COLOR BALANCE AND FUSION FOR UNDERWATER IMAGE ENHANCEMENT

A PROJECT REPORT

Submitted by

MEENU.N (810015104046)

NANDHINI.V (810015104051)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY COLLEGE OF ENGINEERING – BIT CAMPUS,

TIRUCHIRAPPALLI

ANNA UNIVERSITY :: CHENNAI 600 025

APRIL 2019

UNIVERSITY COLLEGE OF ENGINEERING,

BIT CAMPUS,

TIRUCHIRAPPALLI-620 024

BONAFIDE CERTIFICATE

Certified that this project report "**Color Balance and Fusion for Under Water Image Enhancement**" is the bonafide work of **"Ms.N.MEENU (810015104046)** and **Ms.V.NANDHINI (810015104051) "**who carried out the project work under my supervision".

Dr. D. Venkatesan	**Mrs.R.S.Rampriya**
HEAD OF THE DEPARTMENT	SUPERVISOR
Assistant Professor	Teaching Faculty
Computer Science & Engineering	Computer Science & Engineering
University College of Engineering,	University College of Engineering,
Anna University-BIT Campus,	Anna University-BIT Campus,
Tiruchirappalli-620 024	Tiruchirappalli-620 024

Submitted for the project Viva voce examination held on ………………

Internal Examiner **External Examiner**

DECLARATION

We hereby declare the work entitled "**COLOR BALANCE AND FUSION FOR UNDERWATER IMAGE ENHANCEMENT"** is submitted in partial fulfillment of the requirement for the award of the degree in B.E., Computer Science and Engineering, University College of Engineering(BIT Campus), Tiruchirappalli, is a record of our own work carried out by us during the academic year 2018-2019 under the supervision and guidance of Mrs.R.S.Rampriya, Teaching Fellow, Department of Computer Science and Engineering, University College of Engineering(BIT Campus), Tiruchirappalli. The extent and source of information are derived from the existing literature and have been indicated through the dissertation at the appropriate places. The matter embodied in this work is original and has not been submitted for the award of any degree, either in this or any other University.

Signature of the Candidate

N.Meenu (810015104046)

V.Nandhini (810015104051)

I certify that the declaration made above by the candidate is true.

Signature of the Guide

Mrs.R.S.Rampriya

Teaching Fellow,

Department of CSE,

University College of Engineering,

BIT Campus, Anna University,

Tiruchirappalli-620 024.

ACKNOWLEDGEMENT

I would like to thank almighty God for giving me the strength, ability, health to complete the project and giving the best all through my life.

I thank my beloved parents and friends, for their full support in my career development of this project.

I would like to convey my heartfelt thanks to our honorable Dean **Dr. T. SENTHILKUMAR,** Associate Professor for having provided me with all required facilities to complete my project without hurdles.

I would like to express my sincere thanks and deep sense of gratitude to guide **Dr. D. VENKATESAN,** Assistant Professor and Head, Department of Computer Science and Engineering, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of this project work.

I would like to thank my project guide **Mrs. R.S. RAMPRIYA,** Teaching Fellow, Department of Computer Science and Engineering, for his valuable guidance throughout the phase of the project. It is our responsibility to thank our project coordinators **Mr.P.KARTHIKEYAN** and **Mr. C.SANKAR RAM,** Assistant Professor, Department of Computer science and Engineering for his constant inspiration that he has all through the project period.

I would like to thank **Mr. C. SURESH KUMAR,** Teaching Fellow, Department of Computer Science and Engineering, for his encouragement for this work.

I extend my thanks to all other teaching and non-teaching staffs for their encouragement and support.

ABSTRACT

Underwater images suffer from poor visibility resulting from the attenuation of the propagated light, mainly due to absorption and scattering effects. The absorption substantially reduces the light energy, while the scattering causes changes in the light propagation direction. Proposed system introduces a novel approach to remove the haze in underwater images based on a single image captured with a conventional camera. Our approach builds on the fusion of multiple inputs, but derives the two inputs to combine by correcting the contrast and by sharpening a white-balanced version of a single native input image. The white balancing stage aims at removing the color cast induced by underwater light scattering, so as to produce a natural appearance of the sub-sea images. Image fusion is considered to enhance the edges and details of the scene, to mitigate the loss of contrast resulting from back scattering.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 IMAGE PROCESSING

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually Image Processing system includes treating images as two-dimensional signals while applying already set signal processing methods to them. It is among rapidly growing technologies today, with its applications in various aspects of a business. Image Processing forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps.

- Importing the image with optical scanner or by digital photography.
- Analyzing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs.
- Output is the last stage in which result can be altered image or report that is based on image analysis.

Purpose of Image processing

The purpose of image processing is divided into 5 groups. They are:

- Visualization - Observe the objects that are not visible.
- Image sharpening and restoration - To create a better image.
- Image retrieval - Seek for the image of interest.

1.2 TYPES OF IMAGE PROCESSING

The two types of methods used for Image Processing are Analog and Digital Image Processing. Analog or visual techniques of image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. The image processing is not just confined to area that has to be studied but on knowledge of analyst. Association is another important tool in image processing through visual techniques. So, analysts apply a combination of personal knowledge and collateral data to image processing.

Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information, it has to undergo various phases of processing. The three general phases that all types of data have to undergo while using digital technique are Pre- processing, enhancement and display, information extraction.

1.3 IMAGE PROCESSING APPLICATIONS

In recent years, advances in information technology and telecommunications have acted as catalysts for significant developments in the sector of health care. These technological advances have had a particularly strong impact in the field of medical imaging, where film radiographic techniques are gradually being replaced by digital imaging techniques, and this has provided an impetus to the development of integrated hospital information systems and integrated teleradiology services networks which support the digital transmission, storage, retrieval, analysis, and interpretation of distributed multimedia patient records.

One of the many added-value services that can be provided over an integrated teleradiology services network is access to high-performance computing facilities in order to execute computationally intensive image analysis and visualization tasks. In general, currently available products in the field of image processing (IP) meet only specific needs of different end user groups. They either aim to provide a comprehensive pool of ready to use software within a user-friendly and application specific interface for those users that use IP software, or aim for the specialised IP researcher and developer, offering programmer's libraries and visual language tools.

However, we currently lack the common framework that will integrate all prior efforts and developments in the field and at the same time provide added-value features that support and in essence realise what we call a 'service'.

In the case of image processing, these features include: computational resource management and intelligent execution scheduling; intelligent and customizable mechanisms for the description, management, and retrieval of image processing software modules; mechanisms for the "plug-and-play" integration of already existing heterogeneous software modules; easy access and user transparency in terms of software, hardware, and network technologies; sophisticated charging mechanisms based on quality of service; and, methods for the integration with other services available within an integrated health telematics network.

1.4 BLOCK TRUNCATION CODING

Block Truncation Coding (BTC) is a simple image compression technique, introduced by Delp and Mitchell. BTC is based on the conservation of statistical properties.

