Ph.D. Thesis

RECONFIGURABLE PLATFORM BASED DESIGN IN FPGA FOR UNDERWATER IMAGE ENHANCEMENT, OBJECT DETECTION AND PIPELINE TRACKING

Submitted to

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

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

Doctor of Philosophy

by

S. M. ALEX RAJ

Under the guidance of

Prof. (Dr.) Supriya M. H.

DEPARTMENT OF ELECTRONICS
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
COCHIN – 682 022, INDIA

NOVEMBER 2017

RECONFIGURABLE PLATFORM BASED DESIGN IN FPGA FOR UNDERWATER IMAGE ENHANCEMENT, OBJECT DETECTION AND PIPELINE TRACKING

Ph. D. Thesis in the field of Ocean Electronics

Author

S. M. Alex Raj
Research Scholar
Department of Electronics
Cochin University of Science and Technology
Cochin – 682 022, India
e-mail: alexrajsm@gmail.com

Research Advisor

Dr. Supriya M. H.
Professor
Department of Electronics
Cochin University of Science and Technology
Cochin – 682 022, India
e-mail : supriyadoe@cusat.ac.in

November, 2017

Dedicated to…………

My Parents, Wife & Kids

CERTIFICATE

This is to certify that this thesis entitled, ***Reconfigurable Platform Based Design in FPGA for Underwater Image Enhancement, Object Detection and Pipeline Tracking*** is a bonafide record of the research work carried out by Mr. S. M. Alex Raj under my supervision in the Department of Electronics, Cochin University of Science and Technology. The results presented in this thesis or parts of it have not been presented for any other degree(s).

Cochin - 682022
16th November 2017

Prof. (Dr.) Supriya M. H.
Supervising Guide

DECLARATION

I hereby declare that the work presented in this thesis entitled ***Reconfigurable Platform Based Design in FPGA for Underwater Image Enhancement, Object Detection and Pipeline Tracking*** is a bonafide record of the research work carried out by me under the supervision of Dr. Supriya M. H., Professor, in the Department of Electronics, Cochin University of Science and Technology. The results presented in this thesis or parts of it have not been presented for other degree(s).

Cochin – 22 **S. M. ALEX RAJ**
16th November 2017

COCHIN UNIVERSITY OF
SCIENCE AND TECHNOLOGY

CERTIFICATE

This is to certify that this thesis entitled, ***Reconfigurable Platform Based Design in FPGA for Underwater Image Enhancement, Object Detection and Pipeline Tracking*** has been modified to effect all the relevant corrections suggested by the Doctoral Committee and the audience during the Pre-synopsis Seminar.

Cochin - 682022
16th November 2017

Prof. (Dr.) Supriya M. H.
Supervising Guide

Acknowledgements

I would like to express my deepest sense of gratitude to my research guide, **Prof. (Dr.) Supriya M. H.**, Professor, Department of Electronics, Cochin University of Science and Technology for her excellent guidance and incessant encouragement. It has been a great pleasure and privilege to work with her and she was always there when I needed help.

Sincere thanks are due to **Prof. (Dr.) K. Vasudevan, Prof. (Dr.) P.Mohanan, Dr. C. K. Aanandan and Dr. James Kurian** , Department of Electronics, Cochin University of Science and Technology, for providing adequate help and fruitful suggestions.

I would like to give a special word of thanks to the research fellows in the ***Advanced Centre for Signal Processing, Instrumentation and Research*** **(ASPIRE)**, Mr. Mohan Kumar K., Mr. Suraj Kamal, Mr. Satheesh Chandran, Mr. Shameer K.Mohammed, Mrs. Rithu James, Mrs. Sherin and Mrs. Sabna.

A word of mention is deserved by Mr Manoj, Mrs. Aruna, Mrs. Khadeeja, Mr. Abhilash, Ms. Rita, Ms. Claris and Mr. Nandan who all were there for me as a good support throughout my research period.

I thankfully bear in mind the sincere co-operation and support I received from the **library and administrative staff** of the Department.

It is beyond words to express my gratitude to my family members especially to my parents **Mr. M. N. Sathiaraj** and Mrs**. S. T**. **Margaret Mary,** my wife **Mrs. Rithu C. S.** and my kids **Febin Alex** and **Jibin Alex** for their sacrifice in connection with the preparation of my thesis. I am sure I could not have completed this great task without their support and cooperation.

S. M. ALEX RAJ
16th November 2017.

Table of Contents

List of Figures

List of Tables

Abbreviations

AIHM	-	Adaptive and Iterative Histogram Matching
ASIC	-	Application Specific Integrated Chip
AUV	-	Autonomous Underwater Vehicle
CCD	-	Charge Coupled Device
CLAHE	-	Contrast Limited Adaptive Histogram Equalisation
DCT	-	Discrete Cosine Transform
DSP	-	Digital Signal Processor
EM	-	Expectation Maximisation
FPGA	-	Field Programmable Gate Array
GMM	-	Gaussian Mixture Model
GPR	-	Ground Penetration Radar
GPU	-	Graphical Processing Unit
GW	-	Grey world
HA	-	Hungarian Assignment
HBF	-	High Boost Filtering
HDL	-	Hardware Description Language
HSV	-	Hue Saturation Value
IMECO	-	Image Enhancement Co-Processor
IR	-	Infra Red
JND	-	Just Noticeable Difference
KDE	-	Kernel Density Function
LoG	-	Laplacian of Gaussian
MAE	-	Mean Absolute Error
MRF	-	Markov Random Field
MSC	-	Multi-State Convolution
MSR	-	Mean Square Error
MSRCR	-	Multi-Scale Retinex with Colour Restoration
PSNR	-	Peak Signal to Noise Ratio
RMSE	-	Root Mean Square Error
ROV	-	Remotely Operated Vehicle
RTL	-	Register Transfer Logic
SDPS	-	Symmetric Dynamic Programming Stereo
SMQT	-	Successive Mean Quantization Transform
UART	-	Universal Asynchronous Receiver Transmitter
UUV	-	Unmanned Underwater Vehicle

CHAPTER 1

INTRODUCTION

Water is the real elixir of life. Ocean covers approximately 70 percentage of the earth's surface. Despite the importance of oceanic environment, humans are still unable to inquire the full depth of the ocean and discover its resources and wealth due to the dangerous, cold and dark, unfamiliar environment. The safety and surveillance of oceanic environment thus become very relevant for research.

Underwater oil and gas resources account for around six percentage of global oil production. Once oil and gas are discovered in an underwater field, massive production platforms and specially designed systems and pipelines are required to extract and transport the oil and gas to shore [1] which makes it very mandatory to keep track of their physical condition. Recently, remotely operated vehicles (ROV) and autonomous underwater vehicles (AUV) are on the threshold of playing a key role in inspecting pipelines because they eliminate the need for humans to be at great depth or in dangerous conditions. It can collect and save required information without human intervention making it possible to introduce complete level of autonomy in inspecting pipelines. Vision based system is one of the cheapest methods for underwater pipeline inspection [2].

This chapter deals with the area of research work undertaken and the significance of the work. It explains different methods used for autonomous vision based underwater navigation. Various underwater image enhancement techniques and object detection techniques that are used prior to underwater pipeline tracking are also explained. Field programmable gate array (FPGA), which is used for hardware implementation, is also introduced in this chapter.

1.1 Underwater Vehicle Navigation

The typical techniques adopted by marine biologists and marine guards, to study and analyse underwater objects and animals involve underwater observation and photography, combined net casting and acoustic (sonar) and human hand-held video filming. Even though the net casting method is accurate, it damages the environment. Periodic observations are conducted by marine biologists which can also help in other marine related applications. This time consuming work heavily depends on strict knowledge and the ability to observe, count and describe. Limited information is provided by human filming and habitat, though they do not damage the habitat. The level of accuracy and interpretation of the collected information strictly depends on the knowledge and database available to the observer. Underwater vehicle navigation is an alternative method to solve this issue.

The precise navigation of underwater vehicles is a difficult task due to the challenges imposed by the variable oceanic environment. It is particularly difficult if the underwater vehicle is trying to navigate under the Arctic ice shelf. Indeed, in this scenario, traditional navigation devices such as GPS, compasses and gyrocompasses are unavailable or unreliable. Also, the shape and thickness of the ice shelf is variable throughout the year. Current Arctic underwater navigation systems include sonar arrays to detect the proximity to the ice. However, these systems are undesirable in a wartime environment, as the sound gives away the position of the underwater vehicle[3].

1.2 Traditional Methods for Underwater Vehicle Navigation

The different methods such as acoustic and vision based techniques can be used to explore the underwater environment.

1.2.1 Acoustic Methods

One of the major techniques used to analyse object distribution is Sonar Technology. Acoustic echo-sounding is a popular method to determine the presence of objects in the underwater environment. Typically, the acoustic energy back scattered from the object is analysed by a receiver located at the same site as the transmitter. When sound travels through a moving heterogeneous medium such as river, it is refracted in various ways due to the presence of significant and insignificant objects in the medium. Observation of these distorted signals lacks crucial information regarding the sources causing the refraction. So this method cannot be considered as the optimum technique to detect the underwater objects. Under such complicated conditions video based object detection can provide better results compared to other methods available. Fig. 1.1 shows acoustic method of underwater vehicle navigation. Here, a receiver located at the same site as the transmitter analyses the acoustic energy back scattered from the object.

