Chapter: 8
User Interface Development

This chapter discusses feature and development procedure for Graphical User Interface using MATLAB®. Chapter also describes the User Interface developed for the presented research work.

8.1 Graphical User Interface Environment (GUI)

A graphical user interface (GUI) is a pictorial interface to a program. A good GUI can make programs easier to use by providing them with a consistent appearance and with intuitive controls like pushbuttons, list boxes, sliders, menus, and so forth. The GUI behaves in an understandable and predictable manner, so that a user knows what to expect when he or she performs an action. For example, when a mouse click occurs on a pushbutton, the GUI should initiate the action described on the label of the button. This section introduces the basic elements of the MATLAB® GUIs.


A graphical user interface provides the user with a familiar environment in which to work. This environment contains pushbuttons, toggle buttons, lists, menus, text boxes, and so forth, all of which are already familiar to the user, so that he or she can concentrate on using the application rather than on the mechanics involved in doing things. However, GUIs are harder for the programmer because a GUI-based program must be prepared for mouse clicks (or possibly keyboard input) for any GUI element at any time. Such inputs are known as events, and a program that responds to events is said to be event driven. The three principal elements required to create a MATLAB® Graphical User Interface are:

1. **Components**: Each item on a MATLAB® GUI (pushbuttons, labels, edit boxes, etc.) is a graphical component. The types of components include graphical controls (pushbuttons, edit boxes, lists, sliders, etc.), static elements (frames and text strings), menus, and axes. Graphical controls and static elements are created by the function uicontrol, and menus are
created by the functions uimenu and uicontextmenu. Axes, which are used to display graphical data, are created by the function axes.

2. **Figures:** The components of a GUI must be arranged within a figure, which is a window on the computer screen. In the past, figures have been created automatically whenever data is plotted. However, empty figures can be created with the function figure and can be used to hold any combination of components.

3. **Callbacks:** Finally, there must be some way to perform an action if a user clicks a mouse on a button or types information on a keyboard. A mouse click or a key press is an event, and the MATLAB® program must respond to each event if the program is to perform its function. For example, if a user clicks on a button, that event must cause the MATLAB® code that implements the function of the button to be executed. The code executed in response to an event is known as a callback. There must be a callback to implement the function of each graphical component on the GUI.

MATLAB® GUIs are created using a tool called guide, the GUI Development environment. This tool allows a programmer to layout the GUI, selecting and aligning the GUI components to be placed in it. Once the components are in place, the programmer can edit their properties: name, color, size, font, text to display, and so forth. When guide saves the GUI, it creates working program including skeleton functions that the programmer can modify to implement the behavior of the GUI.

When guide is executed, it creates the Layout Editor, shown in Figure 8.1. The large white area with grid lines is the layout area, where a programmer can layout the GUI. The Layout Editor window has a palate of GUI components along the left side of the layout area. A user can create any number of GUI components by first clicking on the desired component, and then dragging its outline in the layout area. The top of the window has a toolbar with a series of useful tools that allow the user to
distribute and align GUI components, modify the properties of GUI components, add menus to GUIs, and so on.

Figure 8.1: Layout of Guide in MATLAB®

8.1.2 GUI Components

This section summarizes the basic characteristics of common Graphical User Interface components. It describes how to create and use each component, as well as the types of events each component can generate. The components discussed in this section are

- **Text Fields**: A text-field is a graphical object that displays a text string. User can specify how the text is aligned in the display area by setting the horizontal alignment property. By default, text fields are horizontally centered. A text field is created by creating a uicontrol whose style property is 'edit'. A text field may be added to a GUI by using the text tool in the Layout Editor. Text fields do not create callbacks, but the value displayed in the text field can be updated in a callback function by changing the text field's String property.
• **Edit Boxes:** An edit box is a graphical object that allows a user to enter a text string. The edit box generates a callback when the user presses the Enter key after typing a string into the box. An edit box is created by creating a uicontrol whose style property is 'edit'. An edit box may be added to a GUI by using the edit box tool in the Layout Editor.

• **Frames:** A frame is a graphical object that displays a rectangle on the GUI. User can use frames to draw boxes around groups of logically related objects. A frame is created by creating a uicontrol whose style property is 'frame'. A frame maybe added to a GUI by using the frame tool in the Layout Editor. Frames do not generate callbacks.

