

Toolbox for material classification of MMWR and FLIR for Enhanced Vision System

Devendran.B*, Sudesh Kumar Kashyap**, T.V.Rama Murthy***

*PG Student, REVA Institute of Technology and Management, Bangalore, yashudevendran@gmail.com

**MSDF lab, FMCD, CSIR-National Aerospace Laboratories, Bangalore, India, Sudesh@nal.res.in

***Dept. of Electronics and Communication Engineering, REVA Institute of Technology and Management, Bangalore, drtvramamurthy@revainstitution.org

Abstract–To improve the situational awareness of an aircrew during poor visibility, different approaches emerged during the past few years. Enhanced vision systems (EVS – based on sensor images) are one of those. Typically, Enhanced vision systems concept is a combination of sensor data, environmental variables, internal and external state data and material database of the given geographical area. EVS uses weather penetrating forward looking image sensors such as Forward Looking Infrared Radar (FLIR) and HiVision Millimeter Wave Radar (HiVision MMWR). To generate the backscatter image from the imaging sensors, it is much important to develop a material classified database of the given geographical area. But development of such material classified database requires various image processing tasks. Hence the main contribution of this paper is the implementation of GUI based toolbox which is capable of developing material database for both MMWR and FLIR by using different material properties for Enhanced Vision Systems functionalities.

Keywords - Enhanced Vision Systems (EVS), Graphical User Interface (GUI), Region of Interest (ROI) and Phong like lighting model, Normalized Radar Cross Section (NRCS).

I. INTRODUCTION

Typical Enhanced vision Systems concept is shown in fig.1. The performance of the Enhanced Vision System relies on the performance of the imaging sensors [1]. The reliability of the Imaging sensor highly depends on the accuracy of used material classified database, also imaging sensors generates backscatter image by taking the material classified database of the given geographical area as a reference [2]. Hence designing of such database that is accurate is a significant challenge and it requires number of image processing functions to classify the objects in the given image.

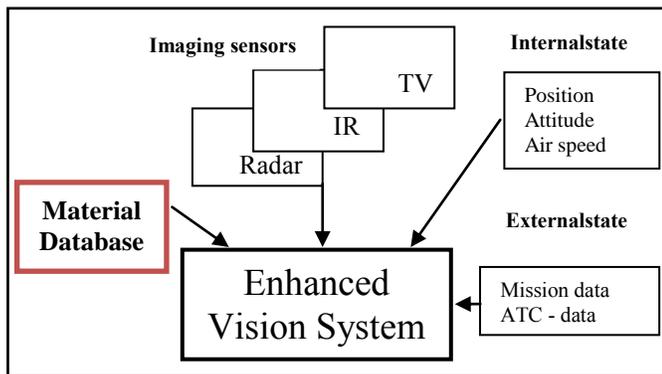


Fig.1. Typical Enhanced Vision System concept [3]

As mentioned earlier sensor vision is one of the integral part of Enhanced vision systems technology. Sensor vision

uses weather penetrating Forward Looking Infrared Radar (FLIR) and MilliMeter Wave Radar (MMWR) sensors for imaging. In case of MMWR, imaging is based on Backscattering Coefficient (Normalized Radar Cross Section (NRCS) σ_0). Whereas for FLIR, Solar absorptance, Albedo, Emissivity, Conductivity, Thickness, Slope and Surface azimuth parameters are to be consider. More details about radar modelling requirements are obtained from ref [1-4].

For a given geographical area(airport image)the classifications of terrain are: (1) Asphalt or Bitumen, (2) Concrete, (3) Tar, (4) Bare Soil, (5) Green Grass, (6) Trees, (7) Desert Sand, (8) Ice or White paint, (9) Snow, (10) Water, (11) Brick or Urban areas, (12) Wood. Material classification has to be done with respect to the above terrain types. Material coded airport database can be subsequently used in modelling and simulation of MMWR and FLIR for the purposes of EVS studies. Material classification of an airport image is purely an image processing task, which requires several algorithms to be applied on an image at a time. Material classification can also be done by using edge detection and colour coding methods. But the results of these methods will not meet the desired accuracy. Hence this drawback motivates to develop a software toolbox to create material database based on different aforementioned parameters.

