36. Procedure of the auto-cleaning module

### Mesh Editing

The auto-cleaning module helps remove undesired part of the mesh; however, it may excessively clean the mesh in a way that is out of the user's expectations. In order to improve this, it is desired to undo the action and let the user redefine the region for cleaning. The idea is that the original mesh can be recovered by clicking a button, and the cleaned region will be colored and displayed with a dashed outline as in Figure 37. To redefine the cropping region, users may drag the dashed line to reshape. Finally, users may click on a button to remove the outlined mesh.



Figure 37. Undo the auto-cleaning

## 5.2 Conclusions

The core of the system is the highly integrated set of analysis modules. The complicated algorithms from landmark detection to symmetry scores calculation wrapped and shown with a friendly user interface was the goal of the development. In addition to the functionality for facial analysis, we also implemented a mesh editing tool to increase flexibility. Additional nice properties of this system are that it allows users to run analysis in batch mode, and it is easy to set arguments.

To verify that the goal of a user interface with high usability was indeed reached, we evaluated the GUI with a questionnaire in which the questions can be categorized into four groups:

- (1) Usefulness,
- (2) Ease of Use,

(3) Ease of Learning,

(4) Satisfaction,

(5) User's Comments.

The survey showed the GUI is efficient for cleft severity analysis and easy to learn. Furthermore, it has great promise to facilitate the future medical research. However, it has not been tested on a really large dataset.

In the future, a large-scale testing will be performed to prove the system's reliability. Moreover, it may also add extended functionality to the system, such as new modules and mesh operation tools.

## References

- [1] Jakob Nielsen. Usability Engineering. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN: 0125184050.
- [2] K. Konstantinides and J. Rasure, “The khoros software development environment for image and signal processing”, vol.3, no. 3, pp. 243-252, 1994.
- [3] L. Bavoil, S. Callahan, P. Crossna, J. Freire, C. Scheidegger, C. Silva, and H. Vo, “VisTrails: enabling interactive multiple-view visualizations”, in *Proc. IEEE Visualization VIS 05*, 2005, pp. 135-142.
- [4] O. Ghita, P. Whelan, R. Sadlier, “NeatVision: visual programming for computer-aided diagnostic applications”.
- [5] Michael Lam, Ezgi Mercan, Linda Shapiro, “CraniogUI: A User Interface for Craniofacial Analysis Tools”.
- [6] Jia Wu, Shu Liang, Linda Shapiro, Raymond Tse, “Measuring Symmetry in Children with Cleft Lip. Part 2: Quantification of Nasolabial Symmetry Before and After Cleft Lip Repair”, *The Cleft Palate-Craniofacial Journal*. 2016;53(6):705-713.