• **Pushbuttons:** A pushbutton is a component that a user can click on to trigger a specific action. The pushbutton generates a callback when the user clicks the mouse on it. A pushbutton is created by creating a uicontrol whose style property is 'pushbutton'. A pushbutton may be added to a GUI by using the pushbutton tool in the Layout Editor.

• **Toggle Buttons:** A toggle button is a type of button that has two states: on (depressed) and off (not depressed). A toggle button switches between these two states whenever the mouse clicks on it, and it generates a callback each time. The 'Value' property of the toggle button is set to max (usually 1) when the button is on, and min (usually 0) when the button is off. A toggle button is created by creating a uicontrol whose style property is toggle button. A toggle button may be added to a GUI by using the toggle button tool in the Layout Editor. When a user clicks on the toggle button, it automatically calls the function ToggleButton_Callback, This function locates the toggle button using the handles structure and recovers its state from the 'Value' property. Then, the function locates the text field and displays the state in the text field.
• **Checkboxes and Radio Buttons:** Checkboxes and radio buttons are essentially identical to toggle buttons except that they have different shapes. Like toggle buttons, checkboxes and radio buttons have two states: on and off. They switch between these two states whenever the mouse clicks on them, generating a callback each time. The 'Value' property of the checkbox or radio button is set to max (usually 1) when they are on, and min (usually 0) when they are off.

• **Popup Menus:** Popup menus are graphical objects that allow a user to select one of a mutually exclusive list of options. The list of options that the user can select among is specified by a cell array of strings, and the 'Value' property indicates which of the strings is currently selected. A popup menu may be added to a GUI by using the popup menu tool in the Layout Editor.

• **List Boxes:** List boxes are graphical objects that display many lines of text and allow a user to select one or more of those lines. If there are more lines of text than can fit in the list box, scroll bar will be created to allow the user to scroll up and down within the list box. The lines of text that the user can select among are specified by a cell array of strings, and the 'Value' property indicates which of the strings are currently selected. A list box is created by creating a uicontrol whose style property is 'listbox'.

• **Slide Sliders:** Slide Sliders are graphical objects that allow a user to select values from a continuous range between a specified minimum value and a specified maximum value by moving a bar with a mouse. The 'Value' property of the slider is set to a value between min and max depending on the position of the slider.
8.2 Designing of User Interface

In view to provide better user interface the techniques developed user interface is developed in the MATLAB® the screen shot of the main window of the GUI developed is shown in the figure 8.2[5]:

![Application of Soft Computing Techniques for Computed Tomography](image)

**Figure 8.2: Main Window of the Developed Interface**

As seen from the figure the user interface contains five main menu namely documentation, existing techniques, Soft computing techniques , presentation and quit. Documentation menu contains two sub menu synopsis and thesis related to the research work, presentation contains two sub menus for presentation and the publication based on the research.
As seen it the module which is developed to implement existing reconstruction techniques which are described in the Chapter 5, these techniques are implemented to compare the results obtained with developed technique. The three technique implemented are simple back projection, convolution and back projection and filtration and back projection.
Figure 8.5: Window showing Implementation of Convolution Method

The input parameter required for implementing technique are the image resolution pre-defined phantom data and destination path were reconstructed image will be saved. The other techniques are implemented in the same manner.

Figure 8.6: Window showing sub menus in Developed Techniques
As seen from the figure 8.6 under the menu of soft computing techniques two sub menu are given namely Neural Network Reconstruction and Advance Neural Network which implements the techniques discussed in Chapter 6 and Chapter 7 respectively.

![Figure 8.7: Window showing selection of Beam Profile](image1)

The problem of reconstruction is solved for two beam profile Parallel beam and Fan beam. The user is able to select the type of the beam profile for which image is to be reconstructed.

![Figure 8.8: Window showing solving Parallel Beam Profile](image2)
The user needs to provide the image size i.e. resolution from which corresponding user defined phantom will be generated. Next step to perform back projection operation followed by creating the neural network. As the network needs to be trained this is followed by the reconstruction. At the end of the reconstruction process the two parameters Signal to Noise Ratio and Mean Square Error will be given to the user.

The module for the Advance Neural Network is developed on the similar line where user needs to provide the image resolution and destination path to save the reconstructed image. The next chapter discusses the performance assessment.

8.3 Concluding Remarks
This chapter gives the brief overview regarding the features of GUIDE the tool used to develop the user interface in MATLAB®. Chapter also discusses the features of the developed user interface for the research along with some of the screen shots of the windows of interface