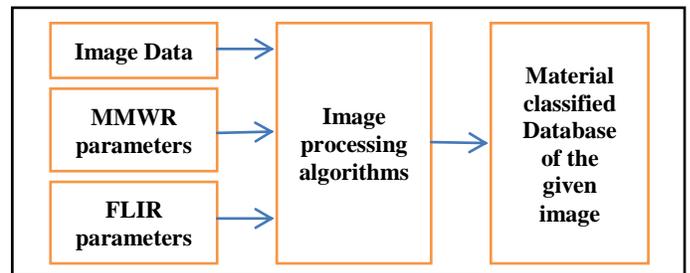


Fig.2. Block diagram for toolbox design

In this paper details of implementation of MATLAB based GUI toolbox for material classification and realization of Phong lighting model for different terrain materials based on Normalized Radar Cross Section are presented. However, the authors have attempted to present complete procedure required for developing material database by using the designed toolbox.

II. TOOLBOX DESIGN USING MATLAB GRAPHICAL USER INTERFACE DEVELOPMENT ENVIRONMENT (GUIDE)

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called *components*, which enable a user to perform interactive tasks. The user of

the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed. GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders just to name a few. GUIs created using MATLAB® tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots [8].

Most GUIs wait for their user to manipulate a control, and then respond to each action in turn. Each control, and the GUI itself, has one or more user-written routines (executable MATLAB code) known as *callbacks*, named for the fact that they “call back” to MATLAB to ask it to do things. The execution of each callback is triggered by a particular user action such as pressing a screen button, clicking a mouse button, selecting a menu item, typing a string or a numeric value, or passing the cursor over a component. The GUI then responds to these *events*. The creator of the GUI, provides callbacks which define what the components do to handle events. This kind of programming is often referred to as *event-driven* programming. In the example, a button click is one such event. In event-driven programming, callback execution is *asynchronous* [6-8].

MATLAB GUIs can be built in two ways:

- Use of GUIDE (GUI Development Environment), an interactive GUI construction kit.
- Create code files that generate GUIs as functions or scripts (programmatically GUI construction).

In this paper the first approach is selected and it starts with a figure that you populate with components from within a graphic layout editor. GUIDE creates associated code file containing callbacks for the GUI and its components. GUIDE saves both the figure (as a FIG-file) and the code file. Opening either one also opens the other to run the GUI. [8] Initial view of material coding toolbox is shown in fig.3.

Important callbacks used in designing toolbox

Some of the very important callback functions which are used in designing material coding toolbox are discussed as follows. To write the other callback functions which are used in toolbox, ref [6-8] are used as a guide.

- **Load_Image_Callback;** by using this callback function one can get any image from any of the directory. Once the image is selected, it will be displayed on main axes. Matlab functions used for this purpose are described as follows.

```
[h,path]=uigetfile({'*.jpg;*.tif;*.png;*.gif',
'All Image Files';...
'*.','All Files'},'mytitle',...
'..:');
handles.a = imread (strcat (path,h));
axes (handles.axes3)
set (gcf,'CurrentAxes',handles.axes3)
set (handles.Original_img, 'string',h)
```

```
imshow (handles.a)
```

- **Select_ROI_Callback;** this callback function can be called as a heart of the designed toolbox. Because this function provides facility for user to select the region of his own interest. The MATLAB function used for this task is described as follows,

```
I = image data;
[BW, x, y] = myroiopoly (I);
```

The above function gives the selected region of interest’s binary mask (BW) and (x, y) co-ordinates of that mask. A region of interest (ROI) is a portion of an image that you want to filter or perform some other operation on. It is possible to define an ROI by creating a binary mask, which is a binary image that is the same size as the image required to process with pixels that define the ROI set to 1 and all other pixels set to 0. By using above callback function one can select more than one ROI in an image.

