A Mini Project Report on

Underwater Image Enhancement Method for SLAM of Autonomous Underwater Vehicle

Submitted in partial fulfilment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

In

Electronics & Communication Engineering

Submitted

By

KUSHAL POOPPAL - 17K81A04L8
MOSALI AMITH REDDY - 17K81A04N0
NIKHIL MISHRA - 17K81A04N4

Under the guidance of

Ms P. PUSHPA, MTech

Assistant Professor

DEPARTMENT OF ELECTRONICS &COMMUNICATION ENGINEERING

St. MARTIN'S ENGINEERING COLLEGE

UGC AUTONOMOUS

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

Dhulapally, Secunderabad-500 100

NBA & NAAC A+ Accredited

2020-2021

St. MARTIN'S ENGINEERING COLLEGE

UGC AUTONOMOUS

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

Dhulapally, Secunderabad-500 100

NBA & NAAC A+ Accredited

2020-2021

Department Of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the mini-project work entitled "**UNDERWATER IMAGE ENHANCEMENT METHOD FOR SLAM OF AUTONOMOUS UNDERWATER VEHICLE**" is a bonafide work carried out by **KUSHAL POOPPAL (17K81A04L8); MOSALI AMITH REDDY (17K81A04N0)**; **NIKHIL MISHRA (17K81A04N4)** in partial fulfilment of the requirements for the degree of **Bachelor of Technology in Electronics & Communication Engineering** by the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2020-2021.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree.

INTERNAL GUIDE	**HEAD OF THE DEPARTMENT**
Ms. P. PUSHPA	**Dr. B. HARI KRISHNA**
Assistant Professor	**Professor**

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We sincerely thank the management of our college **St. MARTIN’S ENGINEERING COLLEGE**, for providing required facilities during our project work.

We derive our great pleasure in expressing our sincere gratitude to our respected Principal **Dr. P. SANTOSH KUMAR PATRA** for his timely suggestions, which helped us to complete the project work successfully.

It is the very auspicious moment we would like to express our gratitude to **Dr. B. HARI KRISHNA**, Head of the Department, ECE for his consistent encouragement during the progress of this project.

We sincerely thank our overall project coordinator, **Mr. A. Anand**, department of ECE for guidance and encouragement in carrying out this project work.

We take it as privilege to thank our project coordinator **Ms. P. PUSHPA**, Assistant Professor, department of ECE for the ideas that led to complete the project work and we also thank them for their continuous guidance, support and unfailing patience, throughout the course of this work.

KUSHAL POOPPAL	**(17K81A04L8)**
MOSALI AMITH REDDY	**(17K81A04N0)**
NIKHIL MISHRA	**(17K81A04N4)**

DECLARATION

We the students of **‘Bachelor of Technology in Department of Electronics and Communication Engineering’**, Session: 2020-2021, **St. Martin’s Engineering College**, hereby declare that the project work entitled **“UNDERWATER IMAGE ENHANCEMENT METHOD FOR SLAM OF AUTONOMOUS UNDERWATER VEHICLE**” is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied is this project report has not been submitted in any university for award of any degree.

KUSHAL POOPPAL	**MOSALI AMITH REDDY**	**NIKHIL MISHRA**
(17K81A04L8)	**(17K81A04N0)**	**(17K81A04N4)**

ABSTRACT

In the recent days, wide range of research has been carried out on visual enhancement of under images in submarine and military operations to discover submerged structural designing and sea floor exploration. But, diving in the deep ocean for a long time has increased the difficulties for analysis of underwater images. Further, other factors such as scattering resulting from presence of particles inside the water and blurring effects reduce the quality of images being captured by underwater optic camera. There are several algorithms have been introduced to improve the visual quality of deep-water images. Therefore, in this project, a novel algorithm based on bidirectional Empirical Mode Decomposition (BEMD) to enhance the visual quality of the underwater images will be implemented and comparison of data with conventional enhancement technique will be illustrated. The implementation will be done using MATLAB software.

INDEX

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO MATLAB:

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

- Math and computation
- Algorithm development
- Data acquisition
- Modelling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard.

Instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis. MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

1.2 THE MATLAB SYSTEM

The MATLAB system consists of five main parts

- **Development Environment**:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

- **The MATLAB Mathematical Function Library**:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

- **The MATLAB Language**

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both “programming in the small” to rapidly create quick and dirty throw-away programs, and “programming in the large” to create large and complex application programs.

- **Graphics:**

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

- **The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using **IMAGE PROCESSING** toolbox.

1.3 GRAPHICAL USER INTERFACE DEVELOPMENT ENVIRONMENT

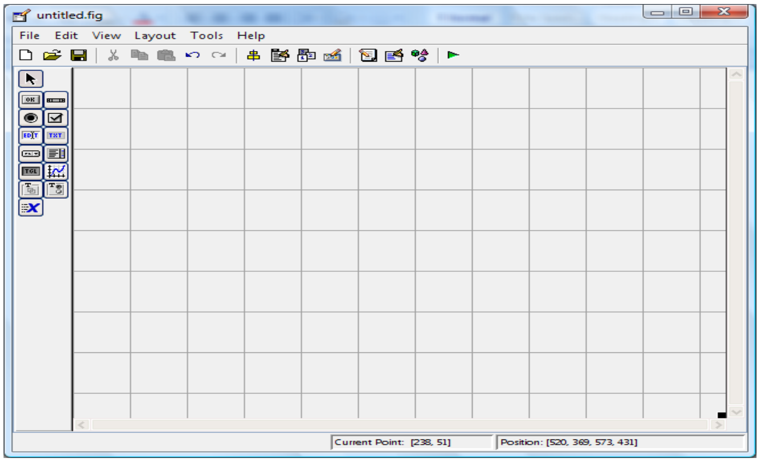
MATLAB’s Graphical User Interface Development Environment (GUIDE) provides a rich set of tools for incorporating graphical user interfaces (GUIs) in M-functions. Using GUIDE, the processes of laying out a GUI (i.e., its buttons, pop-up menus, etc.) and programming the operation of the GUI are divided conveniently into two easily managed and relatively independent tasks. The resulting graphical M-function is composed of two identically named (ignoring extensions) files:

- A file with extension. fig, called a FIG-file that contains a complete graphical description of all the function’s GUI objects or elements and their spatial arrangement. A FIG-file contains binary data that does not need to be parsed when he associated GUI-based M-function is executed.

- A file with extension .m, called a GUI M-file, which contains the code that controls the GUI operation. This file includes functions that are called when the GUI is launched and exited, and callback functions that are executed when a user interacts with GUI objects for example, when a button is pushed.

To launch GUIDE from the MATLAB command window, type

guide filename. Where filename is the name of an existing FIG-file on the current path. If filename is omitted, GUIDE opens a new (i.e., blank) window.



Current Point: [238, 51]

Position: [520, 369, 573, 431]

Figure 1.3 Graphical User Interface Image

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components that 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.

1.4 DEVELOPMENT ENVIRONMENT

1.4.1 Introduction

This chapter provides a brief introduction to starting and quitting MATLAB, and the tools and functions that help you to work with MATLAB variables and files. For more information about the topics covered here, see the corresponding topics under Development Environment in the MATLAB documentation, which is available online as well as in print.

STARTING AND QUITTING MATLAB

1.4.2 Starting MATLAB

On a Microsoft Windows platform, to start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop. On a UNIX platform, to start MATLAB, type matlab at the operating system prompt. After starting MATLAB, the MATLAB desktop opens - see MATLAB Desktop.

You can change the directory in which MATLAB starts, define start-up options including running a script upon start-up, and reduce start-up time in some situations.

1.4.3 Quitting MATLAB

To end your MATLAB session, select Exit MATLAB from the File menu in the desktop, or type quit in the Command Window. To execute specified functions each time MATLAB quits, such as saving the workspace, you can create and run a finish.m script.

1.4.4 MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB. The first time MATLAB starts, the desktop appears as shown in the following illustration, although your Launch Pad may contain different entries.

You can change the way your desktop looks by opening, closing, moving, and resizing the tools in it. You can also move tools outside of the desktop or return them back inside the desktop (docking). All the desktop tools provide common features such as context menus and keyboard shortcuts.

You can specify certain characteristics for the desktop tools by selecting Preferences from the File menu. For example, you can specify the font characteristics for Command Window text. For more information, click the Help button in the Preferences dialog box.

All the desktop tools provide common features such as context menus and keyboard shortcuts. For more information about the topics covered here, see the corresponding topics.

1.4.5 Desktop Tools

This section provides an introduction to MATLAB's desktop tools. You can also use MATLAB functions to perform most of the features found in the desktop tools. The tools are:

- Current Directory Browser
- Workspace Browser
- Array Editor
- Editor/Debugger
- Command Window
- Command History
- Launch Pad
- Help Browser

Command Window

Use the Command Window to enter variables and run functions and M-files.

Command History

Lines you enter in the Command Window are logged in the Command History window. In the Command History, you can view previously used functions, and copy and execute selected lines. To save the input and output from a MATLAB session to a file, use the diary function.

Running External Programs

You can run external programs from the MATLAB Command Window. The exclamation point character! is a shell escape and indicates that the rest of the input line is a command to the operating system. This is useful for invoking utilities or running other programs without quitting MATLAB. On Linux, for example,!emacs magik.m invokes an editor called emacs for a file named magik.m. When you quit the external program, the operating system returns control to MATLAB.

Launch Pad

MATLAB's Launch Pad provides easy access to tools, demos, and documentation.

Help Browser

Use the Help browser to search and view documentation for all your Math Works products. The Help browser is a Web browser integrated into the MATLAB desktop that displays HTML documents.

To open the Help browser, click the help button in the toolbar, or type help browser in the Command Window. The Help browser consists of two panes, the Help Navigator, which you use to find information, and the display pane, where you view the information.

Help Navigator

Use to Help Navigator to find information. It includes:

Product filter - Set the filter to show documentation only for the products you specify.

Contents tab - View the titles and tables of contents of documentation for your products.

Index tab - Find specific index entries (selected keywords) in the MathWorks documentation for your products.

Search tab - Look for a specific phrase in the documentation. To get help for a specific function, set the Search type to Function Name.

Favourites tab - View a list of documents you previously designated as favourites.

Display Pane

After finding documentation using the Help Navigator, view it in the display pane. While viewing the documentation, you can:

Browse to other pages - Use the arrows at the tops and bottoms of the pages, or use the back and forward buttons in the toolbar.

Bookmark pages - Click the Add to Favorites button in the toolbar.

Print pages - Click the print button in the toolbar.

Find a term in the page - Type a term in the Find in page field in the toolbar and click Go.

Other features available in the display pane are: copying information, evaluating a selection, and viewing Web pages.

Current Directory Browser

MATLAB file operations use the current directory and the search path as reference points. Any file you want to run must either be in the current directory or on the search path.

Search Path

To determine how to execute functions you call, MATLAB uses a search path to find M-files and other MATLAB-related files, which are organized in directories on your file system. Any file you want to run in MATLAB must reside in the current directory or in a directory that is on the search path. By default, the files supplied with MATLAB and MathWorks toolboxes are included in the search path.

Workspace Browser

The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory. You add variables to the workspace by using functions, running M-files, and loading saved workspaces.

To view the workspace and information about each variable, use the Workspace browser, or use the functions who and whos.

To delete variables from the workspace, select the variable and select Delete from the Edit menu. Alternatively, use the clear function.

The workspace is not maintained after you end the MATLAB session. To save the workspace to a file that can be read during a later MATLAB session, select Save Workspace As from the File menu, or use the save function. This saves the workspace to a binary file called a MAT-file, which has a .mat extension. There are options for saving to different formats. To read in a MAT-file, select Import Data from the File menu, or use the load function.

ArrayEditor

Double-click on a variable in the Workspace browser to see it in the Array Editor. Use the Array Editor to view and edit a visual representation of one- or two-dimensional numeric arrays, strings, and cell arrays of strings that are in the workspace.

Editor/Debugger Use the Editor/Debugger to create and debug M-files, which are programs you write to runMATLAB functions. The Editor/Debugger provides a graphical user interface for basic text editing, as well as for M-file debugging.

You can use any text editor to create M-files, such as Emacs, and can use preferences (accessible from the desktop File menu) to specify that editor as the default. If you use another editor, you can still use the MATLAB Editor/Debugger for debugging, or you can use debugging functions, such as dB stop, which sets a breakpoint.

1.4.6 Functions

MATLAB provides a large number of standard elementary mathematical functions, including abs, sqrt, exp, and sin. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type help elfun, For a list of more advanced mathematical and matrix functions, type help specfun help elmat

Some of the functions, like sqrt and sin, are built-in. They are part of the MATLAB core so they are very efficient, but the computational details are not readily accessible. Other functions, like gamma and sinh, are implemented in M-files. You can see the code and even modify it if you want. Several special functions provide values of useful constants.

Pi	3.14159265...
I	Imaginary unit, $\sqrt{-1}$
I	Same as i
Eps	Floating-point relative precision, 2^{-52}
Realmin	Smallest floating-point number, 2^{-1022}
Realmax	Largest floating-point number, $(2- \varepsilon)2^{1023}$
Inf	Infinity
NaN	Not-a-number

1.5 GRAPHICAL UNIT INTERFACE(GUI)

A graphical user interface (GUI) is a user interface built with graphical objects, such as buttons, text fields, sliders, and menus. In general, these objects already have meanings to most computer users. For example, when you move a slider, a value changes; when you press an OK button, your settings are applied and the dialog box is dismissed. Of course, to leverage this built-in familiarity, you must be consistent in how you use the various GUI-building components.

Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how they work. The action that results from a particular user action can be made clear by the design of the interface.

The sections that follow describe how to create GUIs with MATLAB. This includes laying out the components, programming them to do specific things in response to user actions, and saving and launching the GUI; in other words, the mechanics of creating GUIs. This documentation does not attempt to cover the "art" of good user interface design, which is an entire field unto itself. Topics covered in this section include:

1.5.1 Creating GUIs with GUIDE

MATLAB implements GUIs as figure windows containing various styles of uicontrol objects. You must program each object to perform the intended action when activated by the user of the GUI. In addition, you must be able to save and launch your GUI. All of these tasks are simplified by GUIDE, MATLAB's graphical user interface development environment.

1.5.2 GUI Development Environment

The process of implementing a GUI involves two basic task.

- Laying out the GUI components
- Programming the GUI components

GUIDE primarily is a set of layout tools. However, GUIDE also generates an M-file that contains code to handle the initialization and launching of the GUI. This M-file provides a framework for the implementation of the callbacks - the functions that execute when users activate components in the GUI.

The Implementation of a GUI

While it is possible to write an M-file that contains all the commands to lay out a GUI, it is easier to use GUIDE to lay out the components interactively and to generate two files that save and launch the GUI:

A FIG-file - contains a complete description of the GUI figure and all of its

children (uicontrols and axes), as well as the values of all object properties.

An M-file - contains the functions that launch and control the GUI and the

callbacks, which are defined as subfunctions. This M-file is referred to as the

application M-file in this documentation.

Note that the application M-file does not contain the code that lays out the uicontrols; this information is saved in the FIG-file.

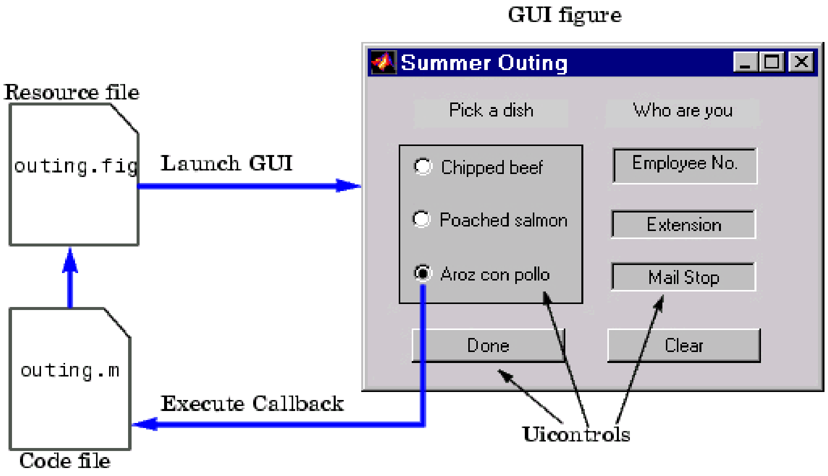


Figure 1.5.2 Graphical User Blocks

1.5.3 Features of the GUIDE-Generated Application M-Fil

GUIDE simplifies the creation of GUI applications by automatically generating an M-file framework directly from your layout. You can then use this framework to code your application M-file. This approach provides a number of advantages:

The M-file contains code to implement a number of useful features (see Configuring Application Options for information on these features). The M-file adopts an effective approach to managing object handles and executing callback routines (see Creating and Storing the Object Handle Structure for more information). The M-files provides a way to manage global data (see Managing GUI Data for more information).

The automatically inserted subfunction prototypes for callbacks ensure compatibility with future releases. For more information, see Generating Callback Function Prototypes for information on syntax and arguments.

You can elect to have GUIDE generate only the FIG-file and write the application M-file yourself. Keep in mind that there are no uicontrol creation commands in the application M-file; the layout information is contained in the FIG-file generated by the Layout Editor.

1.5.4 Beginning the Implementation Process

To begin implementing your GUI, proceed to the following sections:

Getting Started with GUIDE - the basics of using GUIDE.

Selecting GUIDE Application Options - set both FIG-file and M-file options.

Using the Layout Editor - begin laying out the GUI.

Understanding the Application M-File - discussion of programming techniques

used in the application M-file.

Application Examples - a collection of examples that illustrate techniques

which are useful for implementing GUIs.

Command-Line Accessibility

When MATLAB creates a graph, the figure and axes are included in the list of children of their respective parents and their handles are available through commands such as findobj, set, and get. If you issue another plotting command, the output is directed to the current figure and axes.

GUIs are also created in figure windows. Generally, you do not want GUI figures to be available as targets for graphics output, since issuing a plotting command could direct the output to the GUI figure, resulting in the graph appearing in the middle of the GUI.

In contrast, if you create a GUI that contains an axes and you want commands entered in the command window to display in this axes, you should enable command-line access.

1.5.5 User Interface Control

The Layout Editor component palette contains the user interface controls that you can use in your GUI. These components are MATLAB uicontrol objects and are programmable via their Callback properties. This section provides information on these components.

- Push Buttons
- Sliders
- Toggle Buttons
- Frames
- Radio Buttons
- Listboxes
- Checkboxes
- Popup Menus
- Edit Text
- Axes
- Static Text
- Figures

Push Buttons

Push buttons generate an action when pressed (e.g., an OK button may close a dialog box and apply settings). When you click down on a push button, it appears depressed; when you release the mouse, the button's appearance returns to its nondepressed state; and its callback executes on the button up event.

Properties to Set

String - set this property to the character string you want displayed on the push button.

Tag - GUIDE uses the Tag property to name the callback subfunction in the application M-file. Set Tag to a descriptive name (e.g., close_button) before activating the GUI.

Programming the Callback

When the user clicks on the push button, its callback executes. Push buttons do not return a value or maintain a state.

Toggle Buttons

Toggle buttons generate an action and indicate a binary state (e.g., on or off). When you click on a toggle button, it appears depressed and remains depressed when you release the mouse button, at which point the callback executes. A subsequent mouse click returns the toggle button to the nondepressed state and again executes its callback.

Programming the Callback

The callback routine needs to query the toggle button to determine what state it is in. MATLAB sets the Value property equal to the Max property when the toggle button is depressed (Max is 1 by default) and equal to the Min property when the toggle button is not depressed (Min is 0 by default).

From the GUIDE Application M-File

The following code illustrates how to program the callback in the GUIDE application M-file.

```
function varargout = togglebutton1_Callback (h, eventdata, handles, varargin)

button_state = get(h,'Value');

if button_state == get(h,'Max')

   % toggle button is pressed

elseif button_state == get(h,'Min')

   % toggle button is not pressed end
```

Adding an Image to a Push Button or Toggle Button

Assign the CData property an m-by-n-by-3 array of RGB values that define a truecolor image. For example, the array a defines 16-by-128 truecolor image using random values between 0 and 1 (generated by rand).

```
a(:,:,1) = rand(16,128);

a(:,:,2) = rand(16,128);

a(:,:,3) = rand(16,128);

set(h,'CData',a)
```

Radio Buttons

Radio buttons are similar to checkboxes, but are intended to be mutually exclusive within a group of related radio buttons (i.e., only one button is in a selected state at any given time). To activate a radio button, click the mouse button on the object. The display indicates the state of the button.

Implementing Mutually Exclusive Behavior

Radio buttons have two states - selected and not selected. You can query and set the state of a radio button through its Value property:

Value = Max, button is selected.

Value = Min, button is not selected.

To make radio buttons mutually exclusive within a group, the callback for each radio button must set the Value property to 0 on all other radio buttons in the group. MATLAB sets the Value property to 1 on the radio button clicked by the user.

The following subfunction, when added to the application M-file, can be called by each radio button callback. The argument is an array containing the handles of all other radio buttons in the group that must be deselected.

```
function mutual_exclude(off); set(off,'Value',0);
```

Checkboxes

Check boxes generate an action when clicked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode (e.g., display a toolbar or generate callback function prototypes).

The Value property indicates the state of the check box by taking on the value of the Max or Min property (1 and 0 respectively by default):

Value = Max, box is checked.

Value = Min, box is not checked.

You can determine the current state of a check box from within its callback by querying the state of its Value property, as illustrated in the following example:

```
function checkbox1_Callback(h,eventdata,handles,varargin)

if (get(h,'Value') == get(h,'Max'))

   % then checkbox is checked-take approriate action

else

   % checkbox is not checked-take approriate action

end
```

Edit Text

Edit text controls are fields that enable users to enter or modify text strings. Use edit text when you want text as input. The String property contains the text entered by the user.

To obtain the string typed by the user, get the String property in the callback.

```
function edittext1_Callback(h,eventdata, handles,varargin)

user_string = get(h,'string');

% proceed with callback...
```

Obtaining Numeric Data from an Edit Test Component

MATLAB returns the value of the edit text String property as a character string. If you want users to enter numeric values, you must convert the characters to numbers. You can do this using the str2double command, which converts strings to doubles. If the user enters non-numeric characters, str2double returns NaN.

You can use the following code in the edit text callback. It gets the value of the String property and converts it to a double. It then checks if the converted value is NaN, indicating the user entered a non-numeric character (isnan) and displays an error dialog (errordlg).

```
function edittext1_Callback(h,eventdata,handles,varargin)

user_entry = str2double(get(h,'string'));

if isnan(user_entry)

   errordlg('You must enter a numeric value','Bad Input','modal')

end

% proceed with callback...
```

Triggering Callback Execution

On UNIX systems, clicking on the menubar of the figure window causes the edit text callback to execute. However, on Microsoft Windows systems, if an editable text box has focus, clicking on the menubar does not cause the editable text callback routine to execute. This behavior is consistent with the respective platform conventions. Clicking on other components in the GUI execute the callback.

Static Text

Static text controls displays 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 and there is no way to invoke the callback routine associated with it

Frames

Frames are boxes that enclose regions of a figure window. Frames can make a user interface easier to understand by visually grouping related controls. Frames have no callback routines associated with them and only uicontrols can appear within frames (axes cannot).

Placing Components on Top of Frames

Frames are opaque. If you add a frame after adding components that you want to be positioned within the frame, you need to bring forward those components. Use the Bring to Front and Send to Back operations in the Layout menu for this purpose.

List Boxes

List boxes display a list of items and enable users to select one or more items.

The String property contains the list of strings displayed in the list box. The first item in the list has an index of 1.

The Value property contains the index into the list of strings that correspond to the selected item. If the user selects multiple items, then Value is a vector of indices. By default, the first item in the list is highlighted when the list box is first displayed. If you do not want any item highlighted, then set the Value property to empty.

The ListboxTop property defines which string in the list displays as the top most item when the list box is not large enough to display all list entries. ListboxTop is an index into the array of strings defined by the String property and must have a value between 1 and the number of strings. Noninteger values are fixed to the next lowest integer

Single or Multiple Selection

The values of the Min and Max properties determine whether users can make single or multiple selections:

If Max - Min > 1, then list boxes allow multiple item selection.

If Max - Min <= 1, then list boxes do not allow multiple item selection

Selection Type

Listboxes differentiate between single and double clicks on an item and set the figure SelectionType property to normal or open accordingly. See Triggering Callback Execution for information on how to program multiple selection.

Popup Menus

Popup menus open to display a list of choices when users press the arrow. The String property contains the list of string displayed in the popup menu. The Value property contains the index into the list of strings that correspond to the selected item. When not open, a popup menu displays the current choice, which is determined by the index contained in the Value property. The first item in the list has an index of 1.

Popup menus are useful when you want to provide users with a number of mutually exclusive choices, but do not want to take up the amount of space that a series of radio buttons requires.

Programming the Popup Menu

You can program the popup menu callback to work by checking only the index of the item selected (contained in the Value property) or you can obtain the actual string contained in the selected item.

This callback checks the index of the selected item and uses a switch statement to take action based on the value. If the contents of the popup menu is fixed, then you can use this approach.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

switch val

case 1

% The user selected the first item

case 2

% The user selected the second item % etc.
```

This callback obtains the actual string selected in the popup menu. It uses the value to index into the list of strings. This approach may be useful if your program dynamically loads the contents of the popup menu based on user action and you need to obtain the selected string. Note that it is necessary to convert the value returned by the String property from a cell array to a string.