Although it is a simple technique, BTC has played an important role in the history of image compression. Many image compression techniques have been developed based on BTC. Block truncation coding is a lossy compression method. It achieves 2 bits per pixel (bpp) with low computational complexity. It is easy to implement compared to vector quantization and transform coding. Lema and Mitchell presented a variant of BTC called Absolute Moment Block Truncation Coding (AMBTC). It preserves the higher mean and lower mean of the sub image blocks before and after compression.

However, the bit rate achieved by both BTC and AMBTC is 2 bpp. In order to reduce the bitrate several techniques such as median filtering, vector quantization, interpolation and prediction have been used to code the two statistical moments and the bit plane of BTC. Yung-Gi Wu proposed a probability-based block truncation image bit plane coding. Yu-Chen Hu presented low bit-rate image compression scheme based on AMBTC giving a bit rate up to 0.40 bpp.

Thus, the improvements on BTC are continuing to reduce the bit rate and computational complexity by keeping the image quality to acceptable limit. In this paper we propose an image compression scheme based on AMBTC. This scheme is a combination of three techniques, prediction, bit plane omission, and interpolative bit plane coding.

Experimental results show that the proposed scheme gives good image quality with low computational complexity and low bit rate. In Section 2 we briefly outline the BTC and AMBTC methods.

1.5 IMAGE FUSION

Image Fusion is one of the major research fields in image processing. Image Fusion is a process of combining the relevant information from a set of images, into a single image, wherein the resultant fused image will be more informative and complete than any of the input images.

Image fusion process can be defined as the integration of information from a number of registered images without the introduction of distortion.

It is often not possible to get an image that contains all relevant objects in focus. One way to overcome this problem is image fusion, in which one can acquire a series of pictures with different focus settings and fuse them to produce an image with extended depth of field.

Image fusion techniques can improve the quality and increase the application of these data. One of the important pre-processing steps for the fusion process is image registration, i.e., the coordinate transformation of one image with respect to other. Fusion algorithms are input dependent. Image fusion algorithms can be categorized into different levels: low, middle, and high; or pixel, feature, and decision levels.

The pixel-level method works either in the spatial domain or in the transform domain. Pixel level fusion works directly on the pixels obtained at imaging sensor outputs while feature level fusion algorithms operate on features extracted from the source images. The prerequisite for such an operation is that the images have been acquired by homogeneous sensors, such that the images reproduce similar or comparable physical properties of the scene.

The feature-level algorithms typically segment the image into contiguous regions and fuse the regions together using their properties. The features used may be calculated separately from each image or they may be obtained by the simultaneous processing of all the images. Decision level fusion uses the outputs of initial object detection and classification as inputs to the fusion algorithm to perform the data integration. Both feature level and decision level image fusion may result in inaccurate and incomplete transfer of information.

CHAPTER 2

LITERATURE SURVEY

2.1 AN EFFICIENT UNDERWATER IMAGE ENHANCEMENT USING COLOR CONSTANCY DESKEWING ALGORITHM: R. SWARNALAKSHMI, B. LOGANATHAN

The objective of image enhancement is to recover the perception of information in underwater images for human viewers, or to provide recovered input for other automated image processing techniques. In underwater conditions, the image clarities are corrupted by light inclusion and diffusion. This causes one color to dominate the image. In order to enhance the observation of underwater images, we proposed an approach based on color constancy algorithms.

The objective of this proposed method is twofold (i) Feature Extraction and (ii) color constancy algorithm. Firstly, to capture the image characteristics, the Feature parameterization (e.g. grain size and contrast) and the optimized constancy mapping is extended to incorporate the statistical nature of images. The proposed algorithms are tested on synthetic images as well as real images.

2.2 REGULARIZED IMAGE RECOVERY IN SCATTERING MEDIA: YOAV Y. SCHECHNER *IEEE MEMBER* AND YUVAL AVERBUCH

When imaging in scattering media, visibility degrades as objects become more distant. Visibility can be significantly restored by computer vision methods that account for physical processes occurring during image formation. Nevertheless, such recovery is prone to noise amplification in pixels corresponding to distant objects, where the medium transmittance is low. We present an adaptive filtering approach that counters the above problems: while significantly improving visibility relative to raw images, it inhibits noise amplification. Essentially, the recovery formulation is regularized, where the regularization adapts to the spatially varying medium transmittance. Thus, this regularization does not blur close objects. We demonstrate the approach in atmospheric and underwater experiments, based on an automatic method for determining the medium transmittance.

2.3 FACTORIZING SCENE ALBEDO AND DEPTH FROM A SINGLE FOGGY IMAGE: LOUIS KRATZ KO NISHINO

Atmospheric conditions induced by suspended particles, such as fog and haze, severely degrade image quality. Restoring the true scene colors (clear day image) from a single image of a weather-degraded scene remains a challenging task due to the inherent ambiguity between scene albedo and depth. In this paper, we introduce a novel probabilistic method that fully leverages natural statistics of both the albedo and depth of the scene to resolve this ambiguity.

Our key idea is to model the image with a factorial Markov random field in which the scene albedo and depth are two statistically independent latent layers. We show that we may exploit natural image and depth statistics as priors on these hidden layers and factorize a single foggy image via a canonical Expectation Maximization algorithm with alternating minimization. Experimental results show that the proposed method achieves more accurate restoration compared to state-of-the-art methods that focus on only recovering scene albedo or depth individually.

2.4 A VISION SYSTEM FOR AN UNDERWATER CABLE TRACKER: ALBERTO ORTIZ1, MIQUEL SIM, GABRIEL OLIVER

Nowadays, the surveillance and inspection of underwater installations, such as power and telecommunication cables and pipelines, is carried out by operators that, being on the surface, drive a remotely operated vehicle (ROV) with cameras mounted over it. This is a tedious and high time consuming task, easily prone to errors mainly because of loss of attention or fatigue of the human operator.

Besides, the complexity of the task is increased by the lack of quality of typical seabed images, which are mainly characterized by blurring, non-uniform illumination, lack of contrast and instability in the vehicle motion. In this study, the development of a vision system guiding an autonomous underwater vehicle (AUV) able to detect and track automatically an underwater power cable laid on the seabed is the main concern.

The vision system that is proposed tracks the cable with an average success rate above 90%. The system has been tested using sequences coming from a video tape obtained in several tracking sessions of various real cables with a ROV driven from the surface. These cables were installed several years ago, so that the images do not present highly contrasted cables over a sandy seabed; on the contrary, these cables are partially covered in algae or sand, and are surrounded by other algae and rocks, thus making the sequences highly realistic.

2.5 NIGHT-TIME DEHAZING BY FUSION: COSMIN ANCUTI, CODRUTA O. ANCUTI, CHRISTOPHE DE VLEESCHOUWER AND ALAN C. BOVIK

We introduce an effective technique to enhance night-time hazy scenes. Our technique builds on multi-scale fusion approach that use several inputs derived from the original image. Inspired by the dark-channel we estimate night-time haze computing the air light component on image patch and not on the entire image.

We do this since under night-time conditions, the lighting generally arises from multiple artificial sources, and is thus intrinsically non-uniform.

Selecting the size of the patches is non-trivial, since small patches are desirable to achieve fine spatial adaptation to the atmospheric light, this might also induce poor light estimates and reduced chance of capturing hazy pixels.