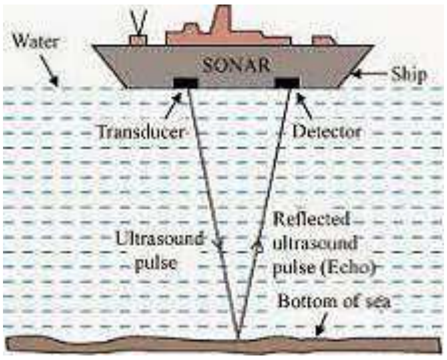


Fig. 1.1 Acoustic method

1.2.2 Vision based Methods

Detection of underwater objects can be carried out using vision based systems. Many relevant information can be revealed by videos and images which can be utilised for mankind. The unique set of constraints existing in underwater, limits the ability to process the underwater images. The physics of light and attenuation of electromagnetic spectrum which are associated with underwater images are the major constraints. Absorption and scattering, the specific features of light, make it difficult to process vision task. Underwater images degrade due to non-uniform illumination, low contrast, and colour change with prominent green or blue hues, low contrast and short range. The task of extracting objects by observation can be extremely difficult as the vision is restricted due to illumination changes, movements of objects, different morphological properties, complex underwater environment, degraded image quality, partial and full occlusions, requirements for real time processing etc.

1.3 Underwater Imaging

Underwater images get degraded due to poor lighting conditions and natural effects like bending of light, denser medium, reflection of light and scattering of light etc. As the density of sea water is 800 times denser than air, when light enters from air (here lighting source) to water, it partly enters the water and partly reflected reverse. More than this, as light goes deeper in the sea, the amount of light enters the water also starts reducing. Due to absorption of light in the water molecules, underwater images will be darker and darker as the deepness increases. Depending on the wavelength also, there will be colour reduction. Red colour attenuates first followed by orange. As the blue colour is having shortest wavelength, it travels longest in seawater there by dominating blue colour to the underwater images, affecting the original colour of the image.

1.3.1 Optical Propagation of Light in the Sea

The overall performance of underwater vision system is influenced by the basic physics of propagation of light in the sea. The overall performance of underwater vision system is influenced by the basic physics of propagation of light in the sea. One must take care about absorption and scattering of light when considering underwater optical imaging systems. Most often accurate values of attenuation and thorough knowledge of forward and backward scatter is required for restoring underwater images.

1.3.2 Underwater Optical Systems

Underwater optical systems can also be classified as passive and active as in the case of sonar systems. In the case of passive systems, the image objects have been illuminated by some other source than the one associated with the imaging system. Here sunlight or some other source like bioluminescence is the main source of light. This type of systems are particularly useful for fish seeking prey or for those who are visualising the scene without any detection On the other hand, active systems make use of their own generated source of light. Here one can use strobe or continuous artificial illumination. The incidental light can be collimated into very narrow beams, having an option of short duration.

For short range operations, simple systems incorporating a good camera with a controlled light are sufficient. For long ranges, separate illumination systems have to be added along with camera.

1.4 Underwater Image Enhancement

Due to apprehension regarding the present conditions of the world's oceans many large scale scientific projects have instigated to examine this. The underwater video sequences are used to monitor marine species. Underwater image is inundated by reduced visibility conditions making images deprived of colour variation and contrast. Due to this restriction other methods like sonar ranging has been preferred previously. Since alternate methods yield poor resolution images which are hard to understand, nowadays for close range studies visible imaging are preferred by scientists. The emphasis of this work deceits primarily in the area of implementation of vision system which involves analysis of enhanced images.

Image enhancement techniques are usually divided in frequency domain and spatial domain methods. The frequency domain methods are normally based on operations in the frequency transformed image while spatial domain methods are based on direct manipulation of the pixels in the image itself. Some fundamental image enhancement methods are introduced in the following sections.

1.4.1 Frequency Domain Techniques

Frequency domain techniques are Fourier transform based filtering, homomorphic filtering etc. Here Fourier transform of the degraded image is taken and is multiplied by an appropriate filter. Few basic filters are explained here.

- **Low Pass Filter**:

  Low pass filtering involves the elimination of the high frequency components from the image resulting in the sharp transitions reduction that are associated with noise. An ideal low pass filter would retain all low frequency components and eliminate all high frequency components but low pass filters suffer from two problems: blurring and ringing.

- **High Pass Filter**:

These filters are basically used to make the image appear sharper. High pass filtering works in exactly the same way as low pass filters but uses the different convolution kernel and it emphasises on the fine details of the image. While high pass filter can improve the image by sharpening overdoing of this filter can actually degrade the image quality.

1.4.2 Spatial Domain Techniques

Spatial domain techniques based image enhancement are contrast enhancement, grey level slicing, histogram equalisation, point processing operation, negative of an image, power law transformation etc. A few methods are described below.

- **Contrast Enhancement**

Image contrast enhancement is a method which is used to increase the visibility of images. Here, the intensity of the pixel grey level is modified based on a function. Intensity based methods are of the form:

$$I_o(x,y) = f\big(I(x,y)\big) \tag{1.1}$$

In Eqn. 1.1, the original image is $I(x,y)$, the output image is $I_o(x,y)$ and f is the transformation function. Intensity based methods transmute the grey levels over the entire image. Even after the transformation pixels with same grey levels throughout the image remain same. Contrast stretching is a generally used method that falls into this group.

- **Histogram Equalisation**

The histogram is a graph showing the number of pixels in an image at each different intensity value found in that image. Histogram equalisation redistributes pixel intensity values in an attempt to flatten (evenly distribute) the image histogram, thus increasing the dynamic range and as a result, increasing the image contrast. The method is useful in images with both backgrounds and foregrounds are bright or dark. It often produces unrealistic effects in photographs, but is very useful in scientific images such as x-ray, satellite or thermal images. Histogram equalisation [4] differs from contrast stretching in that it uses non-linear transfer functions to map between pixel intensity values in the input and output images.

1.5 Underwater Object Detection

Underwater object detection is necessary for obstacle avoidance in front of the vehicle and also for detecting pipeline. Due to the complex and dynamic environment inside the water, fast moving objects cannot be tracked accurately.

The situation becomes more complex when moving object tracking is considered in a dynamic water surface. In particular, blurriness is the most common situation in an underwater video taken in dynamic water, and its removal is difficult. In this environment, movement of the object at different velocities and at different time are the most serious problems. The detection methods like temporal difference, background subtraction or optical flow can detect moving objects.

1.5.1 Temporal Difference Method

Temporal difference computes the difference between two or three consecutive frames. It is good at adapting to the dynamic environments, but poor at extracting enough relevant feature pixels, resulting in holes being generated in the moving object. This method is based on simple convolution. Hence the method is fast and simple to implement. But besides all these advantages, this method is susceptible to noise and variations of the timings of movement of objects.

1.5.2 Background Subtraction method

In this method, a reference image is calculated, and each new frame is subtracted from the reference image followed by segmentation and thresholding. Time-averaged background image is used as a reference image. The main drawback of this procedure is the need for calculation of reference image in such a way that foreground objects are absent. This method is not used when varying illumination or moving backgrounds are considered. The solution to this problem is the re-estimation of the reference image. This method is very tedious due to variations in illumination, swaying vegetation, rippling water and effects of shadows. New models like texture based background removal can be developed to solve this.

Single Gaussian background model, Gaussian Mixture Modelling (GMM) and Kernel Density Estimation (KDE) are the most commonly used methods for background subtraction [5].

1.5.3 Optical Flow Method

Optical flow estimation yields a two-dimensional vector field, i.e., motion field that represents velocities and directions of each point of an image sequence. It presents moving object detection by deformation of frames. The two-dimensional vector that is yielded from optical flow method gives all information about moving object but is complex when implemented for real-time applications [6].

1.6 Underwater Pipeline Tracking

In the last few years, exploitation of underwater gas and oil fields have increased. The produced oil and gas is transported mainly using pipelines which need to be regularly inspected. Submarine communication cables are laid on the sea floor for data communication across stretches of ocean. Due to earthquakes and other environmental changes these underwater cables and pipelines can get wear and tear which need to be found and repaired quickly. The constant need for monitoring the state of the pipelines and cables is highly important.

While inshore pipelines can be maintained by divers, for offshore pipelines, unmanned underwater vehicles (UUV) are preferred [7]. Inspection of underwater pipelines is done using Remotely Operated Vehicles (ROVs) which require human intervention. To avoid this risky method a more practical solution will be, to develop an intelligent vision-based navigation and guidance system which involves an efficient method for vision-based target detection and tracking methods.

1.7 Field Programmable Gate Array (FPGA)

FPGAs are often used as implementation platforms for real-time image processing applications because their structure can exploit spatial and temporal parallelism [8]. In image processing, FPGAs have shown very high performance in spite of their low operational frequency [9].