```
function varargout = popupmenu1_Callback(h,eventdata,handles,varargin)

val = get(h,'Value');

string_list = get(h,'String');

selected_string = string_list{val}; % convert from cell array to string

% etc.
```

Enabling or Disabling Controls

You can control whether a control responds to mouse button clicks by setting the Enable property. Controls have three states:

on - The control is operational

off - The control is disabled and its label (set by the string property) is greyed out.

inactive - The control is disabled, but its label is not grayed out.

When a control is disabled, clicking on it with the left mouse button does not execute its callback routine. However, the left-click causes two other callback routines to execute: First the figure WindowButtonDownFcn callback executes. Then the control's ButtonDownFcn callback executes. A right mouse button click on a disabled control posts a context menu, if one is defined for that control. See the Enable property description for more details.

Figure

Figures are the windows that contain the GUI you design with the Layout Editor. See the description of figure properties for information on what figure characteristics you can control.

CHAPTER 2

LITERATURE SURVEY

The earth is associate aquatic planet and the maximum amount as eightieth of its surface is roofed by water. Moreover, there is a strong interest in knowing what lies in underwater. Present days, an image of deep waters has a scope to large investigation to explore the underwater for sea floor expedition and navigation. Enthusiasm of underwater imaging includes the inspection of plants, seabed exploration, the search for wrecks up and to the exploration of natural resources. There were several issues faced by the human in the underwater, if he dives deep into the ocean and stay there for a long time to perform experimentation [1] – super script. Due to the above reasons, unmanned remote vehicles are used to sea floor exploration.

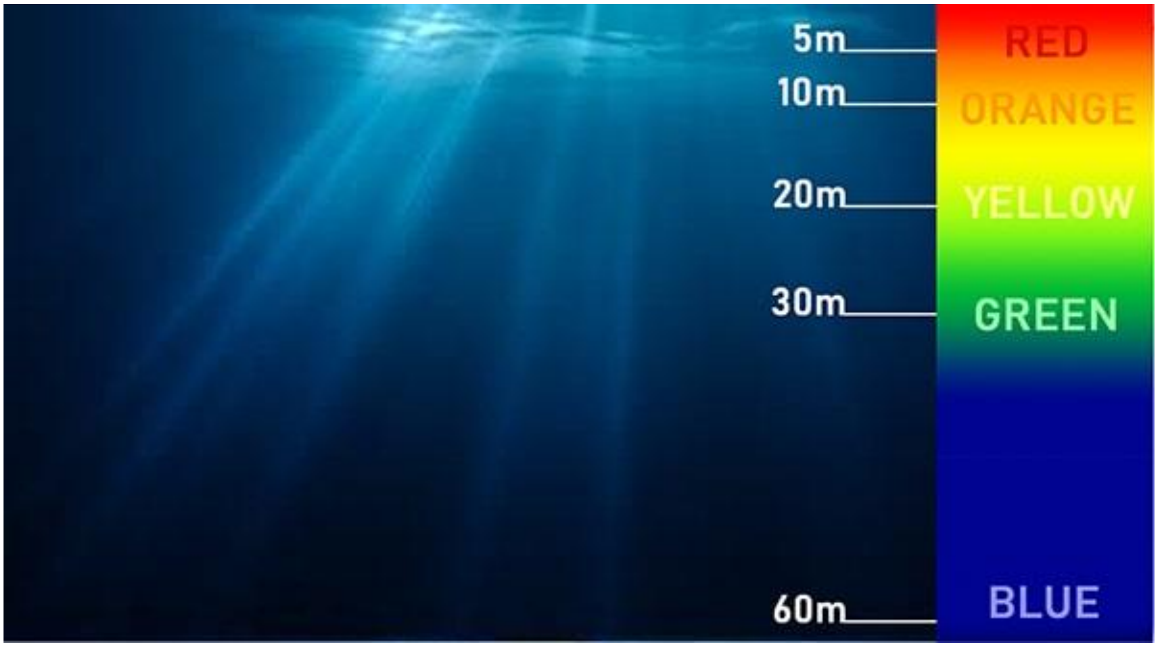


Figure 2.1 Absorption of Light Underwater

Now-a-days, underwater image possess a large research area to discover the underwater for sea floor exploration and navigation. The applications including submerged environment observing, discovering potential outcomes for connected purposes for structural designing and appraisal of coral reefs have an enormous enthusiasm for investigation range for submarine and military operations. But, the lack of ability of the human being in diving in the deep ocean for a long time has increased the difficulties for underwater analysis. Under-water Remotely Operated Vehicle (UROV) are used these days for underwater analysis. But, poor lighting condition and absorption of light inside the water causes low quality to the image taken by the UROV. There are different kinds of phenomenon to disrupt the underwater image like scattering effect which is because of presence of particles inside the water, blurring effect because of poor lighting condition and vignetting effect because of lens used in underwater optic camera [1]. The main problem inside the water is the absorption of light.

One of the scattering effects is the back-scattering. It is an additive noise in the form of marine snow patterns which appear due to reflection of the light from a given natural or artificial source on the suspended particles in the direction of the camera. Adreas et.al.[2] has proposed a method which typically concentrate on local contrast equalization to balance the non-uniform lighting caused by back scattering. To denoise the underwater image in the filter domain Arfia et.al.[3] has introduced a method based on the characteristics of the EMD and wavelet technique. Aysun et.al.[4] has proposed an empirical mode decomposition method to enhance the poor-quality underwater image. Fig. 1: Absorption of light by water Most of the time, due to presence of un-canonical illuminants inside the water heavy unwanted color cast occurs in the image. So images with color cast goes through lost shading differentiation and variety.Shen et.al.[5] has proposed an white balance algorithm through the average equalization and threshold to restore the unwanted color cast. Choudhary et.al.[6] has observed that the images with color cast have standard deviation is different from other color channels. So they proposed a color constancy algorithm using the statistics of image to remove the color cast. Huo et.al.[7] has proposed a white balance algorithm using extracting gray color points in images for color temperature estimation.

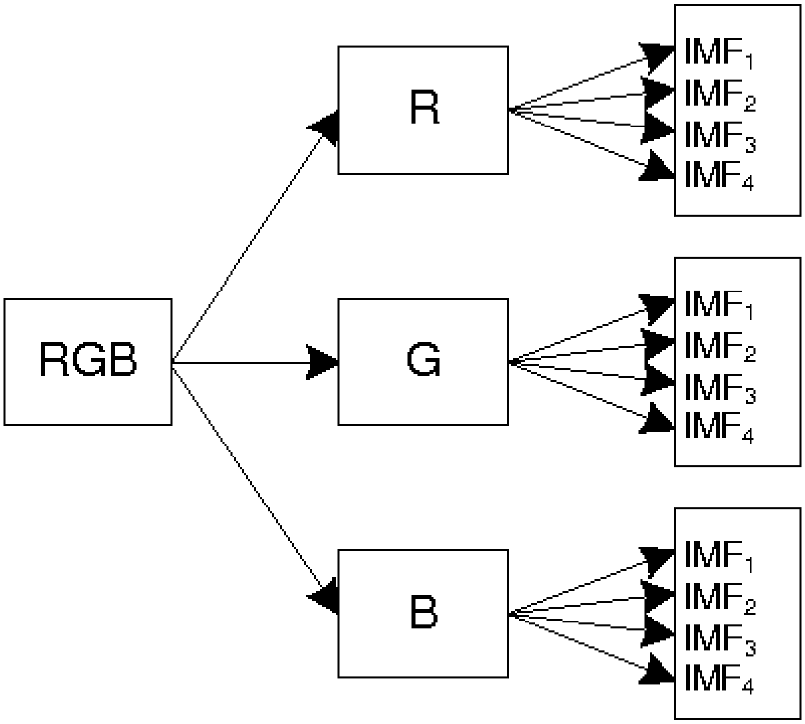


Figure 2.2 EMD Basic Block Diagram

Image enhancement is a universal technique known to restore the optical quality of images or acknowledgment of data in images for human viewers. This paper describes an algorithm where the image is processed through a white balance algorithm which eliminates the unwanted color casts. Then, an EMD technique is used to enhance the resulting image and to restore the color. The final image is compared with different conventional methods visually and statistically, and found to be one of the better enhancement algorithms.

A. Historical Development

Underwater image quality improvement approaches present a path to magnify the object recognition in underwater surrounding. A heap of research started for the upgradation of image visual quality, but a little amount of work has been carried out in this area. In the deep waters, image quality is degrading due to poor illumination conditions and the light properties differ in water compared to air.[2]. There were several parameters which decreases the quality of an image in underground waters. So, in order to remove all these effects

there are several techniques has been implemented and practiced.

B. Need for Pre process

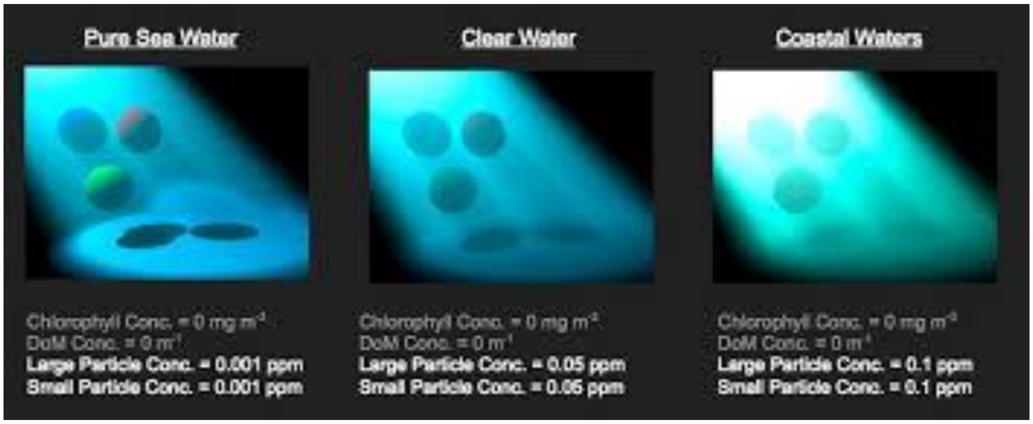

Pure Sea Water
Clear Water
Coastal Waters
Chlorophyll Conc. = 0 mg m-3
DoM Conc. = 0 m-1
Large Particle Conc. = 0.001 ppm
Small Particle Conc. = 0.001 ppm
Chlorophyll Conc. = 0 mg m-3
DoM Conc. = 0 m-1
Large Particle Conc. = 0.05 ppm
Small Particle Conc. = 0.05 ppm
Chlorophyll Conc. = 0 mg m-3
DoM Conc. = 0 m-1
Large Particle Conc. = 0.1 ppm
Small Particle Conc. = 0.1 ppm

Figure 2.3 Water Scattering due to dust

Initially processing is necessary for deep water images due to their poor-quality during acquisition. Necessity for pre-processing of deep-water images [1] are discussed below:

(i) Quality of images taken from deep water is deteriorated due to light ray attributes like scattering and absorption of light.

(ii) Specificity of surroundings such as lighting inequalities, water torridness, and blue complexion is more or less influential when vehicles move.

(iii) Video or image captured from deep waters like unknown rigid scene, and the depth of the scene and low light sensitivity due to Marine snow etc.

TRADITIONAL TECHNIQUES FOR IMAGE ENHANCEMENT