For this reason, we deploy multiple patch sizes, each generating one input to a multiscale fusion process. Moreover, to reduce the glowing effect and emphasize the finest details, we derive a third input. For each input, a set of weight maps are derived so as to assign higher weights to regions of high contrast, high saliency and small saturation.

Finally, the derived inputs and the normalized weight maps are blended in a multi-scale fashion using a Laplacian pyramid decomposition. The experimental results demonstrate the effectiveness of our approach compared with recent techniques both in terms of computational efficiency and quality of the outputs.

2.6 IMAGE ENHANCING TECHNIQUES BASED ON FUSION: METODE DE IMBUNATATIRE A IMAGINILOR BAZATE PE FUZIUNE CODRUTA, ORNIANA ANCUTI

The present habilitation thesis summarizes the research contributions of the candidate obtained between 2011 to this date. The most significant research activity and the obtained results are presented structured in several parts that represent original contributions in the field of image processing and computational photography: image dehazing (day-time single image dehazing, night-time single image dehazing, dehazing evaluation dataset), underwater images enhancement, image decolorization and single scale fusion technique for effectively merging images. Image dehazing deals with the problem of enhancing the visibility in terms of color and details for images degraded by haze.

In outdoor environments, haze phenomena appear when the light reflected from object surfaces is scattered due to the impurities of the aerosol, or due to the presence of fog and haze. The yielding hazy images are characterized by poor contrast, lower saturation and additional noise.

2.7 HIERARCHICAL RANK-BASED VEILING LIGHT ESTIMATION FOR UNDERWATER DEHAZING: SIMON EMBERTON, LARS CHITTKA, ANDREA CAVALLARO

Current dehazing approaches are often hindered when scenes contain bright objects which can cause veiling light and transmission estimation methods to fail. This paper introduces a single image dehazing approach for underwater images with novel veiling light and transmission estimation steps which deal with issues arising from bright objects. We use features to hierarchically rank regions of an image and to select the most likely veiling light candidate.

A region-based approach is used to find optimal transmission values for areas that suffer from oversaturation. We also locate background regions through super pixel segmentation and clustering, and adapt the transmission values in these regions so to avoid artefacts. We validate the performance of our approach in comparison to the state of the art in underwater dehazing through subjective evaluation and with commonly used quantitative measures.

2.8 SINGLE IMAGE SUPER-RESOLUTION FROM TRANSFORMED SELF-EXEMPLARS: JIA-BIN HUANG, ABHISHEK SINGH, AND NARENDRA AHUJA

Self-similarity based super-resolution (SR) algorithms are able to produce visually pleasing results without extensive training on external databases. Such algorithms exploit the statistical prior that patches in a natural image tend to recur within and across scales of the same image. However, the internal dictionary obtained from the given image may not always be sufficiently expressive to cover the textural appearance variations in the scene. In this paper, we extend self-similarity based SR to overcome this drawback.

We expand the internal patch search space by allowing geometric variations. We do so by explicitly localizing planes in the scene and using the detected perspective geometry to guide the patch search process. We also incorporate additional affine transformations to accommodate local shape variations. We propose a compositional model to simultaneously handle both types of transformations.

We extensively evaluate the performance in both urban and natural scenes. Even without using any external training databases, we achieve significantly superior results on urban scenes, while maintaining comparable performance on natural scenes as other state-of-the-art SR algorithms.

2.9 CENTRAL COMMAND ARCHITECTURE FOR HIGH-ORDER AUTONOMOUS UNMANNED AERIAL SYSTEMS

This paper is the first in a two-part series that introduces an easy-to-implement central command architecture for high-order autonomous unmanned aerial systems. This paper discusses the development and the second paper presents the flight test results. As shown in this paper, the central command architecture consists of a central command block, an autonomous planning block, and an autonomous flight controls block.

The central command block includes a staging process that converts an objective into tasks independent of the vehicle (agent). The autonomous planning block contains a non-iterative sequence of algorithms that govern routing, vehicle assignment, and deconfliction. The autonomous flight controls block employs modern controls principles, dividing the control input into a guidance part and a regulation part. A novel feature of high-order central command, as this paper shows, is the elimination of operator-directed vehicle tasking and the manner in which deconfliction is treated. A detailed example illustrates different features of the architecture.

2.10 SINGLE SCALE IMAGE DEHAZING BY MULTI SCALE FUSION MRS.A. DYANAA, MS.SRRUTHI THIAGARAJAN VISVANATHAN, MS. VARSHA CHANDRAN

Because of the strong and successfulness, tampering of digital images depends upon the flexible architecture for a derivative estimation has become an important method to a spectacle for a large number of approaches. Regulated on an inherent range, the tampering of digital image procedure is varied and carried out. The beginning of a bridging origination which dilutes into a simple procedure. Thus, the explanation of imaging is near to tampering of digital images procedure which discards spare estimations. The statement was extracted into large situation intense design.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing underwater dehazing techniques can be grouped in several classes. An important class corresponds to the methods using specialized hardware. For instance, the divergent-beam Under Water Lidar Imaging (UWLI) system uses an optical/laser-sensing technique to capture turbid underwater images. Unfortunately, these complex acquisition systems are very expensive, and power consuming. A second class consists in polarization-based methods.

These approaches use several images of the same scene captured with different degrees of polarization, as obtained by rotating a polarizing filter fixed to the camera. While being effective in recovering distant regions, the polarization techniques are not applicable to video acquisition, and are therefore of limited help when dealing with dynamic scenes.

A third class of approaches employs multiple images or a rough approximation of the scene model. Since this additional information (images and depth approximation) is generally not available, these methods are impractical for common users. A fourth class of methods exploits the similarities between light propagation in fog and under water. Several algorithms that specifically restore underwater images based on Dark Channel Prior (DCP) have been introduced.

The DCP has initially been proposed for outdoor scenes dehazing. It assumes that the radiance of an object in a natural scene is small in at least one of the color components, and consequently defines regions of small transmission as the ones with large minimal value of colors.

Disadvantages

- Complex acquisition systems are very expensive, and power consuming.
- Existing techniques are not applicable to video acquisition, and are therefore of limited help when dealing with dynamic scenes.

3.2 PROPOSED SYSTEM

Proposed image enhancement approach adopts a twostep strategy, combining white balancing and image fusion, to improve underwater images without resorting to the explicit inversion of the optical model. In our approach, white balancing aims at compensating for the color cast caused by the selective absorption of colors with depth, while image fusion is considered to enhance the edges and details of the scene, to mitigate the loss of contrast resulting from backscattering.

White-balancing aims at improving the image aspect, primarily by removing the undesired color castings due to various illumination or medium attenuation properties. In underwater, Gray-World algorithm achieves good visual performance for reasonably distorted underwater scenes. However, a deeper investigation dealing with extremely deteriorated underwater scenes reveals that most traditional methods perform poorly.