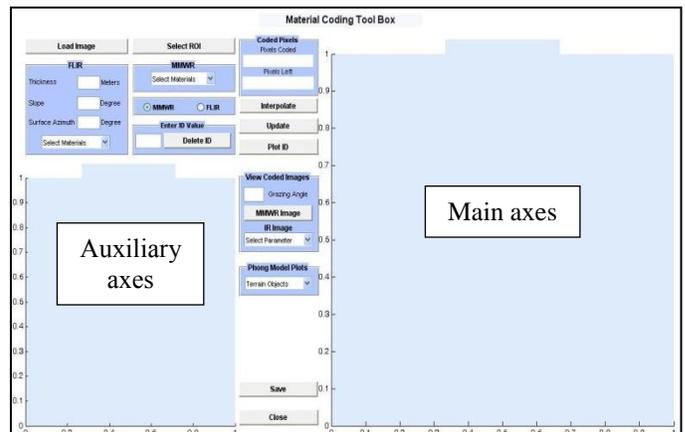


Fig.3. Initial view of the designed material coding toolbox using GUIDE

- **Delete_ID_Callback;** this callback function facilitates user to delete the erroneous region of interest as well as its related data such as (x, y) co-ordinates, ID values, material coding parameters. The following code is used to delete the unwanted data,

```
del_file = load (file_name);
del_file.material.XY (ID_entry) = [];
del_file.material.sigma (ID_entry) = [];
del_file.material.ID (ID_entry) = [];
```

The following —function is used to get back the region of interest and its co-ordinates which user likes to delete.

```
Mask = poly2mask ();
```

- **Select_Materials_Callback;** this callback function is used to select the required terrain material corresponds to selected region of interest. The following code is used to select the different materials in both MMWR and FLIR case of material classification.

```
Select_Materials_Callback (hObject, eventdata, handles)
```

```
v = get(handles.Select_Materials,'value');
s = get(handles.Select_Materials,'string');
handles.sig = importdata('Phong.txt');
Switch s {v}

Case 'Asphalt/Bitumen'
Case ' '

end
```

- **Save_Callback;** this callback function is used to save the material database for both MMWR and FLIR. Material database is saved in a structure along with the number of ID's, (x, y) co-ordinates of each ID, in sigma field either NRCS calculation parameters in case of MMWR or IR material classification parameters for FLIR case should be included for each ID selected and it is implemented as follows.

```
Save_Callback(hObject, eventdata, handles)
material.sigma = handles.mycm;
material.mdb = handles.matdb;
material.ID = handles.myid;
material.XY = handles.Coordinates;
handles.material = material;
[handles.material,path]=uiputfile({'*.m;*.fig;
*.mat;*.mdl','MATLAB Files (*.m,*.mat)';
'*.m', 'program files (*.m)';
'*.mat','MAT-files (*.mat)';
'*.*','All Files (*.*)'}, 'Save as'...
'..\project\MMWR_DataBase\...');
Save ([path, handles.material]);
guidata(hObject, handles)
```

Operating procedure of the Material classification Toolbox/ GUI Model

The procedural steps for developing database of both MMWR and FLIR using the toolbox designed with the help of above discussed callbacks are described as follows,

Step 1:

Run the program file (Material_Coding_Tool_Box.m file) on the MATLAB editor platform.

Step 2:

A GUI model appears on the screen. To start material classification click on the 'Load Image' button. It opens the folder where the images are stored; select the required image by double clicking on that. It will display the selected image on the main axes.

Step 3:

MMWR case: Move to MMWR panel, click on the 'Select Materials' popup menu and select the required material type.

FLIR case: Move to FLIR panel, enter the Thickness, Slope and Surface Azimuth values of the required material in the respective editor windows. If material classification is independent of these three parameters then enter 'zero' and click on the 'Select Materials' popup menu and select the required material type.

Step 4:

Click on the 'Select ROI' button, It works on the graphical input method. A cursor '+' appears on the main axes

where the required image was loaded, it moves as the mouse pointer moves. Move this cursor on the desired material type in the image and mark the points on its boundary. Once the last point meets the first point, the '+' cursor changes to 'o' cursor. After completion of boundary  shape cursor will appear, complete the Region of interest by double clicking on it. The boundary will become red color with number tag. Number tag is the count of number of ROI's created and it is placed at the first coordinates of the selected region.

Step 5:

Repeat step 3 and step 4 for different materials in the same image. After all the region of interest has been selected properly in an image tile, click on 'Interpolate' button. It will assign the un-coded region as Green grass and completes the database of the given image tile.

Step 6:

Click on 'Save' button to save the complete material coded database of that image in '.mat' format. The database file name should be same as the respective image file name which is loaded on main axes.

Step 7:

If the user wishes to see the coded regions, click on 'Update' button. It will display the coded regions on the auxiliary axes of respective image. Also displays the information about how many pixels are coded and how many are left out in the slot given in the tool box.