There are several techniques which are used very frequently for processing the image to improve the visual quality. Some of them are as follows:

(i) Contrast Stretching

(ii) Adaptive Histogram Equalization

A. Contrast stretching

The contrast stretching is a method to transform high intense region of image into brighter and less intense region into darker by using a predefined transformation function T(r) [2]. Generally, the underwater images will have fewer grey values. There are 256 grey values. '0' indicates black and '255' indicates white. In this method the current grey value of the image is stretched towards 255 i.e., from black to white, pixel by pixel. That means the contrast of the image is stretched, so that the quality of the image is improved for better vision.

For example:

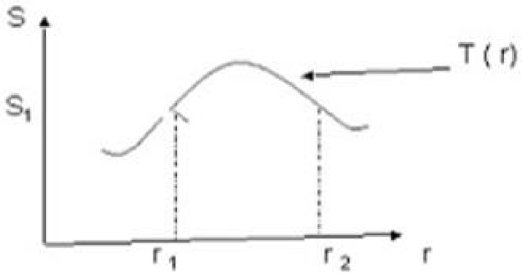


Figure 2.4 Two different gray levels look same

Here two different thresholds are considered for the entire image and the values between them are stretched to the maximum extent, so that the contrast increases. And more over by this method the entire global image contrast is enhanced.

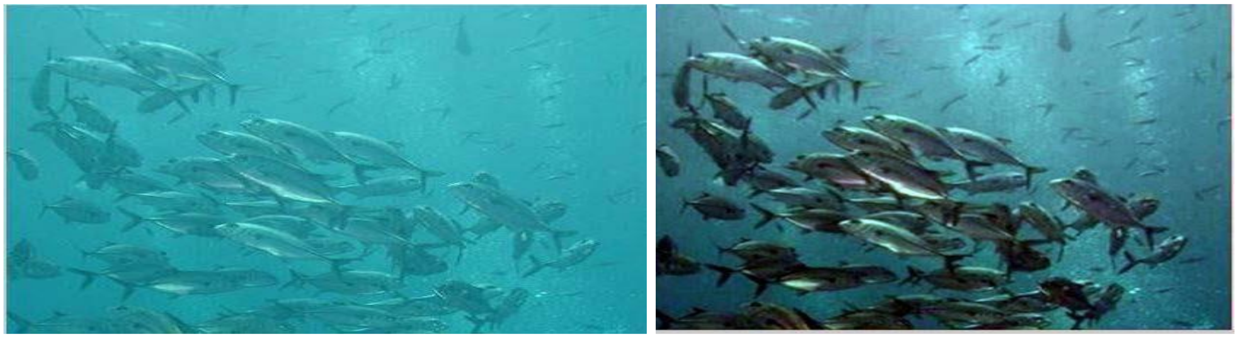


Figure. 2.5 Raw image & Enhanced image (contrast)

But the disadvantage here is that the transformation function is not unique. Depending on the application the suitable transformation function is chosen.

B. Adaptive histogram equalization

Adaptive histogram equalization is a PC based image processing technique which is used to improve the quality of image properties like contrast. It is similar to contrast stretching method but with a slight difference. It computes several intensities of specific Gray value, each corresponding to a distinct portion of an image, and with the help of them intensities are rearranged by applying a suitable transformation function. For example, a simple transformation function such as each pixel transformed based on the histogram of a square surrounding the pixel[3]. Existing values will be mapped to new values keeping actual number of intensities in the resulting image equal or less than the original number of intensities. The transformation function applied on the histogram is proportional to the cumulative distributive function (CDF) of pixel values in the neighbourhood. Therefore, it suits for enhancing the local details and enhancing the edge information of each region of an image.

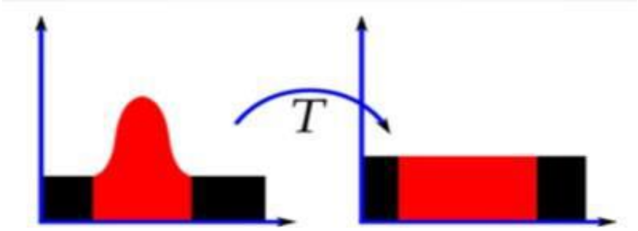


Fig.2.6 Histograms of an image before and after histogram equalization

Histogram equalization is a technique for changing the overall pixel intensities based on transformation function and contrast of an image. Histogram equalization is an effective technique which will benefit for the images with extreme contrast values. The limitation of this technique highlights the unwanted noise present in the background of an image and lead to loss in the information signal. It results in undesired effects in the resultant images[4].

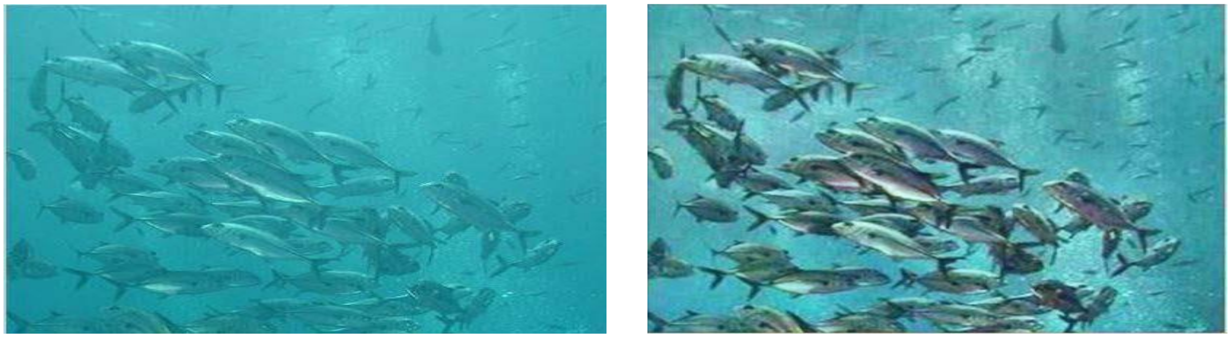


Figure. 2.7 Raw image & Enhanced image (HistEq)

Here the noise in relatively homogeneous regions of the image is amplified which results in poor SNR. And also, only the local objects of the image are enhanced and the background is left unenhanced.

Hitam et al. (2013)[2] have discussed a new method specifically developed for enhancing the underwater images called mixture Contrast Limited Adaptive Histogram Equalization (CLAHE) color model. The method operates Contrast Limited Adaptive Histogram Equalization on RGB and HSV color model and Euclidean norm is used to combine both results together. The combined results show less mean square error and high peak signal to noise ratio (PSNR) then other methods of underwater image enhancing. It shows that the projected method is capable of classifying coral reefs particularly when visual cues are visible.

Shelda Mohan and T.R. Mahesh, 2013[5] has presented Particle Swarm Optimization (PSO) for tuning the enhancement parameter of Contrast Limited Adaptive Histogram Equalization relied on Local Contrast Modification (LCM). The quality of enhanced image is tested using a criteria based on edge information of the image. The planned method provides finest contrast enhancement though preserving the local data and details of the input mammogram picture. Sowmyashree et al. 2014 have presented a relative study of the different image enhancement methods used for enhancing images of the bodies under the water. It also describes the various properties of water due to which the underwater images images are distorted and degraded.

Setiawan et al. 2013[7] used Contrast Limited Adaptive Histogram Equalization (CLAHE) to enhance color retinal image. In this paper, they proposed new enhancement method using CLAHE in G channel to improve the color retinal image quality. The enhancement process conduct in G channel is suitable to enhance the color retinal image quality. Visual observation is used to judge the enhanced images and compare them with the original ones.

Chang et al. 2014[1] have proposed the mean-variance analysis technique that is engaged in partitioning the grey scale image into four associated images for individual image. The contrast of the palm bone X-ray radiographs is enhanced by newly proposed technique i.e. quad histogram equalization technique. Experimental results using this method illustrate that the proposed algorithm is better than the global histogram equalization (GHE) technique and brightness saving bi-histogram equalization (BBHE) technique.

Khan et al. 2012[3] has proposed Bi- and Multi-histogram equalization methods designed for contrast improvement of digital images. Multi-HE methods are projected so that natural look of image is maintained at the cost of either the brightness or its contrast. Simulation results for a number of trial images shows that the proposed method enhances the contrast even as preserving brightness and natural look of the images.

Senthilkumaran N and Thimmiaraja J 2014[6] have compared different techniques such as Global Histogram Equalization (GHE), Local histogram equalization (LHE), Brightness preserving Dynamic Histogram equalization (BPDHE) and Adaptive Histogram Equalization (AHE) by means of diverse objective quality measures for MRI brain image improvement. Quality measures used for comparison are Weber contrast, Michelson contrast, Contrast and AMBE.

Talha et al. 2013[4] have proposed Balanced Contrast Limited Adaptive Histogram Equalization (BCLAHE) for Adaptive Dynamic Range Compression (ADRC) of real time medical pictures. The proposed method scheme is tested and has given away high-quality results in terms of latency and perceptibility of tiny details. They have concluded that Balanced-CLAHE gives accurate results in improving local information than global histogram equalization.