Gray World algorithm will be adopted to compute the white balanced image. A pair of inputs is introduced to respectively enhance the color contrast and the edge sharpness of the white-balanced image, and the weight maps are defined to preserve the qualities and reject the defaults of those inputs, i.e. to overcome the artifacts induced by the light propagation limitation in underwater medium. Our underwater dehazing technique consists in three main steps: inputs derivation from the white balanced underwater image, weight maps definition, and multi-scale fusion of the inputs and weight maps. White balancing suffers from noticeable effects since the absorbed colors are difficult to be recovered. As a result, to obtain our first input we perform a gamma correction of the white balanced image version. Gamma correction aims at correcting the global contrast and is relevant since, in general, white balanced underwater images tend to appear too bright.

This correction increases the difference between darker/lighter regions at the cost of a loss of details in the under-/over-exposed regions. A second input that corresponds to a sharpened version of the white balanced image. Therefore, we follow the unsharp masking principle, in the sense that we blend a blurred or unsharp (here Gaussian filtered) version of the image with the image to sharpen.

The weight maps are used during blending in such a way that pixels with a high weight value are more represented in the final image. The pyramid representation decomposes an image into a sum of bandpass images. In practice, each level of the pyramid does filter the input image using a low-pass Gaussian kernel *G*, and decimates the filtered image by a factor of 2 in both directions.

It then subtracts from the input an up-sampled version of the low-pass image, thereby approximating the (inverse of the) Laplacian, and uses the decimated low-pass image as the input for the subsequent level of the pyramid. Multi-scale fusion is motivated by the human visual system, which is very sensitive to sharp transitions appearing in smooth image patterns, while being much less sensitive to variations/artifacts occurring on edges and textures.

Advantages

- Retrieve image with high quality.
- Image fusion and weight maps acquire less energy consumption.

CHAPTER 4

SYSTEM REQIREMENTS

4.1 HARDWARE REQUIREMENTS

- CPU type : Intel Pentium 4
- Clock speed : 3.0 GHz
- Ram size : 512 MB
- Hard disk capacity : 40 GB
- Monitor type : 15 Inch color monitor
- Keyboard type : internet keyboard

4.2 SOFTWARE REQUIREMENTS

- Operating System : Windows OS
- Language : MATLAB

4.3 SOFTWARE SPECIFICATION:

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
- Modeling, 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 noninteractive 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 uses software developed by the LAPACK and ARPACK projects, which together represent 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.

Toolboxes

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users 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.

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, 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 eigenvalues, 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 complete large and complex application programs.

Handle Graphics

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces.

MATLAB functions:

A MATLAB “function” is a MATLAB program that performs a sequence of operations specified in a text file (called an m-file because it must be saved with a file extension of *.m). A function accepts one or more MATLAB variables as inputs, operates on them in some way, and then returns one or more MATLAB variables as outputs and may also generate plots, etc. Some functions are

Imread() – Reading the image from the graphics file.

A = imread(filename, fmt) reads a grayscale or color image from the file specified by the string**filename**. If the file is not in the current folder, or in a folder on the MATLAB path, specify the full pathname.

The text string **fmt** specifies the format of the file by its standard file extension. For example, specify'gif' for Graphics Interchange Format files. To see a list of supported formats, with their file extensions, use the imformats function. If **imread** cannot find a file named filename, it looks for a file named filename.fmt.The return value A is an array containing the image data. If the file contains a grayscale image, A is an M-by-N array. If the file contains a truecolor image, A is an M-by-N-by-3 array.

For TIFF files containing color images that use the CMYK color space, A is an M-by-N-by-4 array. The class of A depends on the bits-per-sample of the image data, rounded to the next byte boundary. For example**, imread** returns 24-bit color data as an array of uint8 data because the sample size for each color component is 8 bits.

[X,map]=imread(...) reads the indexed image in filename into X and its associated colormap into map. Colormap values in the image file are automatically rescaled into the range [0,1].

Image- write image to graphics file

imwrite(A,filename,fmt) writes the image A to the file specified by filename in the format specified by fmt.

A can be an M-by-N (grayscale image) or M-by-N-by-3 (truecolor image) array, but it cannot be an empty array. For TIFF files, A can be an M-by-N-by-4 array containing color data that uses the CMYK color space. For GIF files, A can be an M-by-N-by-1-by-P array containing grayscale or indexed images — RGB images are not supported. For information about the class of the input array and the output image.

Filename is a string that specifies the name of the output file. fmt can be any of the text strings listed. This list of supported formats is determined by the MATLAB image file format registry. See imformats for information about this registry **imwrite(X,map,filename,fmt)** writes the indexed image in X and its associated colormap map to filename in the format specified by fmt. If X is of class uint8 or uint16, imwrite writes the actual values in the array to the file. If X is of class double, imwrite offsets the values in the array before writing, using uint8(X–1). map must be a valid MATLAB colormap.

Note that most image file formats do not support colormaps with more than 256 entries. When writing multiframe GIF images, X should be an 4-dimensional M-by-N-by-1-by-P array,where P is the number of frames to write.

imwrite(...,filename) writes the image to filename, inferring format to use from the file name’s extension.

imwrite(...,Param1,Val1,Param2,Val2...) specifies parameters that control various characteristics of the output file for HDF, JPEG, PBM, PGM, PNG, PPM, and TIFF files. For example, if you are writing a JPEG file, you can specify the quality of the output image. For the lists of parameters available for each format.

4.4 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 include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

4.5 SIMULINK

Real-Time Workshop is a program that allows you to generate C code from your block diagrams and to run it on a variety of real-time systems.

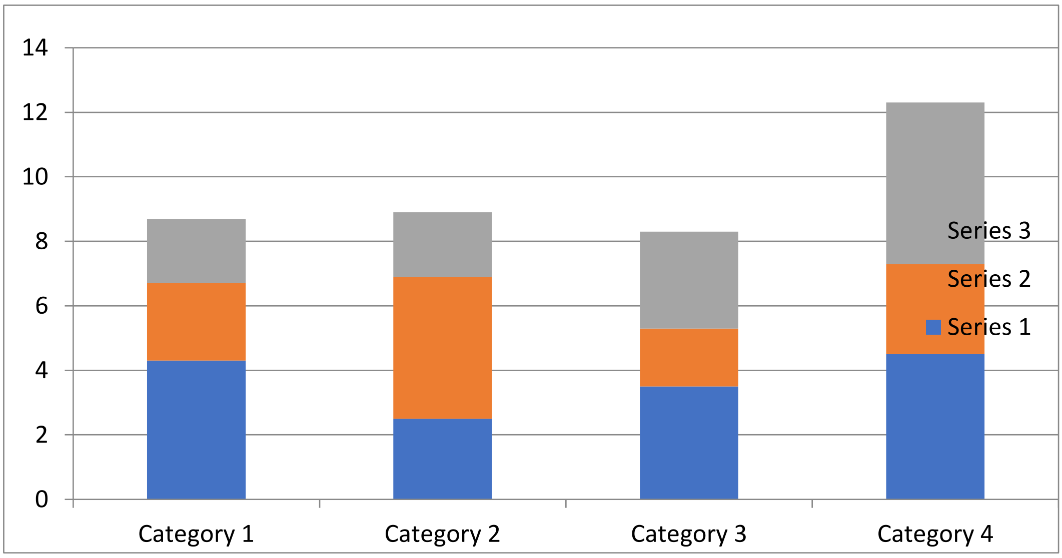


Figure 4.5 Simulink-Mouse driven program

Simulink, a companion program to MATLAB, is an interactive system for simulating nonlinear dynamic systems. It is a graphical mouse-driven program that allows you to model a system by drawing a block diagram on the screen and manipulating it dynamically.

