



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

IMAGE PROCESSING USING MATLAB GUI

1. Onkar Rajendra Potdar. 2. Pradnya Prabhakar Powar. 3. Nikita Sarjerao Rabade. 4. Ritesh Dnyandev Rane. 5. Harshvardhan Vilas Sankpal. (Department of Electronics and Telecommunication Engineering, KITCoEK)

Guidance By

Prof. Varsha suryawanshi

ABSTRACT :

New technologies and the available computing tools are becoming more important every day in the teaching evolution. The use of Graphical User Interfaces (GUI) with MATLAB enables the implementation of practical teaching methodologies to make easier the comprehension of a given subject. In this work, we report on the application of GUI's in order to provide the students with a simple tool for a better understanding on how to design GRIN elements for optical systems. Another GUI's advantage is that they can be converted to an executable file, so any student could use the interface in their own computer without having a MATLAB license. We present a graphical interface to show the performance of an optical device for controlling beam size and for deflecting light for coupling purposes, by a simple geometrical optics study, in a tapered GRIN lens illuminated by a parallel beam of tilted rays. We also show a graphical interface to obtain the maximum coupling efficiency between fundamental modes of two single-mode fibers by a scaling operation carried out by a GRIN fibre lens. With this interface the students can vary the magnification and the image plane in order to get the more suitable GRIN fibre lens to maximize the coupling efficiency between two fibers.

KEYWORDS:

MATLAB, GUI(Graphical User Interfacing), RGB, Application, Functions, Callbacks, Processing, UI(User Interfacing), Histogram, Edge Detection.

1. AIMS AND OBJECTIVES :

After working through this worksheet, you should be familiar with:

- The basic operation of functions in MATLAB.
- Create and use MATLAB m-files effectively.
- Understand some basic MATLAB graphics functions.
- The basic construction of Graphical User Interfaces in MATLAB.
- How to call MATLAB Functions from a simple Graphical User Interface.

2. INTRODUCTION:

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 push buttons, list boxes, sliders, menus, and so forth. The GUI should behave 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 push button, the GUI should initiate the action described on the label of the button. This chapter introduces the basic elements of the MATLAB GUI's. The chapter does not contain a complete description of components or GUI features, but it does provide the basics required to create functional GUI's for your programs.

2.1 How a Graphical User Interface Works

A graphical user interface provides the user with a familiar environment in which to work. This environment contains push buttons, 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, GUI's 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:

- Components. Each item on a MATLAB GUI (push buttons, labels, edit boxes, etc.) is a graphical component. The types of components include graphical controls (push buttons, 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`.
- 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 we have plotted data. However, empty figures can be created with the function `figure` and can be used to hold any combination of components.
- 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 clicks 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 call back.

3 BASIC OPERATIONS OF IMAGE PROCESSING:

3.1 RGB to Grayscale:

Humans perceive colour through wavelength-sensitive sensory cells called cones. There are three different varieties of cones, each has a different sensitivity to electromagnetic radiation (light) of different wavelength. One cone is mainly sensitive to green light, one to red light, and one to blue light. By emitting a restricted combination of these three colours (red, green and blue), and hence stimulate the three types of cones at will, we are able to generate almost any detectable colour. This is the reason behind why colour images are often stored as three separate image matrices; one storing the amount of red (R) in each pixel, one the amount of green (G) and one the amount of blue (B). We call such colour images as stored in an RGB format. In grayscale images, however, we do not differentiate how much we emit of different colours, we emit the same amount in every channel. We will be able to differentiate the total amount of emitted light for each pixel; little light gives dark pixels and much light is perceived as bright pixels. When converting an RGB image to grayscale, we have to consider the RGB values for each pixel and make as output a single value reflecting the brightness of that pixel.

3.2 Histogram:

Histogram of an image represents the occurrence or frequency of a particular gray level. It is a graph depicting gray level intensities on x-axis and frequency of these intensities on y-axis. It gives an insight about the contrast in an image. It helps to categorize images. It provides image statistics for various techniques like thresholding, intensity slicing, segmentation. It helps to know whether the entire dynamic range of digitizer is used.