Erturk et al.2012 have presented a new algorithm based on an Empirical Mode Decomposition (EMD) which is used to improve visibility of underwater images. It is indicated that the proposed method provides finer results compared to regular methods such as contrast stretching, histogram equalization. In the given approach, initially EMD is used for decomposing every spectral part of an underwater image into Intrinsic Mode Functions (IMFs). Then by combining the IMFs of spectral channels, enhanced image is constructed with variables weights in order to attain an improved image with enhanced visual features.

Galdran et al. 2014 proposed a Red Channel method, where colors associated to short wavelengths are recovered, as expected for underwater images, leading to a recovery of the lost contrast. The Red Channel method can be interpreted as a variant of the Dark Channel method used for images degraded by the atmosphere when exposed to haze. Experimental results are also shown.

Sasi et al. 2013 constructed productive color space for enhancing the contrast of myocardial perfusion images. Effects of histogram equalization and contrast limited adaptive histogram equalization are founded by the investigation. The method which gives good contrast improvement outcome is used for the appropriate color space. The color space giving better outcomes is selected experimentally. Exceptionality of this work is that contrast limited adaptive histogram equalization(CLAHE) technique is applicable to the chrominance parts of the cardiac nuclear image. It left the luminance channel unchanged which consequence an improved image as resultant in projected color space.

G.Padmavathi et al. 2010 have compared and evaluated three filters performance. These filters are homomorphic filter, anisotropic diffusion and wavelet denoising by average filter. All these filters are helpful in pre-processing of underwater images. Image quality is improved, noise is suppressed, edges in an image are preserved and image is smoothening by the use of these filters. Among the three filters used wavelet denoising by average filter gives required results in terms of Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR).

Singh et al. have done analysis of different underwater image enhancement techniques. The comparison between performance of contrast limited adaptive histogram equalization method, contrast stretching, and histogram equalization method is done. Mean square error (MSE) and signal to noise ratio (SNR) are used as parameters for comparing the performance of above methods. The methods were examined on different type of underwater images.

Chiang et al. 2012[15]have proposed a fresh efficient approach based on dehazing algorithm, used to enhance underwater images. This algorithm is used to compensate the attenuation inconsistency along the transmission course and to acquire the possible effect of presence of an artificial source of light into consideration. The haze occurrence and deviation in wavelength attenuation along the propagation path underwater to camera are corrected after compensating the influence of artificial light. The performance was evaluated both objectively and subjectively, of the proposed algorithm for wavelength compensation and image dehazing(WCID) by using ground truth color patches.

Garcia et al. 2002 have analysed and compared already available techniques for dealing with the problems of underwater images. These techniques mainly deal with nonuniform illumination, low contrast in underwater images. The analysed methodologies consist the review of the homomorphic filtering, illumination-reflectance model, local histogram equalization and subtraction of the illumination field. Many illustrations on real data have been carried out to compare and contrast the dissimilar methods.

Iqbal et al. 2007 have projected an approach which is based on slide stretching. This approach has dual objectives. First objective is to balance the colour contrast of images by applying the contrast stretching of RGB color model. Second objective is to amplify the true color and resolve the problem of illumination by the use of saturation and intensity stretching of HSI color space. For enhancing the underwater images an interactive software has been proposed.

CHAPTER 3

PROPOSED SYSTEM

3.1 BLOCK DIAGRAM

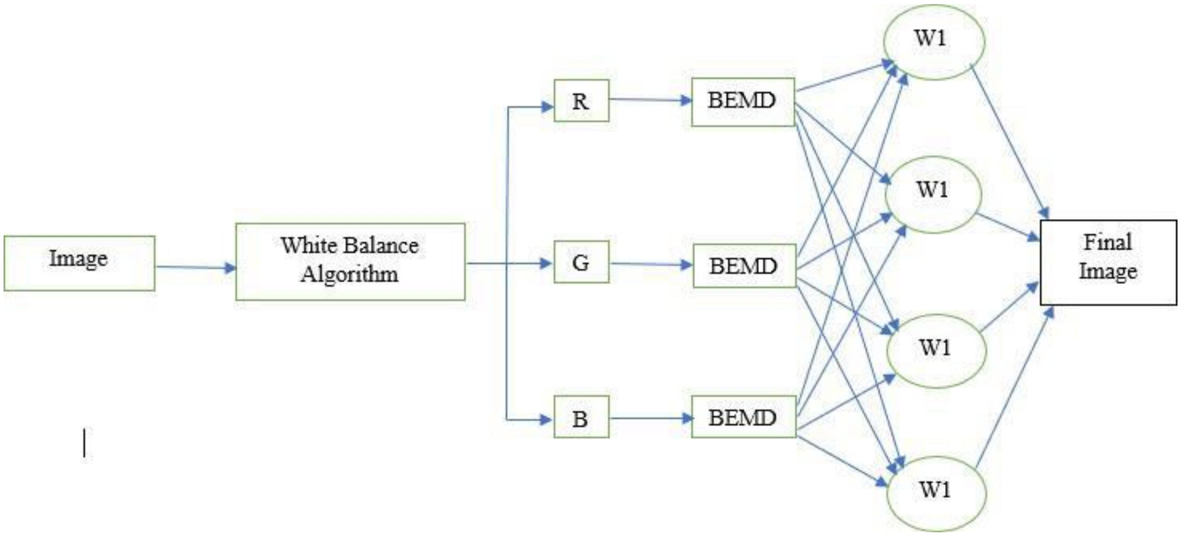


Figure 3.1 Block Diagram of BEMD

3.2 WORKING

The White Balanced Empirical Mode Decomposition is a combination of two approaches i.e., the White Balanced approach and the BEMD approach.

BALANCHING OF PHOTOMETRIC VARIATIONS:

In order to acquire image a source of illumination is required. In the deep waters the main source of illumination is sun. As the acquired image type is of RGB, in the deep waters blue color variations are more dominant due to its shorter wavelength. As the color image consists of 3 planes i.e., red, green and blue, due to improper illumination source the impact of blue color compared to red and green is more. Due to this there is a necessity to balance the color combination such that it is invariant of illumination. In order to balance those variations, assess the color of predominating light and remove the effects of that color and make the acquired image free from photometric variations. These effects are removed by applying the transformation on pixel intensities of each color plane by an average pixel intensity value of that color plane [5].

BIDIRECTIONAL EMPIRICAL MODE DECOMPOSITION:

Empirical mode decomposition is as similar to the wavelet decomposition but it decomposes the signal at different time scales [6] The adaptive decomposition preserves the local signal attributes and assure that the signals in all IMFs are mono-component. As mono-component signals allow the calculation of physically meaningful instantaneous frequencies, any event in the signal can hence be localized on the time axis as well as the frequency axis by presenting the IMFs in frequency-time-energy distributions.

The IMFS obtained are almost perpendicular to the original signal. The necessary conditions to be satisfied by the IMFs are:

a) In each and every IMF, the no. of extremities and the zero crossings rate should be equal or may be differed by one.

b) At any point of time, for each IMF average value of the envelope computed from local maxima or local minima must be zero.

By considering the above two conditions, it came to know that IMFs are correlated to oscillatory modes in time series. Addition of all IMFs and the residual signal recovers the entire signal. The IMFs contain different frequency ranges, where the highest frequencies are usually found in the first IMF which contains the edge information with reference to image and lower frequencies in subsequent IMFs consists of spatial information of the image [7].