The main hardware platform to realise high-performance parallel architecture is Application Specific Integrated Chip (ASIC). However, with the advances in semiconductor technology the capacity and performance of FPGAs have improved to such an extent that, together with their inherent parallelism and reconfigurability, these devices have become a viable prototyping hardware platform for the investigation and implementation of high performance processing algorithm. Subsequently, FPGAs are now used, extensively, in modern high-performance filtering applications such as medical imaging, mobile video applications, satellite data weather forecasting etc. Thus, enormous parallel algorithms applications are implemented in FPGA utilizing it as a reconfigurable hardware platform.

However, as the complexity of FPGAs has increased several inadequacies in the software support tools, associated with FPGA realisations are becoming apparent and needs to be addressed. First, there is a lack of suitable constructs in Hardware Design Language (HDL), used in early design phase, to efficiently describe the structures to implement the parallelism existing in multi-dimensional convolution/correlation algorithms and their parallel filtering architectures, being realised in FPGA technology. Second, to facilitate the rapid prototyping of architectures, an IP library of commonly used cores needs to be developed, which would also include an efficient mapping strategy for the fundamental blocks onto FPGA structure. Finally, as the complexity of the systems being implemented on an FPGA device increased, there is a requirement for a concurrent hardware/software system design flow [10][11].

1.8 Motivation for the work

Undersea cables and pipelines are partly exposed to open water where salt, plants, moving sand, fishing activities, anchors and even shark bites can damage the conduits so that they need to be monitored from time to time. Inshore divers can control the conduits, while offshore, with increasing depth, underwater vehicles have to fulfil this task. When cables and pipelines are to be controlled with a remotely operated vehicle, the operator has to steer the vehicle following the conduit by watching the camera images that are sent by the vehicle to the ship. At the same time, the operator or another person has to search for defects. Such a manual visual control is a very tedious job and tends to fail if the operator loses concentration. Therefore and because of the high cost of ROVs, autonomous underwater vehicles (AUVs) are used to fulfil this task without any user interaction. Recently installed cables and pipelines have defined shape and colour and can therefore, be easily detected with cameras, but, when they get older, their visual appearance changes drastically. Marine flora tends to grow on top and in the neighbourhood of cables and pipelines and moving mud and sand can make them hardly visible. In the deep sea, robots have to carry their light sources because of the lack of ambient light, which results in irregular lighting conditions. Additionally, light suffers from absorption and dispersion along its propagation in the oceanic medium that gives rise to blurred and low-contrast images.

As per literature, detection of clear shaped cables and pipelines with straight, high-contrast borders in camera images can be considered as solved. Simple border enhancing filters followed by a filter can be used for tracking. This method does not work on noisy images of old cables.

The design can be implemented in ASIC, FPGA or DSP. Designing an ASIC for a limited number of applications is not cost effective. Implementing the design in a DSP is just running a software in a special processor. DSP based design is not a true hardware design. On the other hand, FPGA design is a hardware implementation where the design configure a Register Transfer Level (RTL) hardware depending upon the algorithm. So, for a limited number of applications, FPGA can be an alternative to ASIC.

1.9 Objective of the work

The present work aims at FPGA implementation of a new visual cable/pipeline tracking algorithm, comprising of image enhancement and object detection, which can be used to control AUV. Thus the vision system should be able to navigate along a cable or a pipeline respectively without any user interaction. The following objectives have been formulated to achieve this.

- Developing a novel pre-processing algorithm to enhance underwater degraded images.
- Developing a better object detection algorithm to avoid obstacle avoidance in front of the AUV.
- Developing an improved algorithm for tracking underwater pipeline.
- Implementing the developed algorithms in FPGA.

1.10 Research Contribution

Multi-scale image fusion from a single image was used for enhancing degraded underwater images. New derived input from the single input image and new weight maps, suitable for the underwater scenario, were derived and were fused to form the enhanced image. Underwater images suffer from colour degradation due to light scattering and attenuation of light waves through water. Colour correction was done after image fusion. Novel algorithms were also developed based on multi-scale retinex algorithm and Adaptive Histogram Equalisation (AHE) to enhance underwater images. The algorithms were compared on the basis of qualitative metrics.

Next, object detection was carried out using a novel algorithm based on adaptive Gaussian mixture model. Here modified median filter and

Otsu's thresholding were used to obtain better results in an underwater environment. Comparison with existing AGMM and conventional background subtraction was made.

After detection, pipeline was tracked using alpha beta filtering based method. Alpha beta filtering was chosen for its simple computation. The model was modified with assumptions and computations to incorporate speed variations of the vehicles that showed better results for pipeline tracking. The trajectory of the proposed algorithm was compared with the conventional alpha beta filtering method for various trajectories such as sine, step and random shapes.

All designs were implemented in Virtex 4 FPGA. Their device utilisation were tabulated. The work was carried out using offline underwater images/videos.

1.11 Organization of the Thesis

The salient highlights of the work carried out are explained below.

1.11.1 Review of the past work

Chapter 2 addresses review of research works reported in literature in the areas of underwater image enhancement, object detection, underwater cable/Pipeline tracking and Field Programmable Gate Array implementations. Review of various image enhancement techniques was carried out. Various algorithms used for cable/pipeline tracking are reviewed. Various FPGA implementations of image processing algorithms were discussed. The advantages of FPGAs over digital signal processors for implementing parallel algorithms were also noted.

1.11.2 Methodology

Chapter 3 addresses the methods adopted in the proposed work carried out. The proposed model of the work is briefly explained in this section. Various methods that were used for implementing the design is also described.

1.11.3 Multi-scale pixel based image fusion

Chapter 4 mainly deals with the design and implementation of multi-scale fusion method used for enhancing underwater images. Other image enhancement techniques that are used to compare with the fusion method is also narrated. Their implementations in FPGA and the results are also explained.

1.11.4 Underwater moving object detection

Moving objects can be hazardous while tracking pipeline. Detecting moving objects can be helpful for the smooth navigation. Chapter 5 deals with implementation of moving object detection using background subtraction method and Adaptive Gaussian Mixture Model (AGMM) method.

1.11.5 Pipeline Tracking

Pipeline tracking based on Hough transform and its implementation in FPGA are described in chapter 6. Pipeline tracking using optical imaging is one of the recent research areas in the underwater scenario.

1.11.6 Conclusions

Chapter 7 addresses conclusions and future scope of the thesis. The various algorithms implemented in FPGA for underwater pipeline tracking is detailed. The scope of the future work is also described.

1.12 Summary

A brief introduction to underwater imaging and basic image enhancement techniques is explained in this chapter. Moving object detection and cable tracking methods are also highlighted. The importance of FPGAs in image processing applications is also discussed in this chapter. Motivation, objectives and research contribution of the thesis is narrated. The salient highlights of work carried out are also briefly introduced.

CHAPTER 2

LITERATURE REVIEW

This chapter is devoted to the review of research work reported in open literature in the areas of underwater image enhancement, pipeline tracking and FPGA based design. The various methods adopted for underwater image enhancement and its requirement for various applications are reviewed. Different techniques described in the literature for object detection and tracking are also explained. The importance of FPGA in image processing and its advantages over DSP's and general purpose processors are also reviewed. The consolidated findings and results of various researchers for different methods and techniques are discussed.

2.1 Introduction

Visual tracking is one of the options used in AUV for navigation. To study the particular behaviour of a species or behavioural pattern of underwater life, close observation for a long period is required. While observing the target, it may, sometimes, necessary to move along with the object. In situations like maintaining man made under water objects by routine observations, AUV must be capable of doing close observations. Therefore, the underwater exploration and maintenance require that the AUV have proper observations so that close tracking and following along or with these objects is necessary.

FPGAs are often used as implementation platforms for real-time processing applications because their structure is able to exploit spatial and temporal parallelism. Such parallelisation is subject to the processing mode and hardware constraints of the system. Considering its low power consumption, FPGA can be well suited for implementing vision algorithms to be employed with AUV.

The literature review was carried out in three parts. In the first part, review of different image enhancement techniques used for various applications is done. The importance of underwater image enhancement is also reviewed. In the second part, algorithms used for object detection and tracking are reviewed. The advantages and disadvantages of various algorithms are also noted. The importance of FPGA in implementing image processing algorithms in comparison with other hardware implementing platforms such as Digital Signal Processors, ASIC, that are explained in open literature, are included in the third part.

2.1.1 Underwater Image Enhancement.

The autonomous underwater vehicle incorporates vision system for various applications. A review of several applications is carried out. All these applications require pre-processing of an acquired video or image. Various techniques for enhancing underwater images are also reviewed.

Donna et al. [12] suggested that to have a better advancement in underwater imaging the following areas have to be advanced. Affordable, high quality cameras needed, compact, efficient and easy to program digital signal processors should be available and better modelling and simulation software is required. The authors have set up camera system to monitor coral reef communities in marine parks. Not all cameras deployed in the sea are stationary or mounted to underwater vehicle. Since mid-1980's, animals have become imaging platforms.