It can work with linear, nonlinear, continuous-time, discrete-time, multirate, and hybrid systems. Block sets are add-ons to Simulink that provide additional libraries of blocks for specialized applications like communications, signal processing, and power systems.

Tightly integrated with Simulink and MATLAB, Stateflow provides Simulink users with an elegant solution for designing embedded systems by giving them an efficient way to incorporate complex control and supervisory logic within their Simulink models.

With Stateflow, you can quickly develop graphical models of event-driven systems using finite state machine theory, state chart formalisms, and flow diagram notation. Together, State flow and Simulink serve as an executable specification and virtual prototype of your system design.

4.6 ANALYSIS STUDY

The proposed work was implemented on MATLAB 2012b working on a system with an i3 processor having 4GB RAM. For validating the effectiveness of this work, we have taken two sets of images.

The first set consisted of 6 different images with 6 different diseases is selected from planet natural and second dataset consists of about 8 images are selected. The result part is divided into two categories (A) To correctly segment/identify the infected area on plant leaf for a disease and (B) To classify the type of leaf disease.

The performance evaluation of the proposed work for correctly identifying the affected area or disease on the plant leaf is evaluated using two quantitative evaluation parameters that are based on the statistical performance of the ground truth image and segmented image. The parameters are specificity and sensitivity.

The most critical part is the classification of diseases based on some attributes associated with them. The performance of the proposed work for correctly classifying diseases is done by using two entropy functions known as Validation evaluation partition coefficient and Validation evaluation partition entropy.

$$Specificity=TNTN+FP$$

$$Sensitivity=TPTP+FN$$

where True Positive (TP) = no. of pixels exactly classified, False Positive (FP) = no. of pixels incorrectly classified, True Negative (TN) = no. of pixels exactly misclassified, and False Negative (FN) = no. of pixels incorrectly misclassified. The value of specificity and sensitivity lies between 0 and 1 when result is equal to 1 means perfect segmentation.

$$Vpc= \Sigma\Sigma uik2NNi=1Kk=1$$

$$Vpe= -\Sigma\Sigma uik\log\,(uik)=1Kk=1$$

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

The architecture of the proposed system is given as follows. This architecture will provide the overall representation of the system.

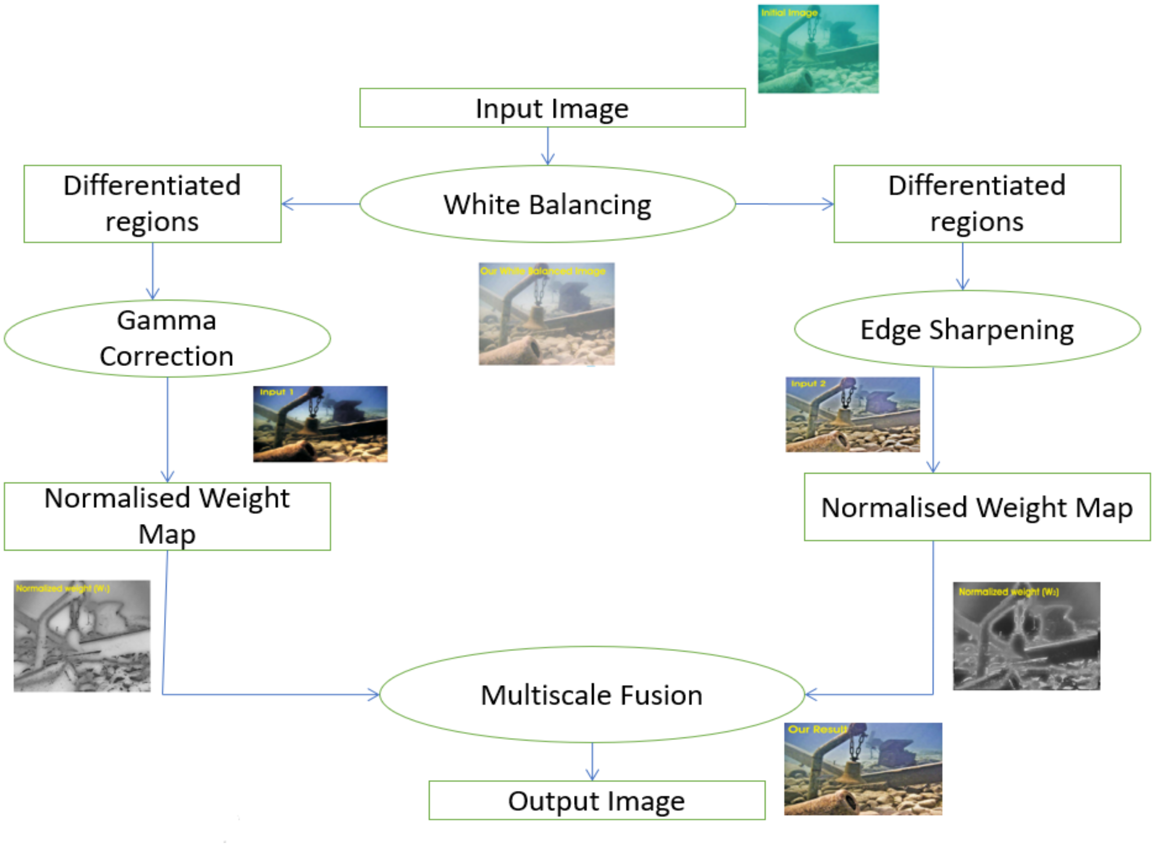


Figure 5.1 Outline of this approach used in this paper. Pictorial representation are given for the input stage of the workflow.

A brief outline of the methodological framework is given in Figure 5.1. The proposed architecture includes five main stages: White balancing, Gamma correction, Edge sharpening, Multiscale fusion.

CHAPTER 6

SYSTEM MODULES

MODULES

The proposed system consists of four modules which are as follows.

1. White balancing
2. Gamma correction
3. Edge sharpening
4. Multiscale fusion

6.1 WHITE BALANCING

The image enhancement approach adopts a two step strategy, combining white balancing and image fusion, to improve underwater images without resorting to the explicit inversion of the optical model. In our approach ,white balancing aims at compensating for the color cast caused by the selective absorption of colors with depth, while image fusion is considered to enhance the edges and details of the scene, to mitigate the loss of contrast resulting from back scattering.

White-balancing aims at improving the image aspect, primarily by removing the undesired color castings due to various illumination or medium attenuation properties. In underwater, the perception of color is highly correlated with the depth, and an important problem is the green-bluish appearance that needs to be rectified.

As the light penetrates the water, the attenuation process affects selectively the wavelength spectrum, thus affecting the intensity and the appearance of a colored surface (see Section II). Since the scattering attenuates more the long wavelengths than the short ones, the color perception is affected as we go down in deeper water. In practice, the attenuation and the loss of color also depends on the total distance between the observer and the scene.