In the White Balanced EMD algorithm, which is shown in the *Fig.5.*, Initially the image is processed to remove the color effects caused due to improper illumination on the image source i.e., to make image free from photometric variations. From the processed input image of RGB, individual color planes are extracted. Now, each color plane is decomposed into its intrinsic mode functions using EMD method. Each IMFs are scaled with a weight and normalized [5].

After this processing, the recovered image is constructed by considering product of weights and the lower frequency IMFs. The reconstructed image is a visually enhanced image.

- **INPUT IMAGE**

  Input image is the colour image which is taken as the input for this project.

- **CHANNEL SEPARATION**

  The aim of pre-processing is an improvement of the image data that suppresses unwilling distortions or enhances some image features important for further processing, although geometric transformations of images (e.g. rotation, scaling, translation) are classified among pre-processing methods here since similar techniques are used.

- **WHITE BALANCING**

  An image transform can be applied to an image to convert it from one domain to another domain.

- **PROCESSING RGB CHANNELS**

  The feature clustering method is used to enhance the feature expression ability of image dictionary and improve the accuracy of the mapping matrix. Moreover, we calculate the image feature mapping matrix offline by using collaborative representation, which increases the image reconstruction speed. Experimental results show that the proposed Super resolution method not only improves the image reconstruction efficiency, but also reconstructs more high-frequency information, making the reconstructed image closer to the input image.

- **CONCATATION OF RGB CHANNELS**

  After the process of super resolution method in the previous step a resultant image is obtained from that image a particular part is zoomed and shown as output in this process.

- **RESULT OF OUTPUT**

  Hence, final output image is obtained.

CHAPTER 4

MATLAB SOFTWARE INSTALLATION & FUNCTIONS

4.1 INSTALLATION PROCEDURE FOR MATLAB SOFTWARE

Redundant clients-Windows XP-SP2/Vista/Windows 7/Windows 8

Download:

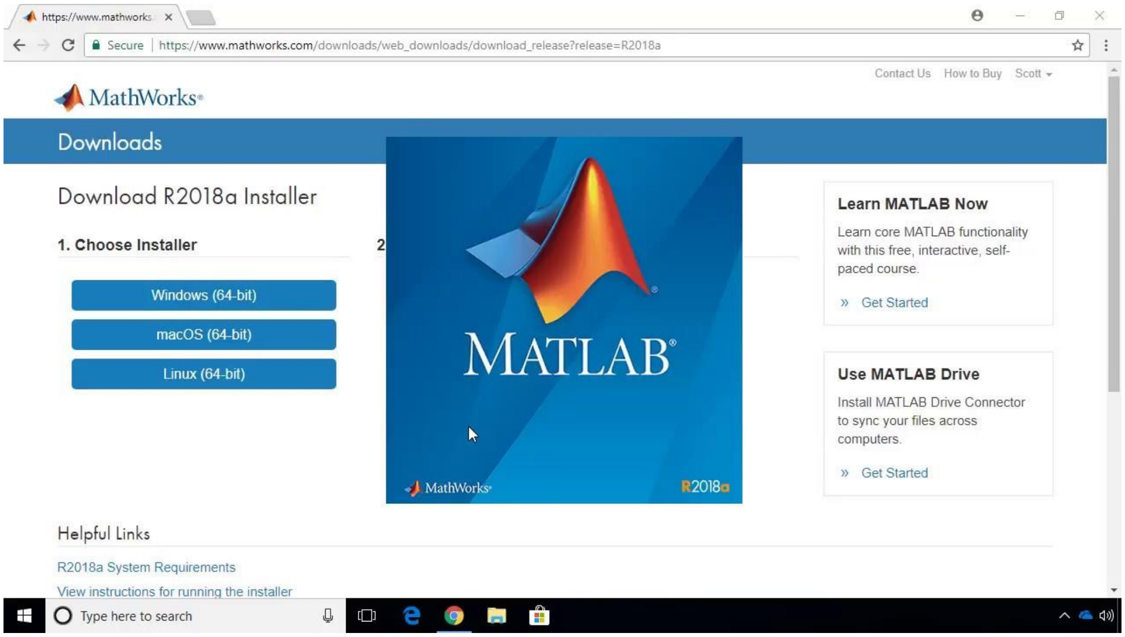
- win32.rar or win64.rar (for 64-bit system)
- PLP
- "license.dat" ***from http://mirror.iitd.ernet.in/matlab*** into Your Desired Folder.

STEPS: Installation of MATLAB Software for window XP/VISTA / Windows 7/8

Get administrator privileges for the system on which you plan to install MATLAB. Use WinRAR to extract RAR file

Step1: Start the installer

For Windows, double-clicking on setup.exe



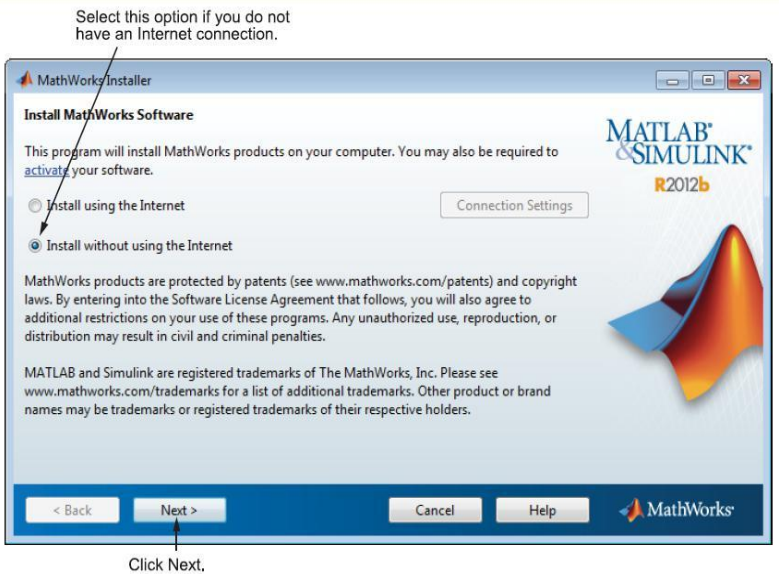
Learn core MATLAB functionality with this free, interactive, self-paced course.

Install MATLAB Drive Connector to sync your files across computers.

Figure 4.1.1 Download the Installer

Step 2: Choose to Install Without Using the Internet

When it starts, the installer displays the following dialog box. Select the Install without using the Internet option and Click OK to proceed with installation.



Select this option if you do not have an Internet connection.

MathWorks Installer

Install MathWorks Software

This program will install MathWorks products on your computer. You may also be required to activate your software.

○ Install using the Internet

● Install without using the Internet

MathWorks products are protected by patents (see www.mathworks.com/patents) and copyright laws. By entering into the Software License Agreement that follows, you will also agree to additional restrictions on your use of these programs. Any unauthorized use, reproduction, or distribution may result in civil and criminal penalties.

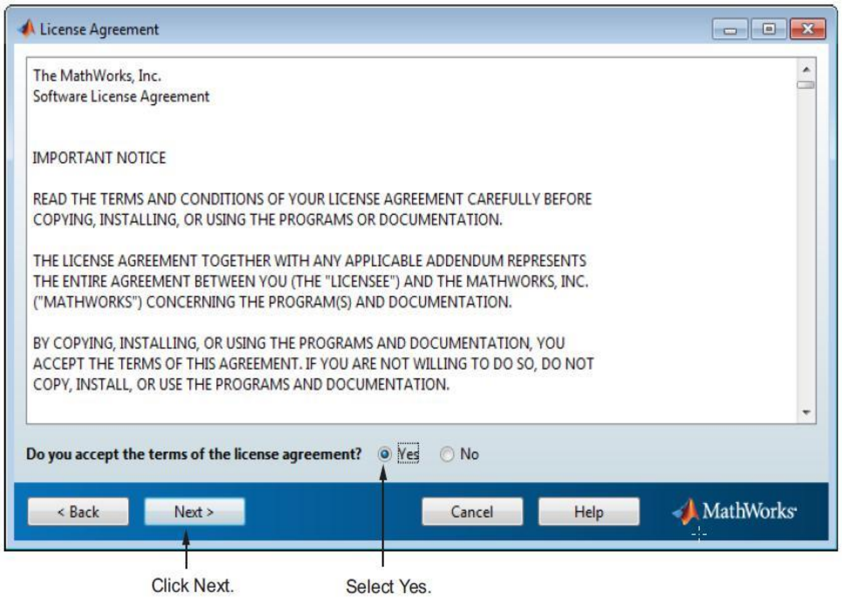
MATLAB and Simulink are registered trademarks of The MathWorks, Inc. Please see www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Click Next.

Figure 4.1.2 Installation step 2

Step 3: Review the License Agreement

Review the software licensing agreement and, if you agree to its terms, click Yes.



License Agreement

The MathWorks, Inc.
Software License Agreement

IMPORTANT NOTICE

READ THE TERMS AND CONDITIONS OF YOUR LICENSE AGREEMENT CAREFULLY BEFORE COPYING, INSTALLING, OR USING THE PROGRAMS OR DOCUMENTATION.

THE LICENSE AGREEMENT TOGETHER WITH ANY APPLICABLE ADDENDUM REPRESENTS THE ENTIRE AGREEMENT BETWEEN YOU (THE "LICENSEE") AND THE MATHWORKS, INC. ("MATHWORKS") CONCERNING THE PROGRAM(S) AND DOCUMENTATION.

BY COPYING, INSTALLING, OR USING THE PROGRAMS AND DOCUMENTATION, YOU ACCEPT THE TERMS OF THIS AGREEMENT. IF YOU ARE NOT WILLING TO DO SO, DO NOT COPY, INSTALL, OR USE THE PROGRAMS AND DOCUMENTATION.

Do you accept the terms of the license agreement? ● Yes ○ No

Click Next.

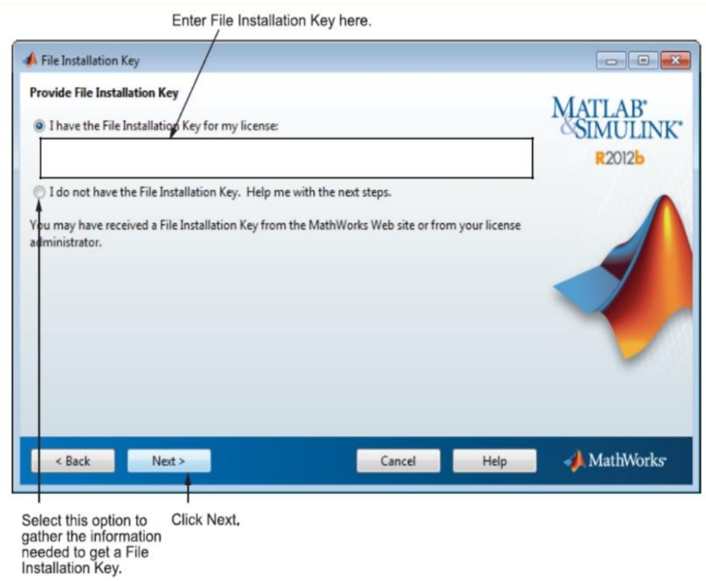
Select Yes.

Figure 4.1.3 Installation step 3

Step 4: Enter the File Installation Key

Enter your File Installation Key and Click OK.

10706-16682-10020-38749-58852-13152-57659-27128-07009-39738-29125-60143



Enter File Installation Key here.

File Installation Key

Provide File Installation Key

● I have the File Installation Key for my license:

○ I do not have the File Installation Key. Help me with the next steps.

You may have received a File Installation Key from the MathWorks Web site or from your license administrator.

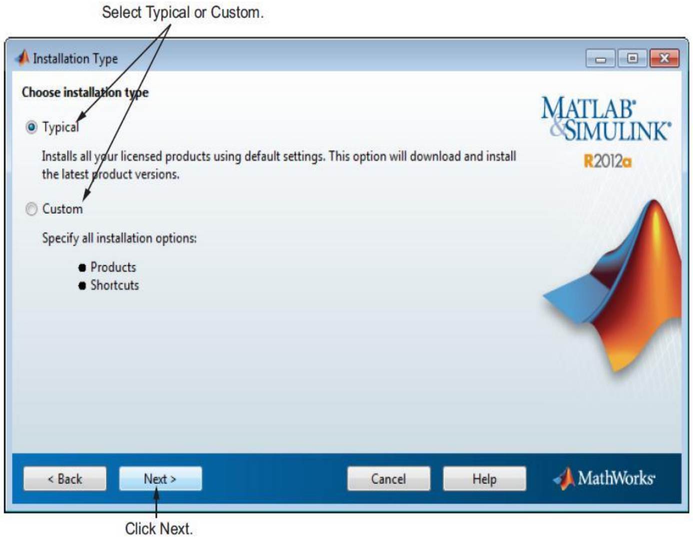
Select this option to gather the information needed to get a File Installation Key.

Click Next.

Figure 4.1.4 Installation step 4

Step 5: Choose the Installation Type

In the Installation Type dialog box, specify whether you want to perform a Custom installation and click **Next**.



Select Typical or Custom.

Installation Type

Choose installation type

● Typical

Installs all your licensed products using default settings. This option will download and install the latest product versions.

○ Custom

Specify all installation options:

- Products
- Shortcuts

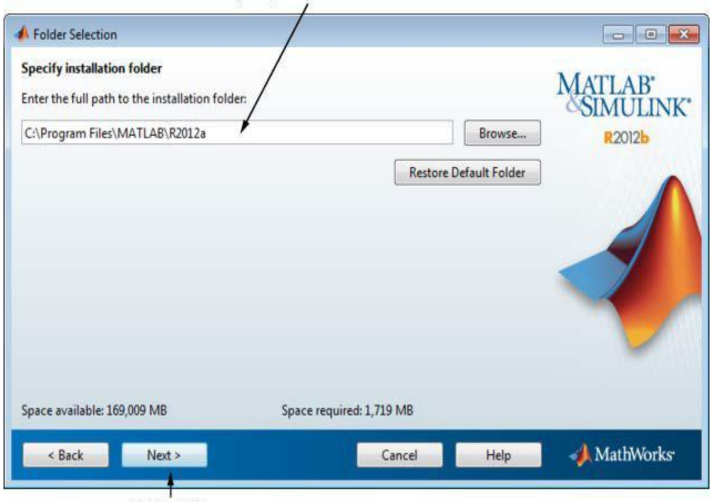
Click Next.

Figure 4.1.5 Installation step 5

Step 6: Specify the Installation Folder

Specify the name of the folder where you want to install MathWorks products. Accept the default installation folder or click Browse to select a different one.

Specify name of installation folder.



Folder Selection

Specify installation folder

Enter the full path to the installation folder:

C:\Program Files\MATLAB\R2012a

Space available: 169,009 MB

Space required: 1,719 MB

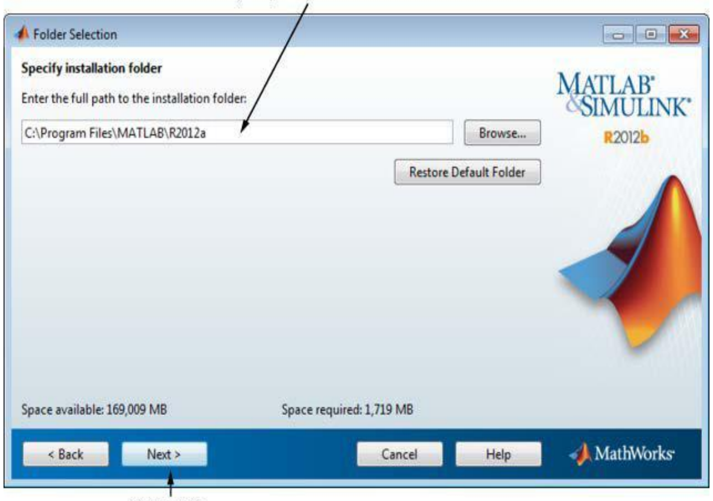
Click Next.

Figure 4.1.6 Installation step 6

Step 7: Specify Products to Install (Custom Only)

Leave it by default and continue.

Specify name of installation folder.



Folder Selection

Specify installation folder

Enter the full path to the installation folder:

C:\Program Files\MATLAB\R2012a

Space available: 169,009 MB

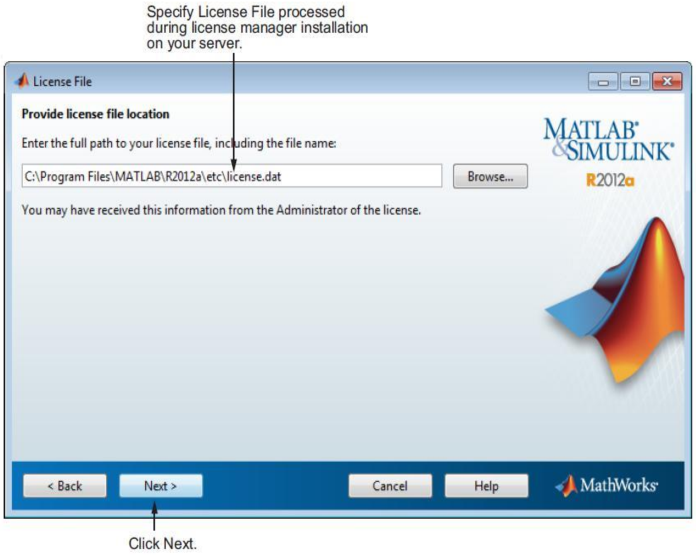
Space required: 1,719 MB

Click Next.

Figure 4.1.7 Product Installation

Step 8: Specify the Location of the License File

Enter the full path of your License File in the text box (or drag and drop the file) and click **Next**.



Specify License File processed during license manager installation on your server.