Padmavathy et al. [13] described the need of pre-processing for underwater images because of the poor quality of the captured image. Various filters such as homomorphic, wavelet denoising etc. are compared. For Gaussian noise, wavelet method was better, and for salt and pepper noise homomorphic filter has shown a better result. Speckle noise can easily be removed by using wavelet denoising.

In their work, Prabhkar et al. [14], have taken images using Canon D10, a waterproof camera, at an approximate depth of 2 m, measured from the surface level of water. The pre-processing method consists of homomorphic filtering which helps in correction of non-uniform illumination of light, and another wavelet denoising technique was used to remove additional Gaussian noise present in underwater images. In the end, a bilateral filtering was applied to smoothen underwater images and contrast stretching was used to normalise the RGB values.

Plakas et al. [15] stated in their paper that the use of uncalibrated computer vision techniques was used in the field of underwater application of computer vision where thermal and mechanical strain on the camera along with a usually unknown and highly unstructured environment, poorly illuminated source and noisy images were present. Uncalibrated means, intrinsic and extrinsic parameters affecting images were unknown. Calibration could be lost due to collisions, changes in the environment or other unpredictable events. The authors used two modules for reconstruction. One, computing projective reconstruction from a set of feature correspondences between images in a video stream. The second one, recovering the Euclidian target reconstruction from the projective one, employing a priori knowledge of the distances between five identifiable seen points.

John et al. [16] stated that none of the existing approaches could handle the scattered light and colour image distortions which took place in underwater images with the possible presence of artificial lighting. The authors claimed that their algorithm tried to minimise the effect of the probable presence of the artificial source. Dark channel prior (method to derive scene depth) was used first to evaluate the scene distance from the camera. After the derivation of the depth of map, the foreground and background areas inside the image were divided. The light intensities of background and foreground were then compared to find the usage of the artificial light source during image capturing process. The hazing algorithm and wavelength compensation eliminated haze effect and colour image variation along the underwater propagation.

The approach of Fan et al. [17] applied multi-scale retinex algorithm to the luminance component of the image. Then they did subtraction operation followed by median filter.

An approach was employed by Kaiming et al. [18] using the dark channel prior. Using this prior, the thickness of the haze can be directly evaluated and a good quality image free form haze can be obtained. The drawback of above discussed algorithms was a slower response i.e., processing time of 35 seconds, on a 600*400 image.

Jin Hwan et al. [19], in their paper discussed an algorithm for the estimation of the air light in a haze distorted image, based on the quad tree subdivision. This approach did estimation of the transmission map to maximise the contrast of the output image. To measure the contrast, they used a cost function, which contains standard deviation term and a histogram uniformness term. Obtained results showed that the above discussed method was able to eliminate haze from the image efficiently.

Oakley et al. [20] under the assumption that all pixels in the whole image have similar depth values, subtracted the same offset from all pixel values. Their algorithm however failed to remove haze when a captured image had variable scene depths. It was implemented using MATLAB language, and a host pc with a 2.5GHz Core Quad processor. The algorithm took around 20 to 30 seconds to dehaze a 600 × 400 image.

Raanan et al. [21], proposed a new approach for estimation of the optical transmission in hazy scenes given a single input image. In this scattered light was removed to increase scene visibility and for obtaining haze free scene contrasts of an image. They formulated a model for refined image formation that would held responsible for surface shading along with transmission function as well. From the degraded version of the image, the inputs and weight measures were derived.

Based on fusion principles, two derived inputs that represented colour corrected and contrast enhanced versions of the original image were taken by Ancuti et al. [22]. The authors obtained three weight maps which aimed to improve the visibility of the distant objects, which were degraded due to scattering and absorption. This technique did not need any special hardware or knowledge about the scene structure. The authors claimed a better image quality over polarization method.

Nicholas et al. [23] discussed a technique which removed the effects of light scattering, termed as dehazing, from underwater images. The basic idea here was to exploit the large variation in attenuation among the three image colour channels in water to determine the scene depth. Then this depth was used to reduce the spatially changing effect of haze. The output was shown with underwater images.

Kratz and Nishino [24] used a factorial Markov Random Field (MRF) to clear haze from an image. This provided a common framework, which included natural image statistics and different priors on depth.

A novel dark channel statistical prior, which was at the minimum intensity colour channel in an image patch, was introduced by He et al. [25]. It utilised the fact that object inside a clean image patch will have minimum a single colour channel with very low intensity, whereas all colour channels in a hazy patch will have larger intensities due to the presence of air light. For the elimination of haze from a hazy image, they have utilised statistical priors on the properties of natural images.

Tan et al. [26] recovered the haze less image through maximisation of the contras over local patches. The acoustic and optical sensors to be used for archaeological purpose were explained.

In this paper, Singh et al. [27] presented the importance of acoustic sensors to be used for distinguishing objects as large as shipwrecks or as small as an amphora. Optical sensing, including film, analogue video and digital imagery is a common choice for high-resolution imaging of underwater, but the rapid attenuation of electromagnetic radiation in underwater limited their use. Qualitative detection of a change in underwater at a very fine scale was possible, once acoustic and optical sensors were fused. No single sensor was sufficient, but a combination of several sensors can function a very powerful mechanism for addressing the needs of the archaeologist working in underwater.

Paik et al. [28] discussed several techniques and used sensors for mine detection. Most of the methods consist of signal processing, sensor and decision processor. For the sensor part Ground Penetration Radar (GPR), infrared (IR) and ultrasound sensors were mainly used. For the signal processing and decision parts, a set of image processing techniques which includes, enhancement, feature extraction, filtering were used.

According to Alian et al. [29], military AUV's would be able to execute survey missions in both known (maritime approaches, harbour areas, access channels etc.) and unknown environments to detect a potential threat as mine field. The implemented robots significantly improved exploration, analysis and provided a large decisional autonomy. Since obstacle must be seen at a long distance in order for the vehicle to have sufficient time to change its route, the use of optical imaging in AUV's for obstacle avoidance was limited.

Alan et al. [30] explained that detection and identification of objects in harbour underwater is a tough and annoying task as the identification required visual means in water. The authors were successful in making second generation LUCIE, underwater range gated imaging system which were placed on a ROV. The output video was digitised with the help of a frame grabber at a full resolution of 640x480 pixels and at a rate of 30 frames per second. This application allowed substantial amount of real time processing to the camera.

The authors, Xu Wen et al. [31] have made a sonar image processing system which was installed in the Chinese “Explorer” AUV. Based on a TMS320C30 high speed DSP, the system was used to realise the compression of sonar image and detection and identification of underwater objects which included object identification in real time. Pool and sea testing have performed better. For comparison, the authors used DCT (Discrete Cosine Transform). For object detection, a segmentation algorithm based on grey level thresholding, region growing, and dynamic searching was used in their system.

Xiaohai et al. [32] implemented the system on DSP board connected with a PC. After an image was processed by DSP, CPU started to identify and positioned the object. In the meantime, DSP restarted fetching of the next image. Thus, the authors have performed object detection. After the detection of object, the low-level command is generated which controlled the AUV to observe the complete object.

Toshihiro et al. [33] used a low pass filter to eliminate noise and back propagation algorithm to train and recognize patterns. They suggested a navigation method based on AUV for photo mosaicking of shallow vent regions where bubbles were present. The method utilised SLAM approach and estimated independent, drift free, accurate navigation using bubble plumes and manmade sonar reflectors. The positioning precision was approximated to be around 0. 2 m for horizontal positioning and 0.5 degree for heading.

Balasuriya et al. [34] suggested that an acoustic sensor along with a single camera was needed to determine the relative location of underwater cable with respect to AUV. Visual information helped for the determining two dimensions and the acoustic information determined the third dimension. Image filtering was carried out using Laplacian of Gaussian (LoG) operator. Position was determined using Hough transform method. The proposed model was experimented in an AUV at Lake BIWA. Hardware consists of T800 transputer based INMOS B429 image processing board. Speed of the processing was 10 frames per second.

A vision based system was designed by Rock et al. [35] which can detect a single ray of light generated from a laser. The hardware required two cameras, the first one was used for locating the target. It scanned the image from the previous known location of the target and if the target was unknown in view, then from the centre of the screen. The pixels were analysed row by row, progressing outward towards the edge. If a target was detected, its angle and elevation with respect to the centre of the image is computed and were sent to the vision processor. Its limitation was that it only worked for a single distinguishable target.

A vision based guidance for an AUV named Kambara which required double cameras was proposed by Gaskett et al. [36]. As per the proposed model, the guidance can be performed by a feature tracking algorithm which needed two correlation operations within the feature tracker. The tracker followed every single feature between previous image and current image via a single camera while the feature range estimator correlated between the right and left camera images. The feature motion tracker correlated stored feature templates to evaluate the image location and thus direction to each feature. Determination of range takes place by correlating the features in both images to detect pixel disparity in them. This pixel disparity depended on range which were evaluated by calibration. The range and direction to every feature was then transmitted to the controller which computed a set of thruster commands, which were used to guide AUV and to act as a function of the position of visual features.