3.3 Edge Detection:

An edge is a set of connected pixels that lie on the boundary between two regions, where the brightness changes sharply. Edges bear a lot of information about underlying objects in the scene. Edge detection is applicable to binary images only [3]. Processing is reduced by considering just the edge elements instead of all pixels [4]. Edge detection is mainly used for feature extraction and detection. The magnitude of first derivative calculated within a neighbourhood around the pixel of interest are used to detect presence of edge in an image [2]. Gradient operators like Sobel, Prewitt and Canny are used for detecting the edge.

4. BENEFITS OF GUI AND IMAGE PROCESSING:

➤ **Advantages**

- Simplicity.
- It is visually appealing and makes anyone to get involved in working with the machine.
- Even a guy with no computer knowledge can use the computer and perform basic functions. GUI is responsible for that.
- Searching becomes very easy as GUI provides a visual representation of files present and provides details about it.
- Each and every response from the computer is visually communicated through GUI.
- A user with no computer knowledge can literally start learning about the machine because of GUI as it provides scope for users to explore and provides discover ability.
- If, for example, a user starts using a computer with no Interface, then he/she has to provide commands to the machine to execute each task. In a way, the user must have some kind of programming knowledge.

➤ **Disadvantages**

- One can only do what is already pre-programmed by some other developer.
- You cannot change the basic functionality of a system.
- It takes more power for the system to function.
- It is slow compared to simple command-based Interfaces.
- It consumes more memory space.
- GUI may be simple for a consumer but not as simple for the programmers who have to design and implement each and every function and apply abstraction to feel the advantages of GUI.
- If the functionality that the user needs is not present, then the user must know the commands that are necessary to proceed with the flow or else they are just stuck with it at the exact point.

5. GUI IMPLEMENTATION:

In our work we have developed a GUI using guide. The components used are text boxes, push button, pop-up menu, static text, axes. Push buttons generate an action when clicked. For example, an OK button might apply settings and close a dialog box. When you click a push button, it appears depressed; when you release the mouse button, the push button appears raised [6]. Static text controls display lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Users cannot change static text interactively. Pop-up menus open to display a list of choices when users click the arrow. Axes enable your UI to display graphics such as graphs and images. Like all graphics objects, axes have properties that you can set to control many aspects of its behaviour and appearance.

PROJECT CODE:

1. Upload Image

Use 'uploadimage' function.

command `uigetfile('image extension type')` is used for importing image in the MATLAB GUI. Read that file using command `imread()` and then display it with command `imshow()` on axes1 using `axes(handles.axes1)`. Now, with the command `setappdata()`, store the variable in the GUI so the variable will be accessible to one part of the GUI to the other part of the GUI.

```
a=uigetfile('.jpg');
a=imread(a);
axes(handles.axes1);
imshow(a);
setappdata(0,'a',a);
```

2. Convert RGB image into Grayscale

In the 'rgb2gray' function, copy and paste the below code to **convert the RGB image into grayscale** by using command rgb2gray().

```
a=getappdata(0,'a');  
agray=rgb2gray(a);  
axes(handles.axes1);  
imshow(agra);
```

3. Convert to Binary Image

In the 'im2bw' function, copy and paste the below code to **convert the image into binary image** or you can say in format of '0' (black) and '1' (white) by using command im2bw().

```
a=getappdata(0,'a');  
abw=im2bw(a);  
axes(handles.axes1);  
imshow(abw);
```

4. Reset to Original Image

In the 'reset' function, copy and paste the below code to **reset the edited image into the original image**.

```
a=getappdata(0,'a');  
axes(handles.axes1);  
imshow(a);
```

5. Plot Image Histogram

In the 'histogram' function, copy and paste the below code to **plot the histogram of the grayscale image** by using the command imhist('filename') and display it in on axes1. **For plotting the histogram you always have to convert the image into grayscale** and then you will be able to see the histogram of that graphic file.

```
a=getappdata(0,'a');  
ahist=a;  
ahist=rgb2gray(ahist);  
axes(handles.axes1);  
imhist(ahist);
```