License File

Provide license file location

Enter the full path to your license file, including the file name:

C:\Program Files\MATLAB\R2012a\etc\license.dat

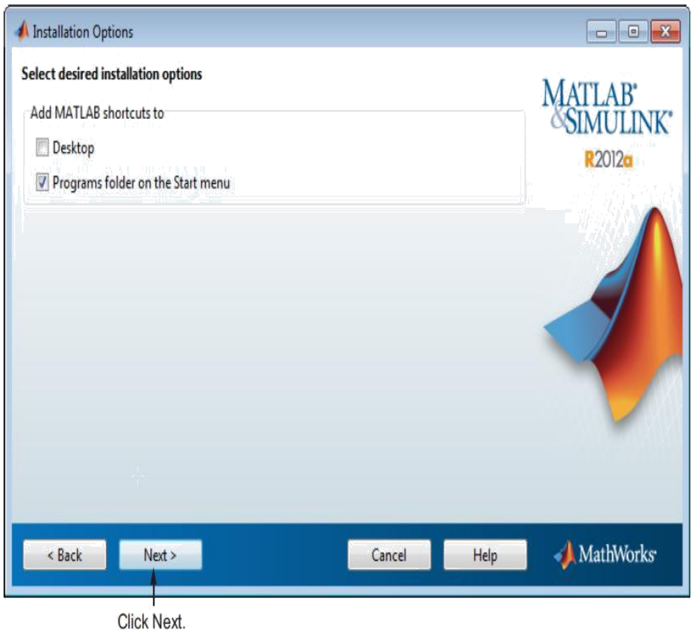
You may have received this information from the Administrator of the license.

Click Next.

Figure 4.1.8 Installation step 7

Step 9: Specify Installation Options (Custom Only)

After selecting installation options, click **Next** to proceed with the installation.



Installation Options

Select desired installation options

Add MATLAB shortcuts to

□ Desktop

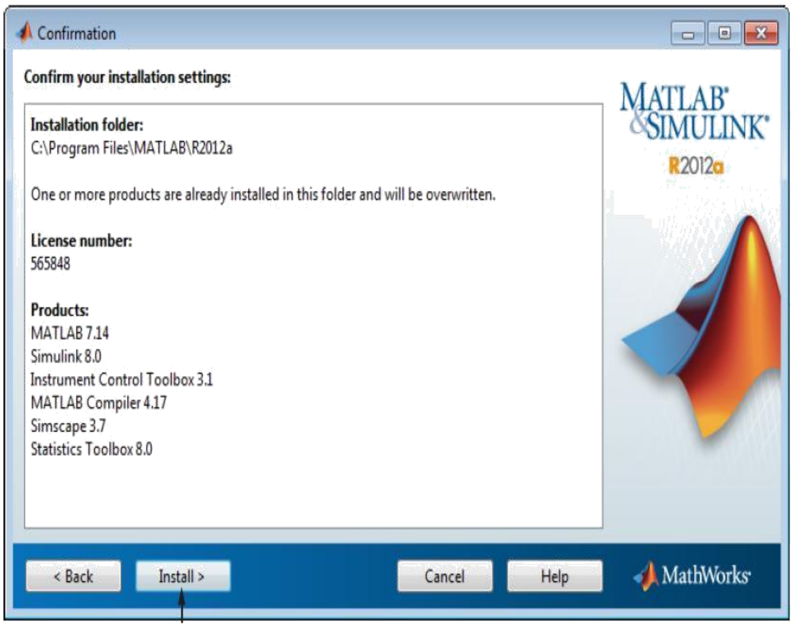
☑ Programs folder on the Start menu

Click Next.

Figure 4.1.9 Installation step 9

Step 10: Confirm Your Choices and Begin Copying Files

Before it begins copying files to your hard disk, the installer displays a summary of your installation choices. To change a setting, click **Back**. To proceed with the installation, click **Install**



Confirm your installation settings:

Installation folder:
C:\Program Files\MATLAB\R2012a

One or more products are already installed in this folder and will be overwritten.

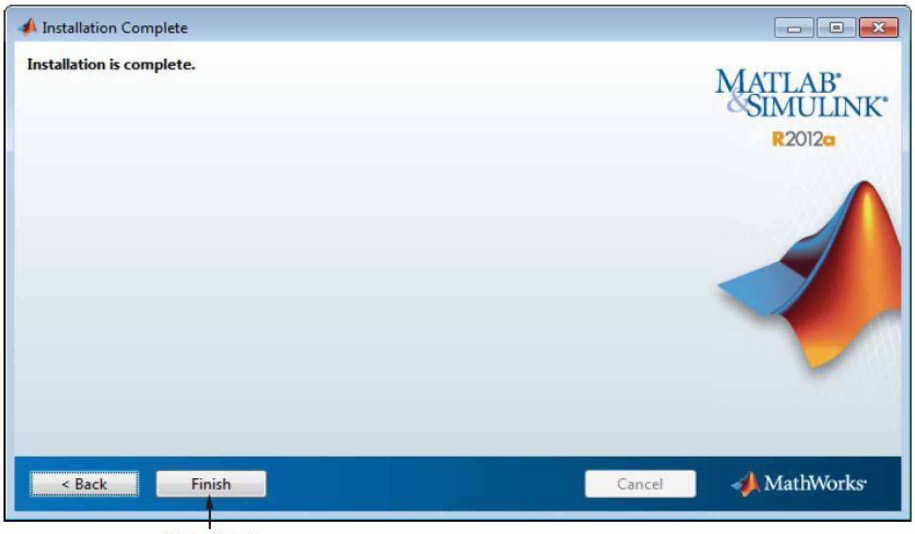
License number:
565848

Products:
MATLAB 7.14
Simulink 8.0
Instrument Control Toolbox 3.1
MATLAB Compiler 4.17
Simscape 3.7
Statistics Toolbox 8.0

Click Install.

Figure 4.1.10 Installation step 10

Step 11: Complete the Installation



Installation is complete.

Click Finish.

Figure 4.1.11 Installation step 11

Step 12: Set Environment Variables

Click right Button on My Computer

Select Properties……..>Advanced System settings …….>Environment

Variables…>New

The environment variable that should be added is as follows:

Variable_ name: MLM_LICENSE_FILEVariable_value:27000@licmngr1.iitd.ernet.in, 27000@licmngr2.iitd.ernet.in, 27000@ licmanager.cse.iitd.ernet.in.

4.2 FUNCTIONS USED:

- Randn()-The randn function generates arrays of random numbers whose elements are normally distributed with mean 0, variance , and standard deviation . Y = randn(n) returns an n -by- n matrix of random entries. An error message appears if n is not a scalar.

- Uigetfile()-Description. uigetfile displays a dialog box used to retrieve one or more files. The dialog box lists the files and directories in the current directory.uigetfile(' FilterSpec ') displays a dialog box that lists files in the current directory.

- **Isequal()-**tf = **isequal**(A,B) returns logical 1 (true) if A and B are equivalent; otherwise, it returns logical 0 (false).

- f = **warndlg**(msg) creates a nonmodal warning dialog box with the specified message and returns the dialog box figure object f . The message text wraps to fit the dialog box. The dialog box title is Warning Dialog .

- A = **imread**(filename) reads the image from the file specified by filename , inferring the format of the file from its contents. If filename is a multi-image file, then **imread** reads the first image in the file.

- d = **size**(X) returns the sizes of each dimension of array X in a vector d with ndims(X) elements. If X is a scalar, which **MATLAB** regards as a 1-by-1 array, **size**(X) returns the vector [1 1] . [m,n] = **size**(X) returns the **size** of matrix X in separate variables m and n .

- **double** is the default numeric data type (class) in **MATLAB**®, providing sufficient precision for most computational tasks. Numeric variables are automatically stored as 64-bit (8-byte) **double**-precision floating-point values.

- B = **imresize**(A , scale) returns image B that is scale times the size of A . The input image A can be a grayscale, RGB, or binary image. If A has more than two dimensions, **imresize** only resizes the first two dimensions. ... By default, **imresize** returns a new, optimized colormap (newmap) with the resized image.

- **imshow**(I) displays the grayscale image I in a figure. **imshow** uses the default display range for the image data type and optimizes figure, axes, and image object properties for image display. ... **imshow** displays the minimum value in I as black and the maximum value as white.

- I = **mat2gray**(A , [amin amax]) converts the matrix A to a grayscale image I that contains values in the range 0 (black) to 1 (white). amin and amax are the values in A that correspond to 0 and 1 in I . Values less than amin are clipped to 0, and values greater than amax are clipped to 1.

- **imagesc**(C) displays the data in array C as an **image** that uses the full range of colors in the colormap. Each element of C specifies the color for one pixel of the **image**. The resulting **image** is an m -by- n grid of pixels where m is the number of rows and n is the number of columns in C .

- **imcrop** returns the cropped image, Icropped . With this syntax and the other interactive syntaxes, the Crop Image tool blocks the **MATLAB**® command line until you complete the operation. ... **imcrop** returns the cropped indexed image, Xcropped , which also has the color map cmap .

- [PSNR , MSE , MAXERR , L2RAT] = **measerr**(X , XAPP) returns the peak signal-to-noise ratio, PSNR , **mean** square error, MSE , maximum squared error, MAXERR , and ratio of squared norms, L2RAT , for an input signal or image, X , and its approximation, XAPP .

- values = set(H,Name) returns the possible values for the specified property. If the possible values are character vectors, set returns each in a **cell** of the **cell array** values . For other properties, set returns a statement indicating that Name does not have a fixed set of property values.

- disp(X) displays the **value** of variable X without printing the variable name. Another way to **display** a variable is to type its name, which displays a leading “ X = ” before the **value**. If a variable contains an empty **array**, disp returns without displaying anything.

- **subplot**(m , n , p) divides the current figure into an m -by- n grid and creates axes in the position specified by p . **MATLAB**® numbers **subplot** positions by row. The first **subplot** is the first column of the first row, the second **subplot** is the second column of the first row, and so on.

- title(fname) evaluates the function that returns a **string** and displays the **string** at the top and in the center of the current axes. title(...,' PropertyName ',PropertyValue,...) specifies property name and property value pairs for the text graphics object that title creates.

- The fprintf function. The fprintf function is used for printing information to the screen. The fprintf function prints an **array** of characters to the screen: fprintf('Happy Birthday\n');

- PSNR- This function displays the PSNR (peak signal-to-noise ratio) between two images. The answer is in decibels (dB). PSNR is very common in image **processing**. A sample use is in the comparison between an original image and a coded/decoded image.

- This **MATLAB function** calculates the **mean-squared error** (**MSE**) between the arrays X and Y