6.2 GAMMA CORRECTION

The color correction is critical in underwater, the first apply our white balancing technique to the original image. This step aims at enhancing the image appearance by discarding unwanted color casts caused by various illuminants. In water deeper than 30 ft, white balancing suffers from noticeable effects since the absorbed colors are difficult to be recovered.

As a result, to obtain our first input to perform a gamma correction of the white balanced image version. Gamma correction aims at correcting the global contrast and is relevant since, in general, white balanced underwater images tend to appear too bright.

This correction increases the difference between darker/lighter regions at the cost of a loss of details in the under-/over-exposed regions. To compensate for this loss, we derive a second input that corresponds to a sharpened version of the white balanced image. Therefore, we follow the unsharp masking principle.

6.3 EDGE SHARPENING

The perception of color is highly correlated with the depth, and an important problem is the green-bluish appearance that needs to be rectified. As the light penetrates the water, the attenuation process affects selectively the wavelength spectrum thus affecting the intensity and the appearance of a colored surface.

Since the scattering attenuates more the long wavelengths than the short ones, the color perception is affected as we go down in deeper water.

In practice, the attenuation and the loss of color also depends on the total distance between the observer and the scene.

6.4 MULTISCALE FUSION

The weight maps are used during blending in such a way that pixels with a high weight value are more represented in the final image They are thus defined based on a number of local image quality or saliency metrics.

Laplacian contrast weight estimates the global contrast by computing the absolute value of a Laplacian filter applied on each input luminance channel. This straightforward indicator was used in different applications such as tone mapping and extending depth of field since it assigns high values to edges and texture.

For the underwater dehazing task, however, this weight is not sufficient to recover the contrast, mainly because it cannot distinguish much between a ramp and flat regions. To handle this problem, the introduce an additional and complementary contrast assessment metric. Saliency weight aims at emphasizing the salient objects that lose their prominence in the underwater scene. To measure the saliency level, we have employed the saliency estimator.

This computationally efficient algorithm has been inspired by the biological concept of center-surround contrast. However, the saliency map tends to favor highlighted areas (regions with high luminance values). To overcome this limitation, we introduce an additional weight map based on the observation that saturation decreases in the highlighted regions.

Saturation weight (WSat) enables the fusion algorithm to adapt to chromatic information by advantaging highly saturated regions. This weight map is simply computed (for each input Ik) as the deviation (for every pixel location) between the Rk ,Gk and Bk color channels and the luminance Lk of the kth input:

WSat = 1/3 (Rk −Lk)2+(Gk−Lk)2+(Bk−Lk)2

In practice, for each input, the three weight maps are merged in a single weight map as follows. For each input k, an aggregated weight map Wk is first obtained by summing up the three WL, WS, and WSat weight maps. The K aggregated maps are then normalized on a pixel-per-pixel basis, by dividing the weight of each pixel in each map by the sum of the weights of the same pixel over all maps. Formally, the normalized weight maps W̄k are computed for each input as

W̄k = (Wk + δ)/(Kk=1 Wk + K.δ), with δ denoting a small regularization term that ensures that each input contributes to the output. δ is set to 0.1 in the rest of the paper. The normalized weights of corresponding weights are shown at the bottom

6.4.1 DISCRETE WAVELET TRANSFORM

The discrete wavelet transformation is used to analyze the two dimensions image where it has the ability to divide the image into four main areas. The distributed data depends on the degree of the important data. The data area is called the low-low (LL) sub band. Occasionally, it is named as an approximation area. The rest of the regions are known as the detail. The present sub bands are as follow: high-low (HL), low high (LH) and the high-high (HH). In brief, they are indicated as (DWT) .

The wavelet analysis is permitted in using the long periods of time where the ambition is to get more accurate information from low frequency, and high frequency from the shorter zones .The important data reside in the low frequency because it gives you the opportunity to get a signal of an identity. When you go to the high frequency content, it is characterized by high accuracy. This is called mechanism sub band coding.

This class of DWT refers to the place of analysis via the approximation area at level j in four zones. Specifically, level j + 1 consists of approximation and the details. The details are distributed in three zones: horizontal, vertical, and diagonal

The HSI model provides a wider color range by controlling the color elements of the image. The Saturation (S) and Intensity (I) are the element that generates the wider color range. In a situation when we have the blue color element in the image it is controlled by the ‘S’ and ‘I’ value in order to create the range from pale blue to deep blue, for instance. Using this technique, we can control the contrast ratio in underwater images either by decreasing or increasing the value.

This is carried out by employing a histogram of the digital values for an image and redistributing the stretching value over the image variation of the maximum range of the possible values Furthermore linear stretching from ‘S’ value can provide stronger values to each range by looking at the less output values. Here a percentage of the saturating image can be controlled in order to perform better visual displays The contrast stretching algorithm is used to enhance the contrast of the image. This is carried out by stretching the range of the color values to make use of all possible values.

The contrast stretching algorithm is applied to color images, each channel is stretched using the same scaling to maintain the correct color ratio. The first step is to balance the red and green channel to be slightly the same to the blue channel.

CHAPTER 7

CONCLUSION AND FUTURE WORK

The proposed system of underwater image enhancement algorithm both on light illumination and shades color models to enhance underwater images. In order to demonstrate the usefulness of our approach, we have developed an interactive software tool to be used for underwater image enhancement.

Our fusion-based algorithm has the advantage to employ only a reduced set of parameters that can be automatically set. Specifically, the white balancing process relies on the single parameter α, which is set to 1 in all our experiments. For the multi-scale fusion, the number of decomposition levels depends on the image size, and is defined so that the size of the smallest resolution reaches a few tenths of pixels.

Our strategy builds on the fusion principle and does not require additional information than the single original image. We have shown in our experiments that our approach is able to enhance a wide range of underwater images (e.g. different cameras, depths, light conditions) with high accuracy, being able to recover important faded features and edges.

APPENDIX

A. SAMPLE CODE

Sample code- White Balancing

```
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @RUN_MAIN_OpeningFcn, ...
                   'gui_OutputFcn',  @RUN_MAIN_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before RUN_MAIN is made visible.
function RUN_MAIN_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to RUN_MAIN (see VARARGIN)

% Choose default command line output for RUN_MAIN
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes RUN_MAIN wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
```

```
function varargout = RUN_MAIN_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
global fname;
[fn pn] = uigetfile('*.jpg','Select Image File');
complete = strcat(pn,fn);
fname = complete;
pic1=imread (complete);
imshow(pic1,'Parent',handles.axes1);
title(handles.axes1,'Input Image');


% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
global fname;
global white;
I = imread(fname);
white = whitebalance(I);
imshow(white,'Parent',handles.axes2);
title(handles.axes2,'White Balance Image');
J =white;
p = 93:0.5:94;
figure;
for i=1:length(p)
   O = PerformAWB(I, p(i));
   subplot(1,3,1), imshow(I), title('Input Image');
   subplot(1,3,2), imshow(J), title('White Balance Image');
   subplot(1,3,3), imshow(O), title('Gamma Correction Image');
end
RESULTS
function pushbutton4_Callback(hObject, eventdata, handles)
```