6. Convert to Complement Image

In the 'complementimage' function, copy and paste the below code to see the **complement of the inserted graphic file** by using command imcomplement().

```
a=getappdata(0,'a');  
acomp=a;  
acomp=imcomplement(acomp);  
axes(handles.axes1);  
imshow(acomp);
```

7. Edge Detection using canny Method

In the 'edge' function, copy and paste the below code to detect and find edges in grayscale image by using command `edge('filename','method')`. In the place of method you can choose among these three, **Canny**, **Prewitt** and **montage**. We are using Canny method for edge detection. Also you cannot detect the edge directly from the original image, **first you have to convert it into grayscale** and then you can able to detect the edges

```
a=getappdata(0,'a');  
aedge=a;  
aedge=rgb2gray(aedge);  
aedge=edge(aedge,'Canny');  
axes(handles.axes1);  
imshow(aedge)
```

8. Rotate Image Clockwise

In the 'clockwise' function, copy and paste the below code to **rotate the image in the clockwise direction** by using command `imrotate(filename,'angle')`.

```
a=getappdata(0,'a');  
aclock=a;  
aclock=imrotate(aclock,270);  
axes(handles.axes1);  
imshow(aclock);
```

9. Rotate Image Anti-Clockwise

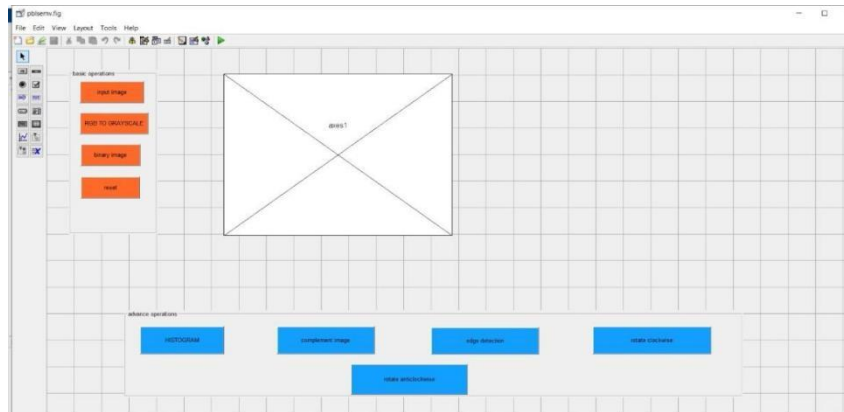
In the 'anticlockwise' function, copy and paste the below code to **rotate the image in the anti-clockwise direction** by using command `imrotate(filename,'angle')`.

```
a=getappdata(0,'a');  
aclock=a;  
aclock=imrotate(aclock,90);  
axes(handles.axes1);  
imshow(aclock);
```

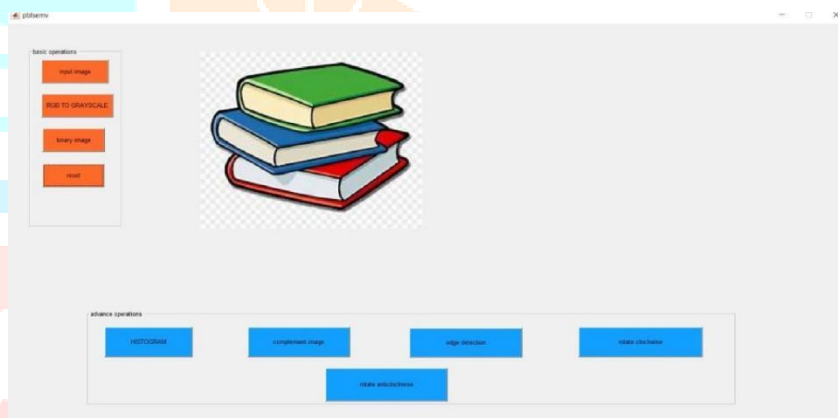
6. RESULTS:

In this section we present the result for various button press events. The text box displays the current operation performed.