- **Sort()** Sorts the in ascending or descending order.

- **adapthisteq () :** Contrast-limited Adaptive Histogram Equalization (CLAHE). adapthisteq enhances the contrast of images by transforming the values in the intensity image I. Unlike HISTEQ, it operates on small data regions (tiles), rather than the entire image. Each tile’s contrast is enhanced, so that the histogram of the output region approximately matches the specified histogram. The neighbouring tiles are then combined using bilinear interpolation in order to eliminate artificially induced boundaries. The contrast, especially in homogeneous areas, can be limited in order to avoid amplifying the noise which might be present in the image.

CHAPTER 5

RESULT ANALYSIS

RESULTS:

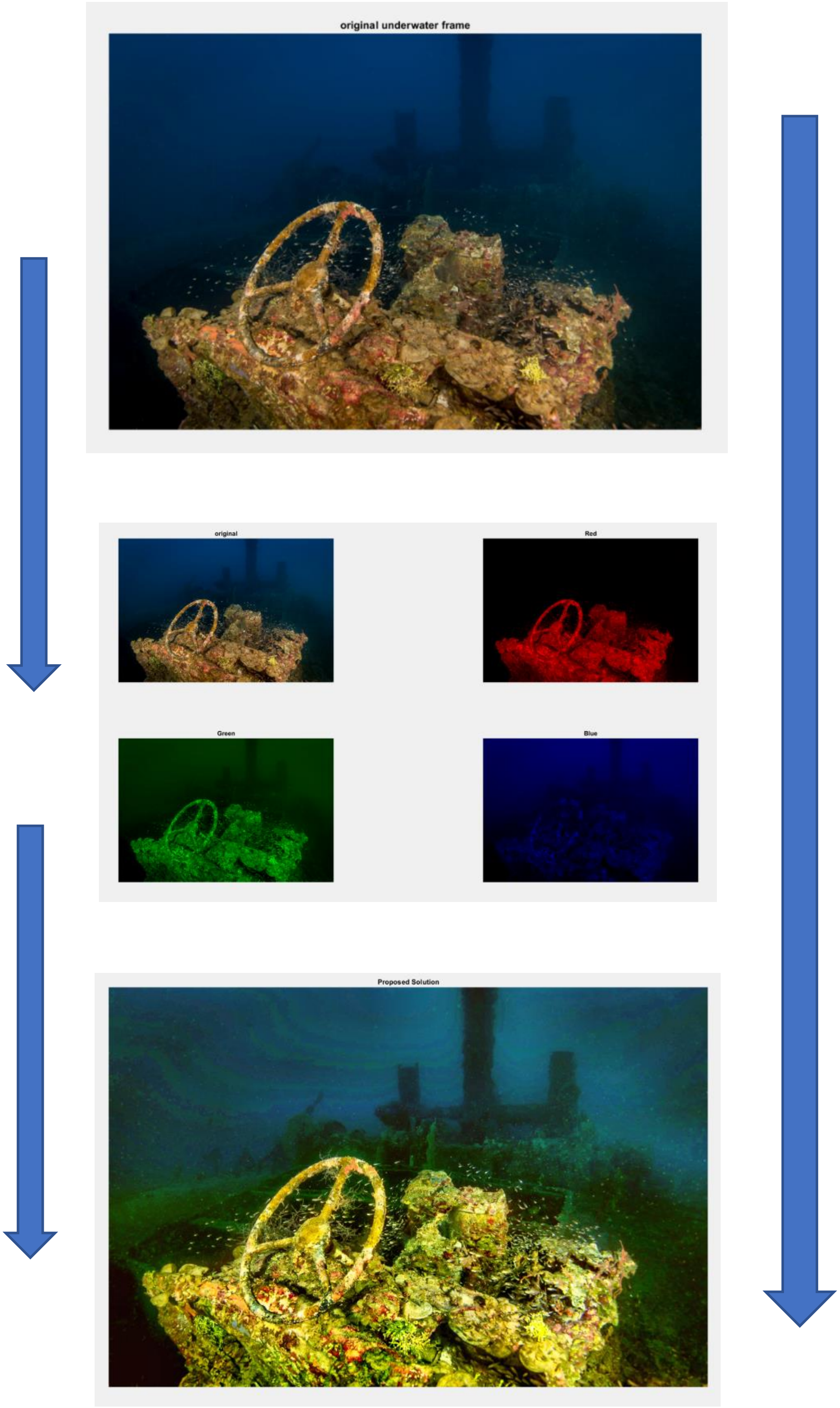


Figure 5.1 The Complete Workflow

STEP 1:

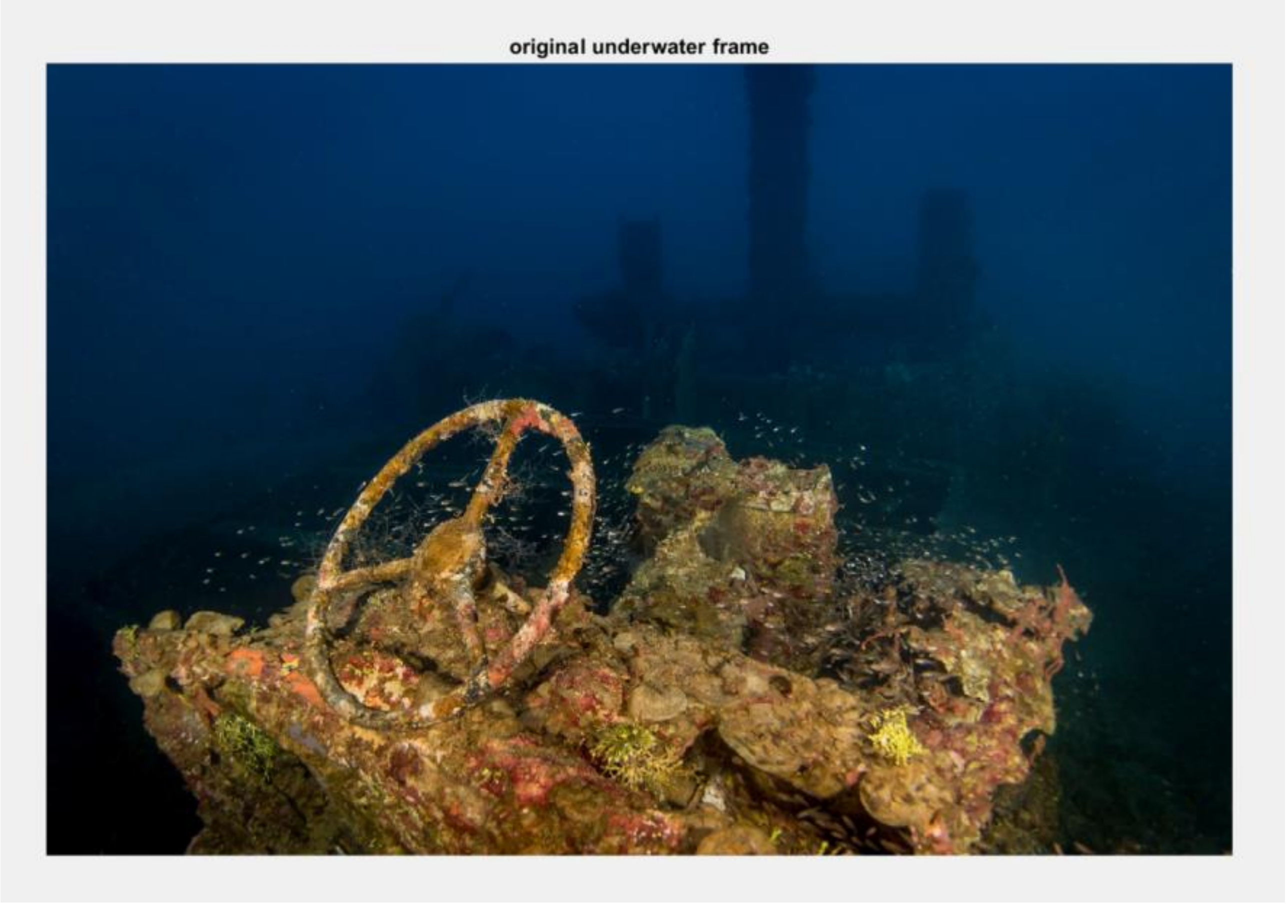


Figure 5.2 Underwater Input Image

As you can see in Figure 5.2, the background of the image is totally is washed out. For SLAM we need to have a clear idea of the proximity and the overall range in the input image.

Having a better input image improves the output of the SLAM program. The workflow is as given in the Figure 5.1

Improvement to be seen:

- Extreme Blue Tint
- Washed Out Background
- Not Color Accurate
- Low Exposure
- Less Sharpness

STEP 2:

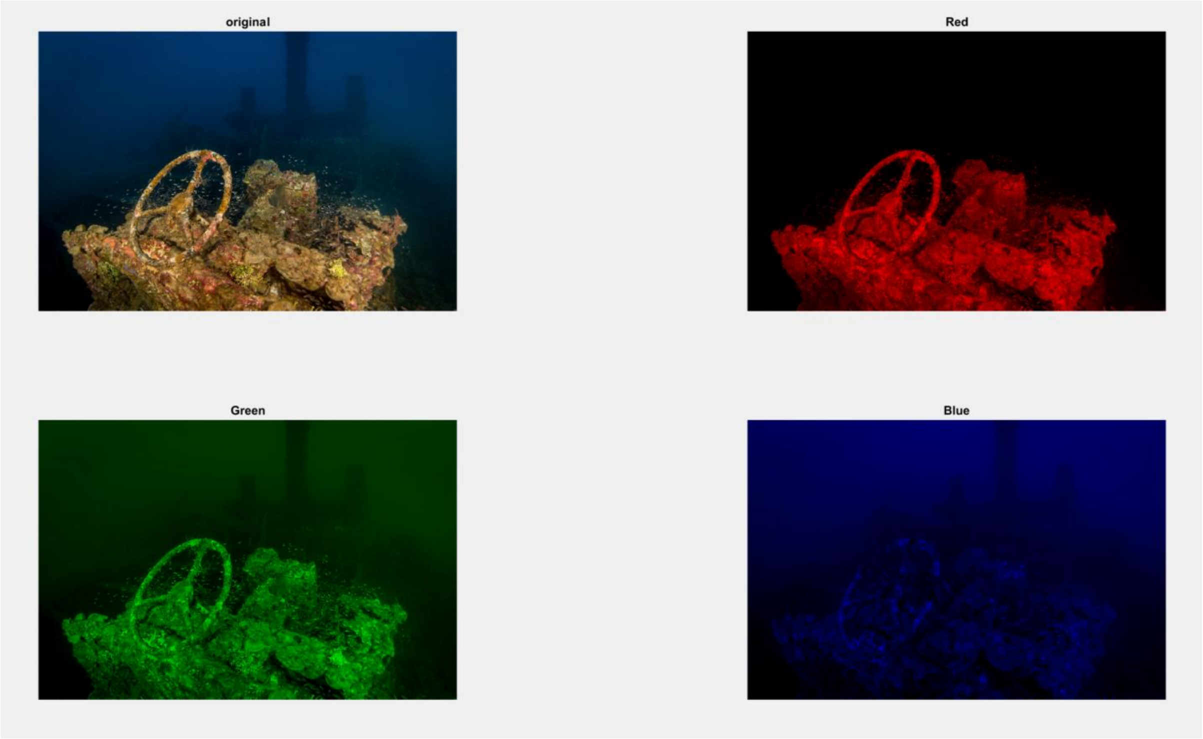


Figure 5.3 Separating all RGB Channels

An RGB colour can be understood by thinking of it as all possible colours that can be made from three coloured lights for red, green, and blue. Imagine, for example, shining three lights together onto a white wall in a dark room: one red light, one green light, and one blue light, each with dimmers. If only the red light is on, the wall will be red. If only the green light is on, the wall will look green. If the red and green lights are on together, the wall will look yellow. Dim the red light and the wall will become more of a yellow-green. Dim the green light instead, and the wall will become more orange.

Figure 5.3 is used to convert the input image into several sub-band coefficient images such as Red band, Green band and the Green band. These separate channels will be further used in the next steps.

STEP-4:

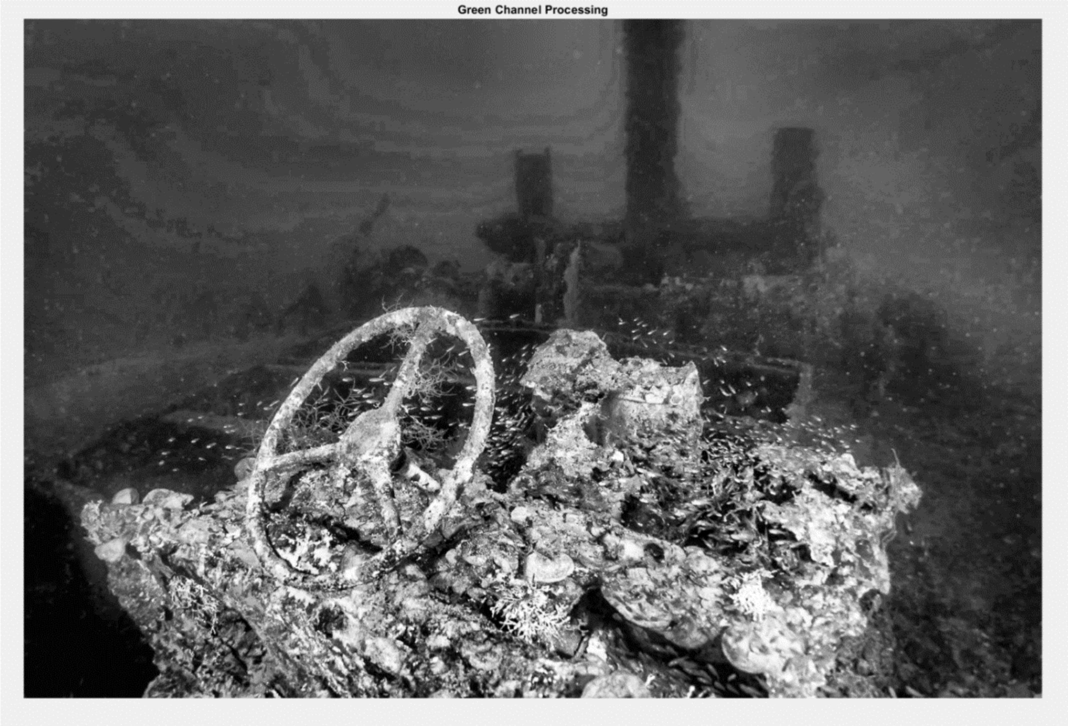


Figure 5.5 Intensity Map of Green Channel

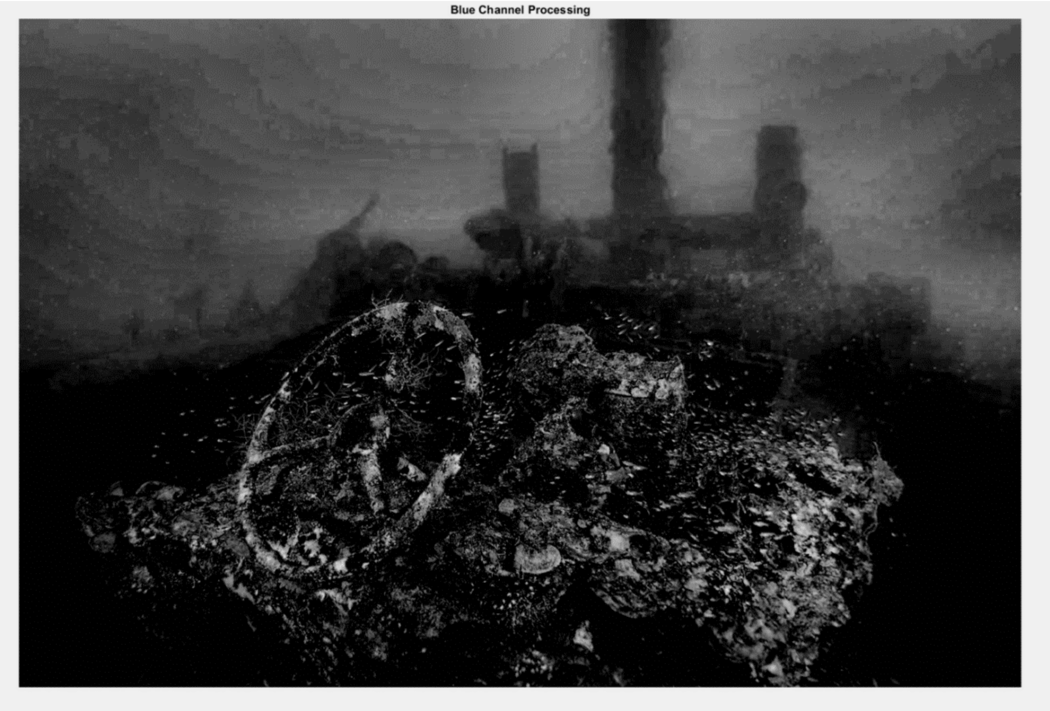


Figure 5.6 Intensity Map of Blue Channel

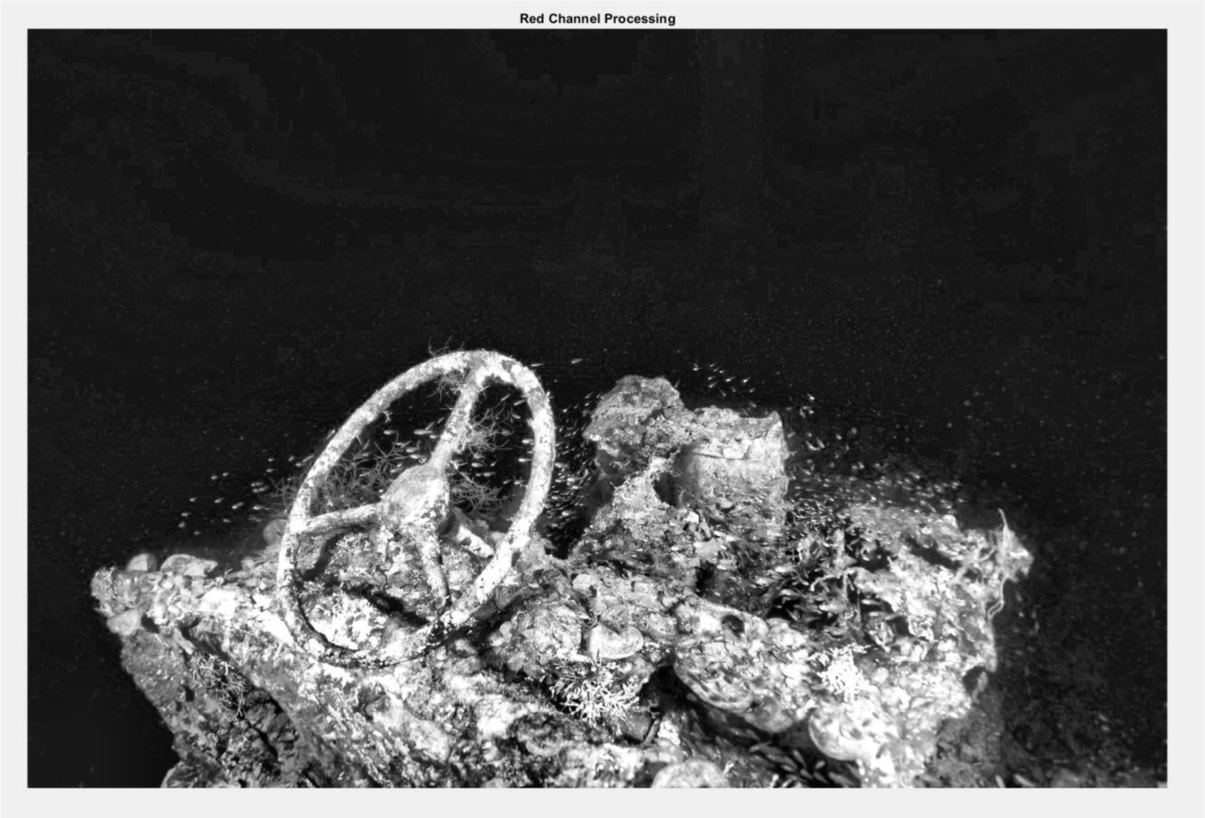


Figure 5.7 Intensity Map of Red Channel

This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values.

Adaptive histogram equalization (AHE) is a computer image processing technique used to improve contrast in images. It differs from ordinary histogram equalization in the respect that the adaptive method computes several histograms, each corresponding to a distinct section of the image, and uses them to redistribute the lightness values of the image. It is therefore suitable for improving the local contrast and enhancing the definitions of edges in each region of an image.

Figure 5.7 consists of three images for every channel present in the RGB colour space. These three channels are separated from the input image and are processed via a BEMD method. The results of the three processed channels are shown in the above Figure.

STEP-5:

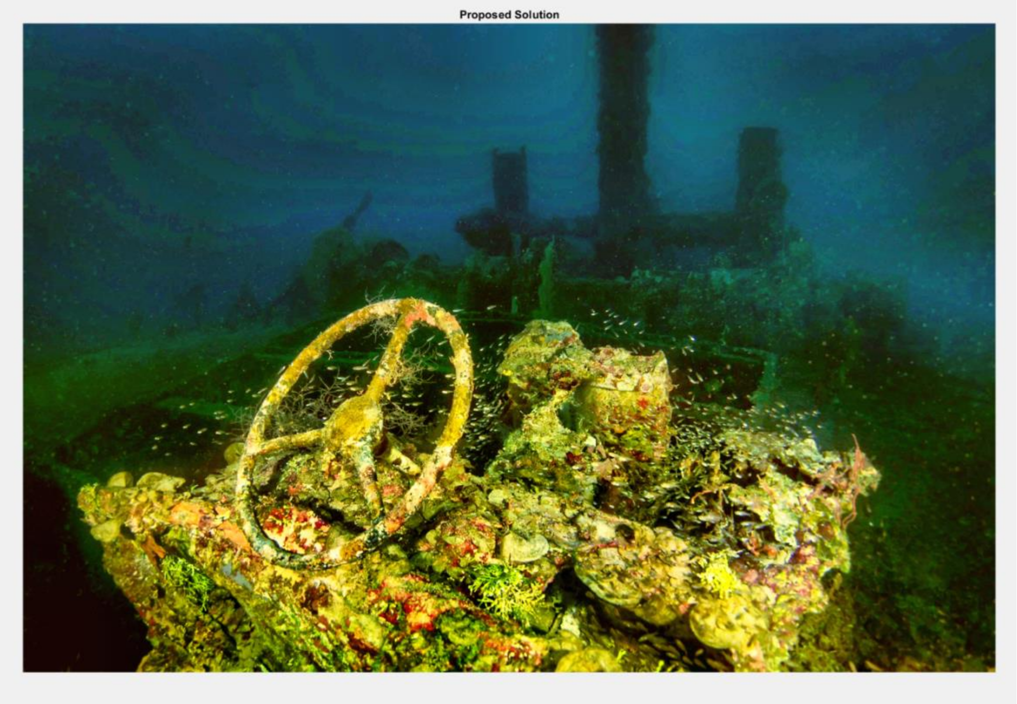


Figure 5.8 Proposed Final Output Image

To get balance the color temperature in your image. so that objects which appear white in person are rendered white in your photo. Proper camera white balance has to take into account the "color temperature" of a light source, which refers to the relative warmth or coolness of white light.

Figure 5.8 represents the Final output Image by Concatenating the individual RGB channels represented in Figure 5.7. The induvial are weighted and the final image is obtained.

CHAPTER 6

ADVANTAGES & LIMITATIONS

6.1 ADVANTAGES:

- The main advantage of this technique is that it improves the input image for the SLAM application.
- The dynamic range of the image is much better.
- It does not require one to extract and match features.
- It provides an advantage as it is costless and the existing low resolution imaging systems can still be utilized.

6.2 LIMITATIONS:

- A disadvantage of the method is that it is indiscriminate. It may increase the contrast of background noise, while decreasing the usable signal. Noise is introduced which cannot be avoided.
- Most images show an increased contrast in the green channel, that is, the images have greener tint.
- Some parts of the images due to over exposure, loose detail and information.
- After zooming the image, the zoomed part quality is reduced.

CHAPTER 7

CONCLUSION

7.1 CONCLUSION:

In the field of image processing, there are several novel techniques are proposed to enhance the visual quality of underwater images. Existing techniques improves the deep-water images to a large extent. But all these methods suffer with some poor-quality metrics. A good quality image will have high PSNR, high GLCM and low MSE. Based on these characteristics the proposed method and the conventional method are compared. The results shown that among both the techniques, White Balanced BEMD gives the better enhancement of quality parameters for underwater images.

7.2 FUTURE SCOPE:

Applying Retinex and complex image enhancement techniques to further improve the results of the implementation of Simultaneous Localization and Mapping (SLAM) for Autonomous Underwater Vehicle (AUV).

REFERENCES:

[1] Padmavathi, G., et al. "Comparison of filters used for underwater image preprocessing." International Journal of Computer Science and Network Security10.1 (2010): 58-65.

[2] Balvant singh, Ravi shankar mishra, Puran gour, “Analysis of contrast enhancement techniques for underwater image” International journal - of computer technology and electronics engineering (IJCTEE) Volume 1, issue 2, pp: 190-195

[3] Ritu singh, Dr. Mantosh Biswas, “Adaptive histogram equalization based fusion Technique for hazy underwater image enhancement” IEEE international conference on computational intelligence and computing research, 2016, pp: 1-5

[4] Sonam Bharal, “Review of underwater image enhancement techniques”, International Research Journal of Engineering and Technology (IRJET), Volume: 02 Issue: 03, June-2015, pp: 340-344.

[5] Sudhansu Mallik, Salman S. Khan, Umesh C. Pati, “Visual Enhancement of Underwater Image by White-Balanced EMD”, 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT) -2017, pp: 1-6.

[6] N. E. Huang et. al, “The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis,” in proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 454, no. 1971. The Royal Society, 1998, pp. 903–995.

[7] Jin Chen et.al “Using Empirical mode decomposition to process marine magneto telluric data,” in Geophysical Journal International, vol. 190,Issue 1,July 2012,pg:293-309.