A guidance law based on vision using only one camera had been implemented in a test bed underwater robot, Twin Burger 2, to track cable and moving object and was proposed by Balasuriya et al. [37]. The feature to be tracked produced a particular geometric feature in the images taken through CCD camera. The vision processor then labelled these features, extracting their location in the image and interpreting the visuals into a guidance parameter.

T.L. Tan, K.S. Sim and C.P. Tso [38] discussed the technique based on histogram equalisation which was used to improve the image contrast under the limitation of over enhancing brightness in the image background. Thus brightness preserving Bi-Histogram Equalisation (BHE) was described for preserving the image brightness by splitting the image based on the input mean. The sub images were then individually and independently, equalised and combined into the output image.

An Adaptive and Iterative Histogram Matching (AIHM) method for chromosome contrast enhancement particularly for the banding patterns had been proposed by Seyed Pooya et al. [39]. The reference histogram was matched with the initial image which was created based on some processes of the initial image histogram.

In this paper, Min Liu and Peizhong Liu [40] discussed an image enhancement technique for analysing videos. CI value was used as an evaluation function, which was used as a degree of enhancement. The technique utilised point analysis method of multi-dimensional biomimetic informatics.

R.K. Jha et al. [41] proposed a stochastic resonance based method for enhancing very low contrast images. An equation for threshold has been derived. Gaussian noise for enhancing standard deviation has been added iteratively to the images having low contrast value until the quality of image was maximised.

Praveen Sankaran et al. [42] discussed a technique based on Multi-Scale Retinex with Colour Restoration (MSRCR) that would provide superior output for enhancement. Because of less dynamic range of a camera as against human visual system, captured images were dependent on illuminating conditions.

A method for enhancement of the colour images using a nonlinear transfer function and preserving details of neighbourhood pixel was proposed by Deepak Ghimire and Joonwhoan Lee [43]. Here, the image enhancement was performed for V component of HSV colour image. S and H components were kept unvaried to avoid the disturbance in colour balance among HSV components. Enhancing of V channel was done in two steps. Firstly, the V component of HSV image was decomposed into small overlapping blocks and for every pixel inside the block, the luminance enhancement was applied through a nonlinear transfer function. There by second step included further enhancement of each pixel, to adjust the image contrast based upon the value of centre pixel and its neighbourhood pixel values. In the final step, original H and S component image along with the enhanced V component were restored in RGB image.

A multi-scale enhancement algorithm utilising Logarithmic Image Processing (LIP) model was proposed by Hong ZHANG et al. [44]. Features of the Human Visual System (HVS) were taken into consideration for a new measure of enhancement. A Just Noticeable Difference (JND) model of human visual system was proposed and it was used as a tool for evaluation of the performance of the enhancement technique.

Proposal on content aware algorithm was given by Adin Ramirez Rivera et al. [45]. The algorithm was supposed to enhance dark image, gave information about textured regions, sharpen edges and maintain the smoothness of flat regions. To produce maximum enhancement, this method introduced an ad hoc transformation for each image, adapting the mapping functions to every image feature. The author analysed the contrast of the image along the boundary and textured regions, and grouped the data with common features. These common groups realised the relations inside the image, from where the transformation functions were evaluated. Thus the results were adaptively mixed, by taking into consideration of the human vision system characteristics.

A real time model for detection of fog, based on a low cost black and white camera, for a driving application was proposed by Bronte et al. [46]. This system required two evaluations: first, determination of the visible distance, which was computed through the camera projection equations and the blurring due to the fog. Second is the darkest zones in the image. Sky light diffuses in the floating water particles in the air and focuses on the road zone, creating one of the darkest zones in the image. These two sources of information were used to realise a more robust system.

The influence of some image processing techniques on the watermark detection rate was discussed by A. Poljicak et al. [47]. There is a decrease in detection rate of a watermark method due to compression, print scan etc. To increase the detection rate various algorithms like unsharp, Laplacian were performed to reduce image degradation. A dataset of 1000 images was considered, watermarked and then compressed followed by scanning. Enhancement of degraded images was done using Laplacian and blind DE convolution filter. Then a comparison of detection rate was done before and after enhancement.

For low contrast image an approach was proposed by Khairunnisa Hasikin et al. [48]. Here, a fuzzy grey scale enhancement technique based on a maximum of fuzzy measures contained in an image was used to enhance the image. The membership function was modified with saturation operator and power law transformation.

Y. Schechner et al. [49] discussed an image enhancement method based on polarisation method. This method, worked under atmospheric and viewing conditions. They analysed the formation of image, through the polarisation effects that occurred due to atmospheric scattering. Then they inverted the procedure to allow the elimination of haze from images. This can also be used with as few as two images captured from a polarizer at different orientations. This method worked at the instant and did not rely on variations in weather conditions. They obtained a good improvement in scene contrast and colour correction. This approach gave a map of the scene, and knowledge regarding properties of atmospheric particles. The limitation was that this method failed under dynamic scenes i.e. when changes were more rapid.

An approach regarding the use of multiple images captured during bad weather scenes was described by S. G. Narasimhan et al. [50]. The key idea in this approach was to exploit the difference between two or more images of the same scene containing different properties of the participating medium. The methods used in here were able to significantly enhance visibility but forced to wait until there was a change in weather conditions and also could not handle the dynamic scene.

S. G. Narasimhan and S. K. Nayar [51] used a single image and tried to acquire the approximated 3D geometrical model of the input scene. In comparison with the earlier approaches, this approach solved the trouble with the requirements of multiple images while it introduced the new complication of the approximated 3D geometrical models,

2.1.2 Moving Object Detection and Tracking in Vision Systems.

Various techniques are used to track underwater cables from underwater videos and images after pre-processing. Sometimes, it is required to avoid moving objects from the scene for efficiently tracking the pipeline. Detecting moving objects will also helpful for avoiding moving obstacles present in front of the vehicle. A review on vision systems for moving object detection and tracking techniques is discussed.

A multisensory fusion technique had been proposed by Balasurya et al. [52]. They described a sensor fusion technique and used a 2D position model. To predict the region in the image taken by a camera mounted on an AUV, 2D position model was generated a few points using the position of coordinates (x_i, y_i) along the cable that helped to detect the most likely region of the cable.

Junichi et al. [53] explained the operation of an automatic underwater vehicle termed as AQUA EXPLORER 2 (AE2) and Aqua Explorer 2000 (AE2000) used for the inspection of submarine cables. In their work, for tracking cables, low frequency currents were applied to conductors in submarine cables. The coaxial alternating magnetic field produced by currents was utilised to locate cables. The sensor placed in front of the vehicle recognised and identified the cable. Their experiment showed that AUV could support mobile sensors as well.

An Autonomous underwater vehicle, which can detect and track an underwater power lying cable automatically on the seabed was described by Antich et al. [54]. The average success rate achieved was 90% for a frame rate greater than 25 frames per second. By using CCD cameras, the cost and size of AUV can be reduced. In this paper, to evaluate cable parameters, the model divided the image in a grid of cells which were processed individually. Then an optimised segmentation process was applied to identify image regions. Once the cable was detected, its location and orientation were estimated by Kalman filter. The experiments were conducted on a Pentium III 800MHz machine.

The proposed vision system by Ortiz et al. [55] tracked cable with an approximate success rate of more than 90%. In this method, first of all, the initial position and orientation was computed from first few images in sequence. The presence of cable was predicted using Kalman filter. For optimising, among the several gradient operators, Sobel operator was used. Hardware included 350MHz, AMD K62 machine executing 320x240 pixels.

Cowen et al. suggested an optical terminal guidance scheme for the docking of an AUV using a beacon [56]. The beacon could be a light emitting device, which could be found using photodetectors on the AUV. This method was similar to a missile locked on to its target. The drawback of this approach was that in shallow water, especially during the day, the photodetector could be locked on to the sunlight. The solution might be in adjusting the frequency of the light coming out from the beacon.

Automated system for event detection from video sequences collected by ROVs during dives was proposed by D. R. Edgington et al. [57]. It was based on a saliency based attention selection system. Events detected were labelled as 'interesting' for video annotator's interesting candidate object presented a special sequence of underwater video, and another essence of "boring" video frames, that did not contain any "interesting" events. After pre-processing to remove some static noise component, saliency model was developed from feature map. The pixels were clustered for dimension calculation after passing through the binary morphological filter, which extracted the edges of the detected object. Tracking of object took place through comparison of each object's position against the expected position for each case, from consecutive frames.

Detection of marine animals visible from video sequences collected by ROV's was proposed by Dirk Walther et al. [58]. Pre-processing was done by background subtraction method and detection by again a saliency based approach with additional feature maps. To avoid detection of marine snow as object, the outcome of orientation filter that detected the edges was normalised. This provided a clear improvement in detecting faint elongated objects representing marine animals.