Step 8:

Suppose the region of interest is erroneously coded or the user is not satisfied with the coded region, in this case user can delete that particular region. First select whether it is in FLIR mode or MMWR mode by clicking on the radio buttons (By default it will take MMWR mode), then move to 'Enter ID Value' panel, enter the number tag of the region would like to delete in the editor window, then click on 'Delete ID' button. This function will clear all the data related to that particular region.

Step 9:

Suppose user stopped the material classification at some stage and he wanted to continue with the previously coded data of the same image. To continue the material classification of the previous session it is required to load the previously selected region of interests in order to avoid repetitive coding. This task is accomplished as follows; first select the appropriate mode (i.e. MMWR or FLIR mode by clicking radio button) then click on 'Plot ID' button. This will plot all the previously selected regions of the image on to the main axes and it loads the previously coded data on to the workspace.

Step 10:

If the user wants to see the coded image with respect to different parameters which are considered for material classification of both MWR and FLIR, first select MMWR or FLIR mode then move to View Coded Images panel,

MMWR case: Enter the grazing angle at which user likes to see the image and click on 'MMWR Image' button. It will display the coded image with respect to the NRCS and grazing angle on the auxiliary axes.

FLIR case: Click on the ‘Select Parameters’ popup menu and select the required parameter for which user likes to see the coded version of the original image tile.

Step 11:

To realize the Phong model plots for different terrain materials, move to Phong Model Plots panel then click on the ‘Terrain objects’ popup menu and select the required material for which user likes to see the Phong model plot.

Step 12:

To close the GUI click on the ‘Close’ button.

Phong like lighting backscatter (NRCS) generator model implementation for MMWR

Radar simulation involves the computation of a radar response based on the terrain’s normalized radar cross section (RCS). MMWR imaging depends on backscattering coefficient. The amount of the radiated energy is proportional to the target size, orientation, physical shape and material which are all lumped together in one target-specific parameter called Radar Cross Section (RCS) denoted by “ σ ”. The radar cross section is defined as the ratio of the power reflected back to the radar to the power density incident on the target.

$$\sigma = \frac{P_r}{P_d} (m^2)$$

Where, P_d is power delivered to the radar signal processor by the antenna. Radar simulation involves the computation of a radar response based on the terrain’s normalized radar cross-section (NRCS). To compute normalized radar cross section for different types of terrain objects, we are using a well-known model called Phong like lighting model. Phong lighting is an empirically derived BRDF model for the computation of optical reflections [5]. The method is very popular in computer graphics and is broadly supported by different software and hardware platforms. Although the Phong lighting model is not physically correct since it does not obey all the laws of physics involved, it has easily interpretable parameters which may explain its popularity. Using the Phong model we compute the mean normalized radar cross section as,

$$\sigma^0 = a \sin \theta + b \sin^c \theta$$

Where a , b and c are the model parameters, a controls the amount of diffuse reflection of a material, b is the specular reflection coefficient and c is the specularity, that is, the sharpness of the directional highlight for a material (see [5] for more details about terrain types and a , b and c values).

III. SIMULATION RESULTS

Toolbox Views

In this section views of the designed toolbox at different stages of material classification and Phong model plots of different materials are presented in figures 4, 5 and 6. The parameters used for developing the Database of both MMWR and FLIR are obtained through Literature survey. In MMWR case toolbox facilitates user to see the coded image with respect to grazing angle. Whereas, FLIR coded images can be seen with respect to different parameters.