Sample code-Gamma correction

```
function O = PerformAWB(I, p)

if nargin == 1
   p = 96;
end
```

```
p = 100-p; % always the top % is considered
szI = size(I);

if(length(szI) < 3)
   error ('Input image should be in RGB color space')
end

   [iR, iG, iB] = EstimateIlluminantRGB(I, p);


   iEstm = [iR, iG, iB];
   CCT_Estm = EstimateCCT(iEstm);


   iRef = [iG, iG, iG];
   CCT_Ref = EstimateCCT(iRef);


   K = ComputeGainFactorMatrix(iEstm);
   T = ComputeOffsetMatrix(K, CCT_Estm, CCT_Ref);



   O = PerformWhiteBalanceCorrection(I, K, T);


end

function [Rc, Gc, Bc] = EstimateIlluminantRGB(I, p)

   R = I(:,:,1);
   G = I(:,:,2);
   B = I(:,:,3);


   Rc = EstimateIlluminantGrey(R, p);
   Gc = EstimateIlluminantGrey(G, p);
   Bc = EstimateIlluminantGrey(B, p);
end


function Ic = EstimateIlluminantGrey(I, p)
   Ic = 0;
```

```
    L = 256;
    sz = size(I);

    pxlTh = (p*sz(1)*sz(2))/100;
    histI = imhist(I);

    Imin = min(min(I));
    Imax = max(max(I));


    for k=Imin:(Imax-1)


        j = double(k+1);
        cnt1 = sum(histI(j:L));


        j = j+1;
        cnt2 = sum(histI(j:L));

        if( (cnt1 > pxlTh) && (cnt2 < pxlTh) )
            Ic = k;
             break;
           end
      end
end


function CCT = EstimateCCT(iEstm)

    A0 = -949.86315;
    A1 = 6253.80338;
    A2 = 28.70599;
    A3 = 0.00004;

    t1 = 0.92159;
    t2 = 0.20039;
    t3 = 0.07125;

    xe = 0.3366;
    ye = 0.1735;
```

```
    XYZ_Conv_matrix = [ 0.4124 0.3576 0.1805;
                        0.2126 0.7152 0.0722;
                        0.0193 0.152 0.9505];

    XYZ = XYZ_Conv_matrix * double(iEstm');

    x = XYZ(1) / (sum(XYZ));
    y = XYZ(2) / (sum(XYZ));

    H = -((x-xe)/(y-ye));

    CCT = A0 + (A1*exp(H/t1)) + (A2*exp(H/t2)) + (A3*exp(H/t3));

end


function [K] = ComputeGainFactorMatrix(iEstm)

    iEstm = double(iEstm);

    iEstm_R = iEstm(1);
    iEstm_G = iEstm(2);
    iEstm_B = iEstm(3);

    iRef_R = iEstm_G;
    iRef_G = iEstm_G;
    iRef_B = iEstm_G;

    Kr = iRef_R / iEstm_R;
    Kg = iRef_G / iEstm_G;
    Kb = iRef_B / iEstm_B;

    K = [Kr, 0, 0;
         0, Kg, 0;
         0, 0, Kb];

end

function T = ComputeOffsetMatrix(K, CCT_Estm, CCT_Ref)

    A = 100;
```

```
    Kr = K(1,1);
    Kb = K(3,3);

    Tr = max(1, (CCT_Estm - CCT_Ref)/A ) * (Kr-1);
    Tg = 0;
    Tb = max(1, (CCT_Ref - CCT_Estm)/A ) * (Kb-1);

    T = [Tr; Tg; Tb];

end

function O = PerformWhiteBalanceCorrection(I, K, T)

    sz = size(I);
    O = uint8(zeros(sz));

    for x = 1:sz(1)
        for y = 1:sz(2)
            Fxy = double([I(x,y,1), I(x,y,2), I(x,y,3)]');
            FWB = K * Fxy + T;

            for p=1:3
                O(x,y,p) = uint8(FWB(p));
            end
        end
    end

end
```

Sample code-Multiscale fusion

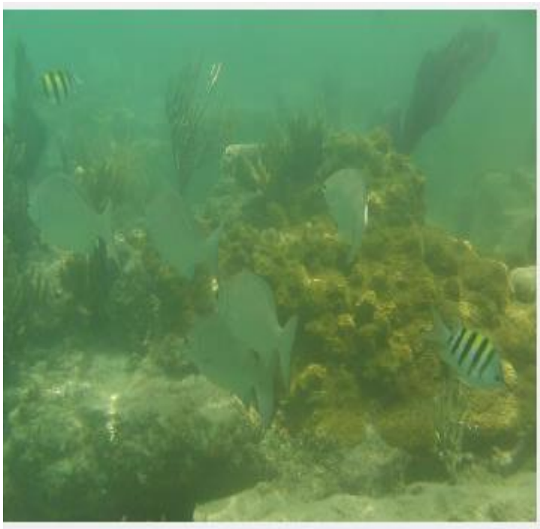
```
function varargout = RESULTS(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @RESULTS_OpeningFcn, ...
                   'gui_OutputFcn',  @RESULTS_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end
```

```
if nargout
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
function RESULTS_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
global fname;
img = imread(fname);
global white;
img1 = SimplestColorBalance(white);
imshow(img1,'Parent',handles.axes1);
title(handles.axes1,'Color Balance Image');
lab1 = rgb_to_lab(img1);
imshow(lab1,'Parent',handles.axes2);
title(handles.axes2,'Lab Image');
lab2 = lab1;
lab2(:, :, 1) = adapthisteq(lab2(:, :, 1));
img2 = lab_to_rgb(lab2);
R1 = double(lab1(:, :, 1)) / 255;
WL1 = abs(imfilter(R1, fspecial('Laplacian'), 'replicate', 'conv'));
h = 1/16* [1, 4, 6, 4, 1];
WC1 = imfilter(R1, h'*h, 'replicate', 'conv');
WC1(find(WC1 > (pi/2.75))) = pi/2.75;
WC1 = (R1 - WC1).^2;
WS1 = saliency_detection(img1);
imshow(WS1,[],'Parent',handles.axes3);
title(handles.axes3,'Saliency Map');
sigma = 0.25;
aver = 0.5;
WE1 = exp(-(R1 - aver).^2 / (2*sigma^2));
R2 = double(lab2(:, :, 1)) / 255;
WL2 = abs(imfilter(R1, fspecial('Laplacian'), 'replicate', 'conv'));
imshow(WL2,'Parent',handles.axes4);
title(handles.axes4,'Laplacian contrast weight');
h = 1/16* [1, 4, 6, 4, 1];
WC2 = imfilter(R2, h'*h, 'replicate', 'conv');
WC2(find(WC2 > (pi/2.75))) = pi/2.75;
WC2 = (R2 - WC2).^2;
WS2 = saliency_detection(img2);
imshow(WS2,[],'Parent',handles.axes5);
```

```
title(handles.axes5,'Saliency weight ');
sigma = 0.25;
aver = 0.5;
WE2 = exp(-(R2 - aver).^2 / (2*sigma^2));
imshow(WE2,[],'Parent',handles.axes6);
title(handles.axes6,'Exposedness weight ');
W1 = (WL1 + WC1 + WS1 + WE1) ./ ...
    (WL1 + WC1 + WS1 + WE1 + WL2SS + WC2 + WS2 + WE2);
W2 = (WL2 + WC2 + WS2 + WE2) ./ ...
    (WL1 + WC1 + WS1 + WE1 + WL2 + WC2 + WS2 + WE2);
level = 5;
Weight1 = gaussian_pyramid(W1, level);
Weight2 = gaussian_pyramid(W2, level);

R1 = laplacian_pyramid(double(double(img1(:, :, 1))), level);
G1 = laplacian_pyramid(double(double(img1(:, :, 2))), level);
B1 = laplacian_pyramid(double(double(img1(:, :, 3))), level);
R2 = laplacian_pyramid(double(double(img2(:, :, 1))), level);
G2 = laplacian_pyramid(double(double(img2(:, :, 2))), level);
B2 = laplacian_pyramid(double(double(img2(:, :, 3))), level);

for i = 1 : level
  R_r{i} = Weight1{i} .* R1{i} + Weight2{i} .* R2{i};
  R_g{i} = Weight1{i} .* G1{i} + Weight2{i} .* G2{i};
  R_b{i} = Weight1{i} .* B1{i} + Weight2{i} .* B2{i};
end

R = pyramid_reconstruct(R_r);
G = pyramid_reconstruct(R_g);
B = pyramid_reconstruct(R_b);
fusion = cat(3, uint8(R), uint8(G), uint8(B));
uiqm = UIQM(fusion)
figure, imshow([img, fusion])
title('Input and Enhanced Fusion')

% --- Outputs from this function are returned to the command line.
function varargout = RESULTS_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
```