Vision technique that analysed video sequences collected with static cameras at fish ladder was proposed by Morais et al. [59] . They utilised multi-target likelihood function, which gave comparable likelihoods to hypothesis containing several objects, and a Bayesian filter to track the centroid of their blobs. The Blob analysis tracker updated this function by detecting blobs where all pixels inside the blobs were like an optimised foreground model, and all pixels outside the blobs looks like an optimised background model. The output showed that their method can operate efficiently under severe environmental changes and was able to cope with problems like occlusions. Also, the method supported real-world video streaming, achieving overall efficiency of 81%.

Spampinto et al. [60], proposed a method for detection, counting and tracking of fishes in low quality unconstrained underwater videos. The system was based on a single video camera. Detection was performed by two algorithms, a moving average algorithm and an adaptive Gaussian mixture model which were joint with an ‘and’ operation, purposefully to decrease false positives in the cluttered scenes, to make a background image. The fishes were tracked with a joint venture of two algorithms: first was a feature vector based approach and second was based on the histogram matching using Camshift algorithm. A connected component-labelling algorithm then predicted the counting of fishes in the frame. The model achieved an approximate detection rate of 85% and tracking efficiency of 90%, and a count rate of 85%.

An approach for detection of moving object based on edge based method was proposed by Angel D. Sappa et al. [61]. To detect edges corresponding to a moving object, arithmetic operations were performed between the current frame and the other two equidistant (backwards and forward along the video sequence) ones. In this method, a coarse representation of moving edge was estimated using a canny edge detector. Then, the first stage output was filtered giving rise to an image, which consists of those objects that moved with speed greater than camera's capture rate. Then these two stages were evaluated iteratively to acknowledge all the moving objects present in the current frame. All the non-moving edges were removed during the filtering algorithm. The main advantage of this technique in comparison to those background-modelling techniques was that the proposed approach could be applied whenever it is required, without processing a large part of the video.

Seema Kumari et al. [62] discussed an approach for the moving object detection and object tracking based on the modified frame difference method. Here the variation was examined by using the similarity of frames, in video sequences at defined step length, via the difference of frames. Then applying a threshold moving object were determined. In this method, background modelling was not required. The foreground pixels were set to one, if the deviation was greater than the threshold value otherwise it was set to zero.

Ridden et al. [63], discussed a background subtraction method using a mixture of Gaussians to model the pixel colour that was used for object detection. Here, a pixel in the current frame was analysed against the background model by comparing it with every Gaussian in the model until a match was found. After the match was found, the variance and mean of the founded Gaussian was updated, or a new Gaussian with the mean equal to the current pixel colour and some initial variance was assigned to the mixture.

An automatic moving object detection tracking and recognition method was proposed by Adnan Khashman et al. [64]. For object detection it used two images, first image was the reference image and the second image was current input image. The two images were compared and the variation among pixel values were estimated. In the first case, if the input image pixel values did not match to the pixel values of reference image, then the input image pixel values were thresholded and stored in a third image, termed as output image having a black or white background. The background choice (black or white) of output image was determined by comparing the mean pixel value of the difference between the output image, reference image and input images. If the difference mean pixel value was lesser than a specified threshold value, and then the white image background would be chosen otherwise, the background would be black. In the second case after tracking the moving object motion, the input image in earlier case would now be used as a reference image, and a third image was captured and would be the input image. This process was repeated with the images being captured in every two seconds, and the same comparison method was applied.

Modified statistical mean method was proposed by Vahora et al. [65] to handle the problem which occurred in statistical mean method. First an incremental statistical mean model was initialised. Then it was compared with incremental statistical mean model with k^{th} (initially k=1) video frame. From k number of frames out of total n frames, mean model N was calculated. This mean model N was used for detection of moving object for the first frame. Low pass filters were used as pixel level process for reducing blurring effect as pixel level noise reduction. The process repeated until a total of n frames were compared. This method was computationally fast and eliminated the noise which aroused due to camera, reflectance noise and provided effective results when there was variation of light.

Elgammal et al. [66], utilised Kalman filtering to separate the foreground from the background. The method was based on the assumption that a fixed focal length CCD camera was used for capturing video at stationary condition and assumed that non-rigid objects were moving in a random manner like human bodies. This method undermines the difficulties aroused due to non- continuous movement with variable velocity through suppression of the foreground adaptation. This method successfully detected and tracked human body in real scenes.

To detect moving targets, the pixel wise variation among successive image frames was utilised by Lipton et al. [67]. A metric was put to targets with a temporal consistency constraint to classify them among three categories; first vehicle, second human and third background clutter. This system robustly identified the targets, rejecting background clutter, and kept tracking long distances and over periods despite changes in appearance, occlusions and cessation of target motion. This paper showed a much simpler technique utilising template matching and temporal differencing. Limitation of these methods was that the temporal differencing tracking was not possible if there was any significant motion in camera and it failed when the target was occluded or if it is motionless. In template matching, the target object’s appearance should remain constant.

Classification metric was utilised by Dedeoglu et al. [68] to measure object similarity. This depended on the variation of silhouettes of the identified object regions that were extracted from the foreground pixel map with pre labelled (manually classified) template object silhouettes stored in a database. This method consists of two steps. First step was the offline step that created a template database of sample object silhouettes by manually labelling object types. Second step was an online step that extracted the silhouette of every identified object in every single frame and thus identifying its type by comparing its silhouette based feature to the one in database in real time during surveillance. In the end of comparison of the object against the ones in the database, a template shape with minimum distance was found.

Sethi and Jain [69] discussed an unique approach using proximity and rigidity constraints. This method used two successive frames that was initialised by the nearest neighbour criterion. The algorithm estimated correspondence in the backward direction and forward direction. The limitation of this method was that it could not manage occlusions, entries, or exits. These problems were solved by Salari and Sethi [70]. They first established correspondence for the identified nodes and then extended to the tracking of the unidentified objects by summing a number of theoretical locations.

Rangarajan and Shah [71] discussed another approach, utilising proximal uniformity restraints. Initially correspondences were obtained by computation of optical flow in the starting two frames. The drawback of this approach was that it failed in addressing the entry and exit of objects. If the number of identified points were reduced, occlusion was assumed. It could be overcome by assigning the correspondence for the identified objects in the present frame. For the remaining objects, position was estimated assuming constant velocity.

Intille et al. [72], proposed an improved version for matching centroids of objects, which were identified by background subtraction. The change in the number of objects was handled by investigating specific regions in the image.

In their work, Veenman et al. [73] proposed the common motion constraint. This supported constraint for coherent tracking of points lying on the same object. The drawback was that it was unfit for points lying on isolated objects going in other directions. This approach generated an initial tracks by two pass algorithm, where cost function was minimized by a HA (Hungarian Assignment) algorithm in two successive frames. The main advantage of this approach was that it could handle occlusion and misdetection, under assumption that there was same count of objects throughout the sequence.

Shafique and Shah [74] described a multi frame approach for the preservation of temporal coherency of speed and position. This included formulation of a graph corresponding to the theoretical problem. The graph was estimated by using the points in k frames, which was changed to a bipartite graph by dividing each node into two (+ and −) nodes and by displaying directed edges as undirected edges from + to − nodes. Then window frames were utilised to manage occlusions whose durations were shorter in comparison to the temporal window.

Fieguth and Terzopoulos [75] developed a model by evaluating the average colour of the pixels. To decrease the computational complications rectangular objects were used. The similarity between the object model, M, and the theorised position, H, was analysed by computing the ratio between the colour averages estimated from M and H. The location that gave the largest ratio was selected as the current object location.

To detect an object, Comaniciu and Meer [76] utilised a weighted histogram which was estimated using circular region. They utilised the mean shift procedure, where the tracker maximised the display similarity iteratively by comparison of the histograms, Q, and the window around the theoretical position of object, P. For histogram generation, weighting scheme was utilised by assigning larger weights to the pixels nearer to the object centre.

Comaniciu et al. [77] used the mean shift tracking approach based joint spatial colour histogram. The straightforward advantage of the mean shift tracker against the template matching procedure was the removal of a brutal force search, and the estimation of patch that took less iterations. The constraint mean shift tracking needed was that some part of the object should overlap the circular region upon initialization.

Jepson et al. [78] designed a tracker which would track an object along three component mixture. It consists of the transient features, stable display features, and noise process. The stable component detected the standard display for motion estimation, i.e. motion that did not vary rapidly over time. The transient component detected the rapidly varying pixels. The noise component targeted the outliers in display features, which occurred due to noise. EM algorithm was employed to estimate the parameters of these three component mixtures. Shape of object was ellipse. The motion was depicted using warping the tracked region from one frame to another.

Tao et al. [79] discussed an object tracking method by modelling the whole image, in a set of layers. It utilised only one background layer and a single layer for each object as well. Every layer contains moving model i.e. translational as well as rotational. Layering compensates the background model by projective motion i.e., the object’s motion was predicted using compensated image via 2D parametric motion. After that, pixel’s probability of getting assigned to a layer (object), pl, was evaluated utilising the object’s previous motion and shape characteristics. Every pixel away from a layer, was given a uniform background probability, pb. The model parameters were computed iteratively with an expectation maximisation algorithm. At any time the individual estimation of one set was done, while fixing the others. The variables for each object were iteratively computed until the layer ownership probabilities were maximised.