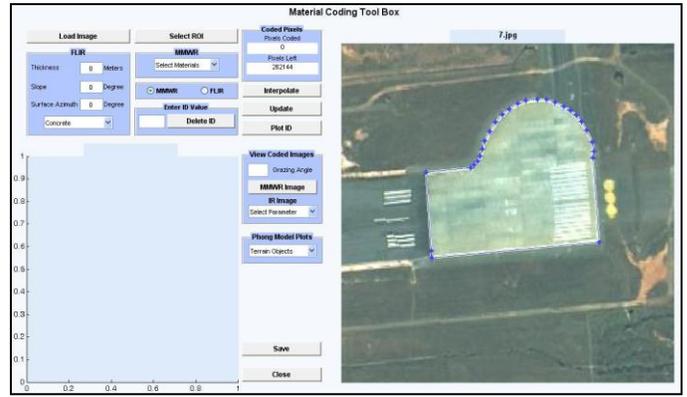


Fig.4. Demo of selecting a ROI using toolbox

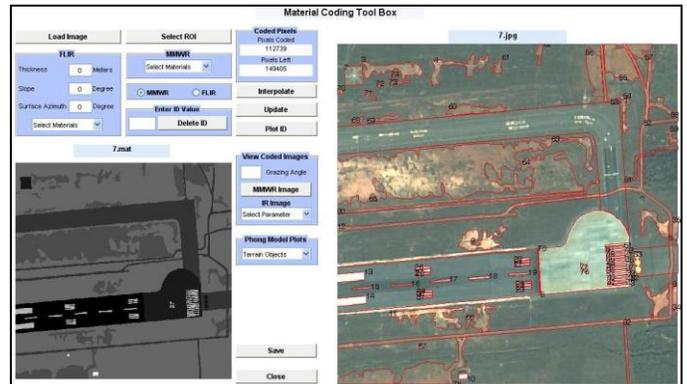
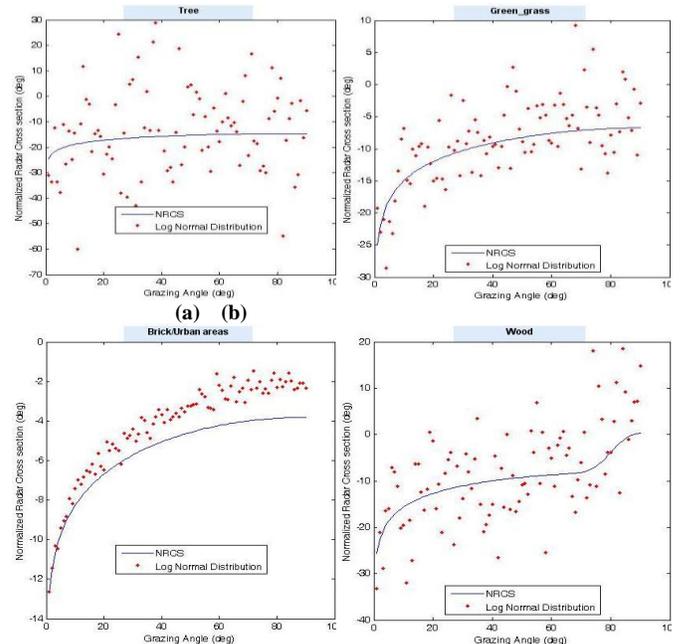


Fig.5. Toolbox looks after selecting all the ROI’s of a given image tile and updated the coded image on to the auxiliary axes.



(c)(d) Phong model plots of different terrain materials implemented on toolbox; (a) Tree (b) Green Grass (c) Brick/Urban areas (d) Wood

Results of Database Comparison

In this section comparison of databases obtained from other methods and toolbox is made. Fig 9 shows the comparison of FLIR Database and fig 10 shows MMWR Database comparison. Figures from 7 to 11 also show the accuracy of the Database obtained using toolbox.



Fig.7. Image of the material classified database developed using color coding method



Fig.8. Image of the material classified database developed using our designed toolbox

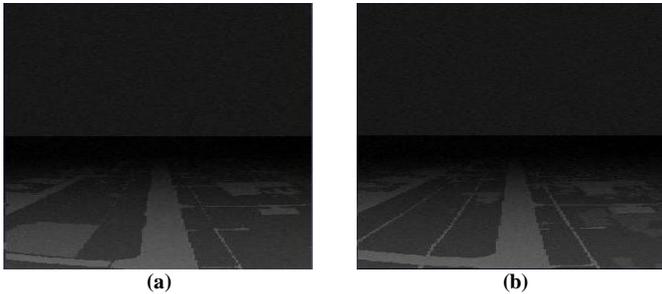


Fig.9. Comparison of FLIR Images generated using two different databases. (a) Image generated using Database created outside the toolbox; (b) Image generated using Database created by toolbox.