B. SCREENSHOTS

7.1 White balancing and Gamma correction



(a): Input image

(b): White balanced image

(c): Gamma Corrected Image

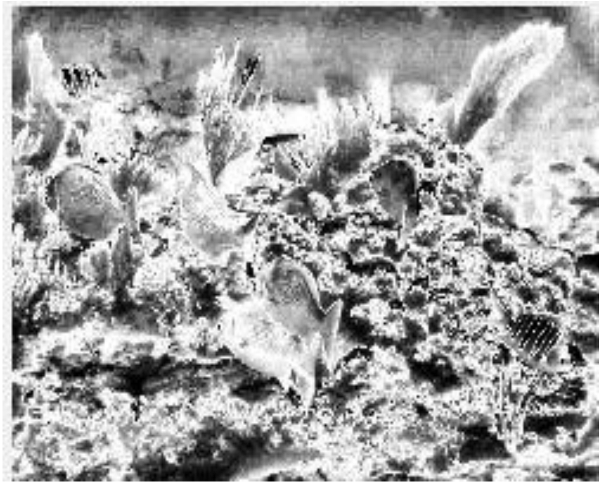
7.2 Multiscale Fusion

(a): Color Balanace

(b): Lab Image

(c): Saliency map

(d): Laplacian Contrast



(e): Saliency Weight **(f) : Exposedness Weight**

PROCESSED IMAGE

Input Image **Processed Image**

REFERENCES

[1] M. D. Kocak, F. R. Dalgleish, M. F. Caimi, and Y. Y. Schechner, "A focus on recent developments and trends in underwater imaging," *Marine Technol. Soc. J.*, vol. 42, no. 1, pp. 52–67, 2008.

[2] G. L. Foresti, "Visual inspection of sea bottom structures by an autonomous underwater vehicle," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 5, pp. 691–705, Oct. 2001.

[3] A. Ortiz, M. Simó, and G. Oliver, "A vision system for an underwater cable tracker," *Mach. Vis. Appl.*, vol. 13, pp. 129–140, Jul. 2002.

[4] A. Olmos and E. Trucco, "Detecting man-made objects in unconstrained subsea videos," in *Proc. BMVC*, Sep. 2002, pp. 1–10.

[5] B. A. Levedahl and L. Silverberg, "Control of underwater vehicles in full unsteady flow," *IEEE J. Ocean. Eng.*, vol. 34, no. 4, pp. 656–668, Oct. 2009.

[6] C. H. Mazel, "In situ measurement of reflectance and fluorescence spectra to support hyperspectral remote sensing and marine biology research," in *Proc. IEEE OCEANS*, Sep. 2006, pp. 1–4.

[7] Y. Kahanov and J. G. Royal, "Analysis of hull remains of the Dor D Vessel, Tantura Lagoon, Israel," *Int. J. Nautical Archeol.*, vol. 30, pp. 257–265, Oct. 2001.

[8] R. Schettini and S. Corchs, "Underwater image processing: state of the art of restoration and image enhancement methods," *EURASIP J. Adv. Signal Process.*, vol. 2010, Dec. 2010, Art. no. 746052.

[9] S. G. Narasimhan and S. K. Nayar, "Contrast restoration of weather degraded images," *IEEE Trans. Pattern Anal. Mach. Learn.*, vol. 25, no. 6, pp. 713–724, Jun. 2003.

[10] D.-M. He and G. G. L. Seet, "Divergent-beam LiDAR imaging in turbid water," *Opt. Lasers Eng.*, vol. 41, pp. 217–231, Jan. 2004.

[11] Y. Y. Schechner and Y. Averbuch, “Regularized image recovery in scattering media,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1655–1660, Sep. 2007.

[12] B. L. McGlamery, “A computer model for underwater camera systems,” *Proc. SPIE*, vol. 208, pp. 221–231, Oct. 1979.

[13] J. S. Jaffe, “Computer modeling and the design of optimal underwater imaging systems,” *IEEE J. Ocean. Eng.*, vol. 15, no. 2, pp. 101–111, Apr. 1990.

[14] H. Koschmieder, “Theorie der horizontalen sichtweite,” *Beitrage Phys. Freien Atmos.*, vol. 12, pp. 171–181, 1924.

[15] M. Levoy, B. Chen, V. Vaish, M. Horowitz, I. McDowall, and M. Bolas, “Synthetic aperture confocal imaging,” in *Proc. ACM SIGGRAPH*, Aug. 2004, pp. 825–834.

[16] S. K. Nayar and S. G. Narasimhan, “Vision in bad weather,” in *Proc. IEEE ICCV*, Sep. 1999, pp. 820–827.

[17] J. Kopf *et al.*, “Deep photo: Model-based photograph enhancement and viewing,” *ACM Trans. Graph.*, vol. 27, Dec. 2008, Art. no. 116.

[18] R. Fattal, “Single image dehazing,” in *Proc. ACM SIGGRAPH*, Aug. 2008, Art. no. 72.

[19] K. He, J. Sun, and X. Tang, “Single image haze removal using dark channel prior,” in *Proc. IEEE CVPR*, Jun. 2009, pp. 1956–1963.

[20] J.-P. Tarel and N. Hautiere, “Fast visibility restoration from a single color or gray level image,” in *Proc. IEEE ICCV*, Sep. 2009, pp. 2201–2208.