For tracking Isard and MacCormick [80] discussed a mixed model of the foreground and background regions. The background representation was based on a mixture of Gaussians and foreground object regions was represented by another mixture of Gaussians. The object shape was cylindrical. Under the assumption that the ground plane was already known, then the 3D object positions can be evaluated, there by tracking was done. The state vector included the 3D position, shape and the velocity of all objects. For particle filtering, they proposed an improved approach that can either increase or decrease the size of the state vector either to include or exclude objects. It supported occlusion tolerance between objects under the constraint that the maximum number of objects should be predefined. The main drawback of this method was the use of the same appearance model for all foreground objects, and it needed training to model the foreground regions.

Narimani M. et al. [81] proposed a model to modify the underwater remotely operated vehicles (ROVs) operations which included an underwater robotics system to perform object tracking and intelligent navigation. The image frames were changed to grey images and then edge detection was applied. Then utilising Hough transform the angle of the robot can be evaluated. This angle was taken as reference input of heading controller and with the help of adaptive sliding mode controller, the ROV was able to track the cable or pipeline.

G. L. Foresti and S. Gentili [82] proposed a vision based model for underwater object detection. The proposed model detected a pipeline in the bottom of sea along with some objects in its neighbourhoods automatically. To reduce the light attenuation problems colour compensation was uniquely added and classified the input image using ANN among various objects present in observed scenario. To avoid detection of false objects geometric reasoning was used. The presence of seaweed and sand, different illumination conditions, small variations of the camera tilt angle etc. were taken into consideration for evaluating the algorithm performances.

Tie dong Zhang et al. [83] discussed a online model to detect and track underwater pipeline. This was initially designed for the purpose of vision system of AUV. The real time information of capture image was pre-processed and pipeline features were computed for detection. The Sobel operation was used to avoid false edges, thus clearing the hindrance for Hough transform. After fetching the line information, Kalman filter was used to detect reference zone depicting the probable position of pipeline in an image and predicting the position in next frame. This showed that online model could track the underwater pipeline effectively.

2.1.3 Vision Algorithm Implementation in FPGA

For implementing vision algorithms, various hardware platforms such as general-purpose computers, digital signal processors, and graphical processing units (GPU) are available. Review to the comparison of these hardware platforms was carried out and various implementations of these algorithms in different platforms are discussed.

Asano et al. [84] explained that FPGAs gave better performance in complicated applications whereas GPUs performed better for simple computations (2D convolution) based on an optical flow algorithm. To compare the performances of CPU and GPU following feature were evaluated. They were speed, reliability, size, cost, hardware requirement, trade-offs of higher operational frequency and higher parallelism. The operational frequency of CPU is bit higher than GPU, though GPU is faster than FPGA, also GPU supported large number of cores, running in parallel and its peak literature performance was better than CPU. However, the cores of GPU were grouped and transfer of data between groups was quiet slow. As the parallelism increases, FPGA and GPU performed much better than CPU. FPGA performance was bounded by the size of FPGA and memory bandwidth.

To double the performance a latest FPGA board and larger FPGA were required. A systematic approach was presented by Cope et al. [85] for comparing GPU and FPGA performance based on several image processing algorithms. Two target devices: the NVidia GeForce 7900 GTX GPU and a Xilinx Virtex4 field programmable gate array (FPGA) were chosen and the approach was characterized on parameters such as data dependence, memory access requirements and arithmetic complexity. An FPGA turned out to be better, over GPU, for algorithms that require more numbers of continuous memory accesses, whereas GPU performed better for algorithms with changing data reuse. In case of data dependency, an FPGA performance exceeded GPU performance approximately by eight times.

In this paper, Pauwels et al. [86] showed the comparison of two real time architecture developed using FPGA and GPU devices computing phase based optical flow, stereo and local image features. The power requirements and a smaller clock frequency made the FPGA as the first choice for embedded applications.

Ratheesh et al. [87] compared the performance of an FPGA with a GPU. Relatively slow speed of FPGA was compared against the high speed and fixed architecture of GPU. The author used Symmetric Dynamic Programming Stereo (SDPS) implementation on both. Both systems were provided with end-to-end stereo capability, rectification including lens distortion removal, etc. and produced exactly similar depth maps. GPU implementation was based on an NVidia GeForce GTX 280 and connected to a host personal computer. GTX 280 consists of 30 multiprocessors along with a processor clock of 1296 MHz. For comparison Virtex6 FPGA model was used. For this application, FPGA implementation dominated, despite a much slower internal clock. However, FPGA was limited by space consideration that made it difficult to fit large circuits.

Mahindra et al. [88] found that implementation with FPGA was faster than those of DSP and GPP implementations, and can also exploit large amount of parallelism. FPGA image pre-processing architecture was nearly two times faster than software implementation on an Intel Core2 Duo GPP. He also found that quad micro blaze design was faster than single power pc implementation on FPGA. FPGA devices were developed with significantly higher amount of internal memory and logic resources with much higher bandwidth. The basic advantage of designs based on FPGA lied in the flexibility to exploit the inherently parallel nature of many image-processing applications.

Sami Hasan et al. [89] implemented 2D filtering algorithm using Xilinx system generator. Lower power consumption of 1.57 W at a maximum sampling frequency (230 MHz) was obtained. The different filtering schemes employed are edge, Sobel X, Soble Y, Smooth, Gaussian etc. 5x5 convolutional kernel and was used for the parallel 2D MRI image filtering algorithms. They proved that at low power, image-processing algorithms could be implemented in FPGA architecture.

In their work, Sami Hasan et al. [90], implemented their model, based on Xilinx System Generator development tool of ISE 12.1 development suite targeting Virtex6 FPGA board. The parallel 1D filtering algorithm consists of three stages, first serial to parallel input stage, second parallel processing stage and third parallel to serial output stage. The implementation was successful and they achieved less power and minimum area at a maximum frequency of two architectures.

Zhang et al. [91] used system generator in developing vehicle image detection algorithm. 3x3 kernels were used for edge detection. An image size of 128x120, 24 bit was used. Image data was stored in FPGA block RAM. The implementation was successfully implemented in Spartan 3E board.

Castillo et al. [92] showed that FPGA can be used for implementation of low level operations, where simple computations were performed, but requirement of them was huge, although they have potential to implement complex systems which can perform excessive time consuming computational process. FFT of images were successfully implemented in FPGA with a maximum frequency of 128.485MHz. An automated surveillance system required a processing speed of less than 40ms for real time processing.

Maleena et al. [93] proved from their results that a processing speed of implemented components in an FPGA was constantly faster than on a Matlab or C++ environment. Hardware co-simulation was carried out using system generator. An image size of 576x768x24 bits was used. The architecture has to be modified if the size of the image has to be changed.

Zhiqiang et al. [94] implemented a complete system to reconstruct CT (Computed Tomography) images on FPGA. A cone beam back projection system under FDK algorithm was implemented. FDK algorithm was first introduced by Field Kamp and was a good approximate reconstruction method of cone beam back projection. The authors were able to reconstruct a 132x132 image using FPGA algorithm on the Xilinx Virtex2 FPGA board. As per the authors, the performance would be better if a FPGA board with more DSP blocks could have been used. The performance of hardware implementation significantly outperformed CPU based solution with a speed up of 100xs.

Ila et al. proposed a matching algorithm for motion estimation [95]. Full search block matching algorithm was used for video coding standards of motion compensation. VHDL language was chosen for hardware design.

Modelism simulation tool was used to target Altera Stat family FPGA. The constraints of the design were frame rate performance and memory access. Their experiment showed that the execution of matching algorithm can run 50 times faster in an FPGA based architecture than in a Pentium based PC computer.

An image matching algorithm using Verilog was implemented in PCI connected Xilinx FPGA by Nakano et al. [96]. From a large image, sub images were formed and matching of sub images were found. This algorithm can be applied for finding a particular pattern in VLSI masks, object recognition etc. It was shown that this approach was faster than one that runs in 2.4 GHz Pentium 4 based PC. The computation time for binary image was approximately around 75msec and was bounded by bandwidth of 33MHz, 32 bit PCI bus.

In a study by Che et al. [97], it was found that the GPU outmatched the performance of required FPGA clock cycles on three different applications: data encryption standard, Gaussian elimination, and Needleman Wunsch sequence alignment.

Idaku et al. [98] implemented a high speed hiroshima hyper human vision on a spartan3 FPGA. Vision chips consisting of sensors along with parallel processing circuits integrated on single chips, were developed for high-speed vision. These vision chips were tough to execute for image measurement with higher spatial resolution.

Suthar et al. [99], in their paper gave the presentation of the basics of image processing in model based approach and demonstrated few image processing application done under simulink and implemented using Xilinx

System Generator (XSG). The XSG tool was a newly developed tool in image processing that enabled them to make processing units using Xilinx block sets. XSG also supported software simulation and capable to synthesise on FPGA hardware having parallelism.