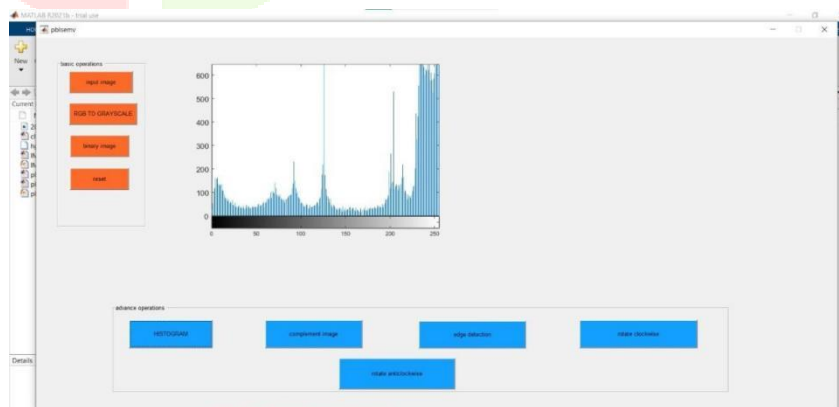
➤ GUI building



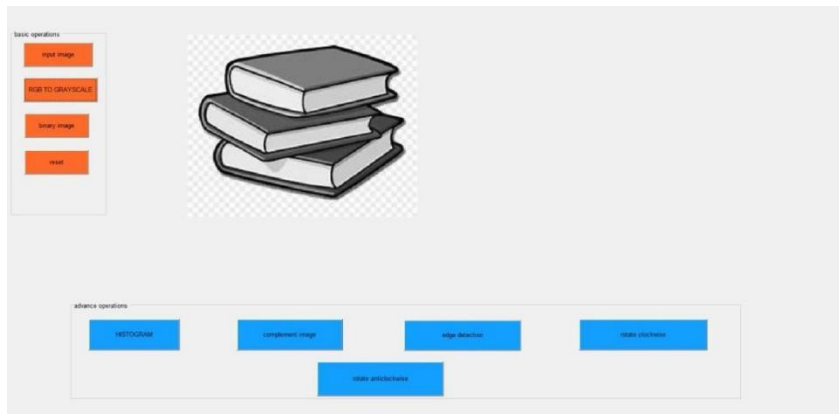
➤ Input Image



➤ Histogram



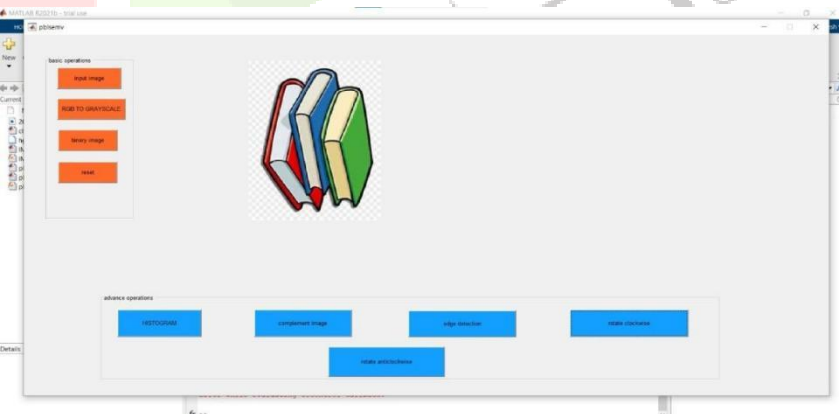
➤ RGB to Grayscale



➤ Rotate Anticlockwise



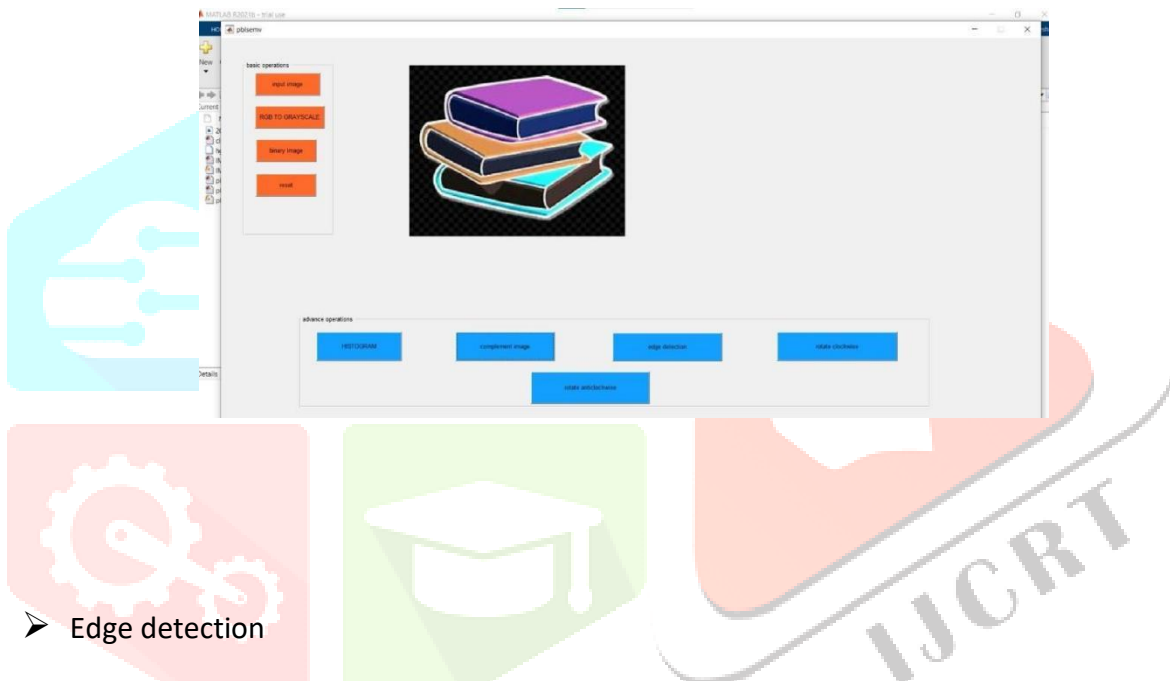
➤ Rotate Clockwise



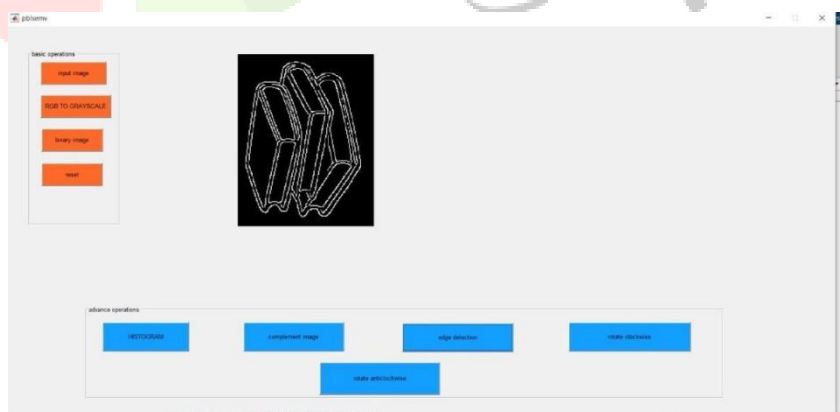
➤ Binary Image



➤ Complement Image



➤ Edge detection



7. CONCLUSION:

Basic important concepts of image processing are briefly presented in this paper. The GUI developed performs the basic operations on the medical images. It gives the user a better view about each operation at the click of the button. This GUI can be used for any general image. The same GUI can be used for other operations by altering the call-backs. Most of the functions presented in this paper can be further investigated and their performance can be improved. Future work aims at expanding the set of applications, calculating the statistical improvement after the application of the image processing techniques.

8. REFERENCES:

- <https://www.educba.com/what-is-gui/>
- https://en.wikipedia.org/wiki/Graphical_user_interface
- <http://ceur-ws.org/Vol-1837/paper6.pdf>
- <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-GUI-Interface.html>