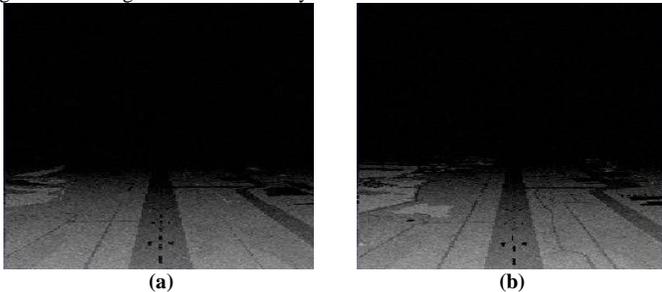


Fig.10. Comparison of MMWR Images generated using two different databases. (a) Image generated using Database created outside the toolbox; (b) Image generated using Database created by toolbox.

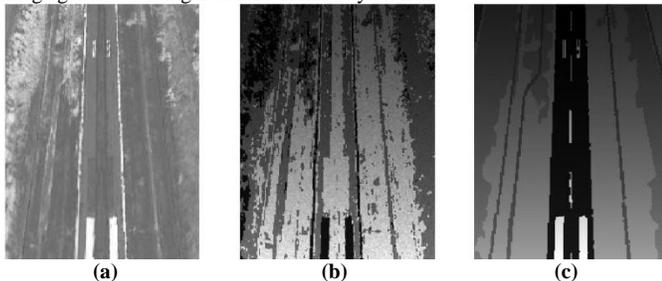


Fig.11. Comparison of MMWR Images generated using two databases developed from color coding method and toolbox. (a) Original image (b) Image generated using Database created using color coding methods; (c) Image generated using Database created by toolbox.

IV. CONCLUSION

Weather penetrating sensors play a prominent role in Enhanced Vision Systems functionalities. Backscatter image

generation capabilities of these sensors are very much dependent on Material database of the given geographical area. Quality of database reflects the quality of the backscatter image from sensor. Hence, Toolbox implementation described in this paper that can provide the facility for user to develop the material Database for both HiVision MilliMeter wave radar and Forward looking Infrared Radar. The procedure for developing the Database using toolbox is presented. The material Database obtained from the designed toolbox is compared with the one obtained from colour coding method. Also comparison of quality of the Database in the simulator also presented. But, to develop the quality material Database much user attention is required.

ACKNOWLEDGMENT

This work is sponsored by National Aerospace Laboratories, Bangalore. Programmatic support and technical advice from MSDF group of FMCD, NAL is gratefully appreciated. Gratitude also owed to director of NAL, head of FMCD and Principal of REVA ITM Bangalore.

REFERENCES

- [1] Maxime E. Bonjeana, Fabian D. Lapiereb, Jens Schiefelec, Jacques G. Verly, *Flight Simulator with IR and MMW Radar Image Generation Capabilities*, Proc of SPIE, Volume 6226, pp 62260A 8-12, April 2006.
- [2] Bernd Korn, Hans-Ullrich Doehler, and Peter Hecker, Institute of Flight Guidance, "*Weather Independent Flight Guidance: Analysis of MMW Radar Images for Approach and Landing*", 0-7695-0750-6/00 IEEE trans 2000
- [3] B. Korn, H.-U. Doehler and P. Hecker. "*MMW radar data processing for enhanced vision*". In J. G. Verly, editor, *Enhanced and Synthetic Vision 1999*, volume 3691, pages 29-38. SPIE, Apr. 1999.
- [4] H.U.Dohler, P.Hecker, R.Rodolff, "*Image data fusion for future Enhanced Vision Systems* "; *Sensor Data Fusion and Integration of the Human Element, system Concepts and Integration (SCI) Symposium*, Ottawa, 14-17 sep. 98, RTO Meeting Proceedings 12
- [5] Niklas Peinecke, Hans-Ullrich Doehler, and Bernd R. Korn, "*Phong-like Lighting for MMW Radar Simulation*" Millimeter Wave and Terahertz Sensors and Technology, edited by Keith A. Krapels, Neil A. Salmon, Proc. of SPIE Vol. 7117, 71170M · © 2008 SPIE
- [6] Patrick Marchand and O. Thomas Holland. "*Graphics and GUIs with MATLAB*"
- [7] Hunt Lipsman & Rosenberg. "*A Guide to MATLAB for Beginners and Experienced Users*"
- [8] Matlab learning guide from Mathworks.com , "*BuildGUI in Matlab*"