Ravi et al. [100] concluded that XSG had been very useful in developing computer vision based algorithms. They focused in the processing of pixel to pixel of an image and modification of pixel neighbourhoods. These transformations can be executed either to the whole image or only to a selected region. Processing an image in real time leads to the implementation on hardware, which offers parallelism and thereby, reduced the processing time significantly.

Elamaran et al. [101] discussed in this paper the real time image processing algorithm that was implemented in FPGA. This had the advantage of using large memory and embedded multipliers that were available on FPGAs. The most basic image processing operations were point processes. Applications like background estimation in videos, image filtering both in spatial and frequency domains and digital image watermarking applications etc. could be easily designed using XSG.

Acharya et al. [102] discussed an FPGA based hardware design for the enhancement of grey scale in an image. They used adaptive histogram equalisation. The AHE technique suited very well for image taken in extremely dark environment or non-uniform lighting environment. This paper showed that reconfigurable FPGAs have both real time and parallel computing nature for the enhancement of images.

Gribbon et al. [103] discussed spatial and temporal parallelism by using FPGA as hardware platform for implementing real time image processing applications. High-level languages and compilers that automatically extract parallelism from the code were not directly compatible to hardware.

Devika et al. [104] explained the use of FPGA in the implementation of real time algorithms suitable for applications in video image processing. FPGA provided basic digital blocks with flexible interconnections thereby achieving realisation of high speed digital hardware. The FPGA consists of a system of logic blocks, such as LUTs, gates or flip flops and some amount of memory. The image was transferred from PC to FPGA board via Universal Asynchronous Receiver Transmitter (UART) serial communication. After filtering, the result was transferred back to PC and the results were compared and validated.

Draper et al. [105] stated that new IP cores were needed to satisfy the newly arrived applications. There was a high demand of applications that included real time video stream encoding and decoding, and/or fingerprint recognition, real time biometric namely face, retina and military aerial and satellite surveillance applications. Simple image operators were faster on FPGAs due to greater input output bandwidth to local memory, although this speedup was not that high (approximately by a factor of ten or even less). More tasks that were complex needed higher speedups. In one experiment, up to a factor of 800x speed was achieved by using parallelism within FPGAs.

Manan et al. [106], in this paper stated the importance of digital image processing and the significance of their implementations on hardware to achieve better performance, for operations such as median filter, morphological operations, convolution and smoothing operation and edge detection on FPGA using VHDL language. Better performance was achieved by FPGA.

Bilsby et al. [107] described the importance of today's military arena for functions such as tactical and strategic surveillance, surface and airborne target acquisition and tracking, self-guided armaments and autonomously guided vehicles. Many application areas where a real time processing solution was required imposed considerable constraints on physical size, power dissipation and cost of the solution. An additional constraint that was often imposed was that of the flexibility of the solution, enabling configurability or modification while in service. The possible need for re-configurability precludes dedicated hardware solutions, which leaves solutions based on general-purpose microprocessors, programmable digital signal processors and FPGAs.

Z. Salcic et al. [108] discussed an approach which improved the image contrast at relatively low cost. Here images used are X-ray images. The High Boost Filtering (HBF) followed by histogram equalisation was applied for enhancing the image. For achieving high performance and low cost, Image Enhancement Co-Processor (IMECO) was proposed.

Chandrashekar et al. [109] used thermo graphic images to perform image enhancement and implementing it on FPGA. The image enhancement capabilities and properties of Successive Mean Quantization Transform (SMQT) were analysed. It showed that a nonlinear and shape preserving stretch of image histogram could be implemented successfully.

Abdullah M Alsuwailen et al. [110] explored nonconventional schemes for real time histogram equalisation to compute histogram statistics and equalisation in parallel. The proposed system used Stratix II family FPGA. The maximum clock frequency used was 250 MHz and the total time needed to operate histogram equalisation on an image of size 256X256 was 0.256ms.

S. Sowmya and Roy Paily [111] have analysed the implementation of contrast stretching, brightness control and histogram equalisation algorithm on FPGA for high-performance DSP applications. For a test image of 100X100 the minimum period in case of all the implementations was 5ns.

Tarek M Bittibssi et al. [112] explored the hardware implementation of five image enhancement algorithms like median filter, contrast stretching, negate image transformation, histogram equalisation, and power law transformation in FPGA and comparison was made.

Nitin Sachdeva and Tarun Sachdeva [113] addressed that the improvement in computational speed was achieved by implementing the different techniques of image enhancement on FPGA. For improving the perception quality of image, they implemented a new system on FPGA for real time histogram equalisation. The system developed was for simultaneous computation of histogram statistics and HE. It depended on the counters used in conjunction with specially designed decoder. They found that the time required for HE of an image of size 256X256 was 0.263ms.

D Dhanasekaran and K Boopathy Bagan [114] considered a high speed nonlinear adaptive median filter for implementation in FPGA. This filter served dual purpose by removing the impulse noise and reducing the distortion of the image. Using this Filtering, image corrupted with an impulse noise of probability greater than 0.2 was successfully filtered.

Tripti Jain et al. [115] presented reconfigurable hardware implementation of median filter with different window sizes. A median filter was designed and implemented on Spartan XC3S500E FPGA.

Yeli et al. [116] implemented median filter and edge detection tasks using double parallel architecture on FPGA. In this scheme, an image was divided into several segments and processed them simultaneously. A drastic improvement in speed was observed even when the scheme was deployed on a low cost FPGA with low frequency and limited resources.

Implementation of an automatic fingerprint recognition system using efficient hardware software architecture was proposed by Francis Fons et al. [117]. Real time and parallel computational intensive demands of the fingerprint image enhancement process was successfully implemented in FPGA.

A DSP/FPGA based parallel architecture oriented to real time image processing applications was discussed by Luxin Yan et al. [118]. The model developed could handle high data transfer bandwidth with low latency as well as high image processing performance.

Mandeep Alluru et al. [119] developed a DCT based image processing system on Xilinx Spartan 3E FPGA. He implemented two different hardware architectures of two-dimensional DCT.

A FPGA based design of vision system for robots was developed by Gholam Hosseini et al. [120]. For robots vision system, it offered the required characteristics such as, parallel processing of the video images, good computational power, low cost and reconfigurable hardware. The image-processing component of the proposed vision system contained one core processor for implementing the software part and up to eight functional units. The core processor controlled and communicated with the functional units. The advantages of both hardware and software techniques were utilised using mixed hardware/software method. They proved that the core processor could perform all parallel operations to minimise the total number of implementation stages.

2.2 Inference from Literature Review

It is observed from the open literature is that image enhancement is needed for an underwater vision system, as the images captured are suffered by various attenuation. Several image enhancement techniques were reviewed. From these reviews, it is seen that pre-processing could be done using techniques like homomorphic filtering, histogram equalisation techniques, dark channel prior method, fusion method, CLAHE etc. Various modifications were made from the existing and conventional methods to improve the quality of the image. In uneven illumination environments, dark channel prior method and fusion method outperformed other enhancement techniques.

Object detection and tracking are important applications employed in underwater vehicles. AGMM is a popular technique for object detection in a complex environment. Back ground subtraction were also performed for simple applications. Automation is required to reduce human efforts for carrying out these operations. Hough transform is a popular technique adopted for line detection. Baysen filtering, Particle filtering, Kalman filtering and alpha beta filtering can be employed for tracking. Kalman filtering and particle filtering are computationally complex and accuracy is less for Baysen filtering.

From the literature review, it is studied that FPGA is a better alternative for image processing applications for its low power and parallel nature. Hence, FPGA is preferred as the implementation platform.

2.3 Summary

This chapter addresses literature survey carried out for various algorithms used for image enhancement, object detection and tracking and different ways of implementation methods.

CHAPTER 3

METHODOLOGY

This chapter addresses the methods adopted for the implementation of the underwater image enhancement, object detection and pipeline tracking system in FPGA. Underwater image enhancement was carried out as a pre-processing technique using multi-scale based image fusion technique and comparison with other underwater image enhancement techniques such as homomorphic filtering, AHE and retinex algorithm has been carried out. The pre-processing was done before object detection and pipeline tracking. Object detection was done by a novel Adaptive Gaussian Mixture Model based method and has been compared with normal background subtraction method and AGMM. Hough transform followed by modified alpha beta tracking was used for pipeline tracking. All designs were implemented in FPGA.

3.1 Introduction

Underwater image acquisition is challenging, as the light propagating through the water suffers from exponential attenuation with the increase in the depth of the water. The non-uniform illumination of the seafloor and the bluish image formed also adds to the difficulty in the application of underwater image processing algorithm.

The method for inspection of underwater pipelines is mainly based on ROVs, which require human intervention. However, this technique is hazardous. Hence, the solution is to support the human activity using an intelligent vision-based navigation and guidance system. This system involves an efficient method for vision-based target detection and tracking methods in an underwater environment. The thesis attempts to implement an efficient marine image enhancement, object detection and pipeline tracking algorithm in FPGA.

3.2 Basic Model of Proposed Method

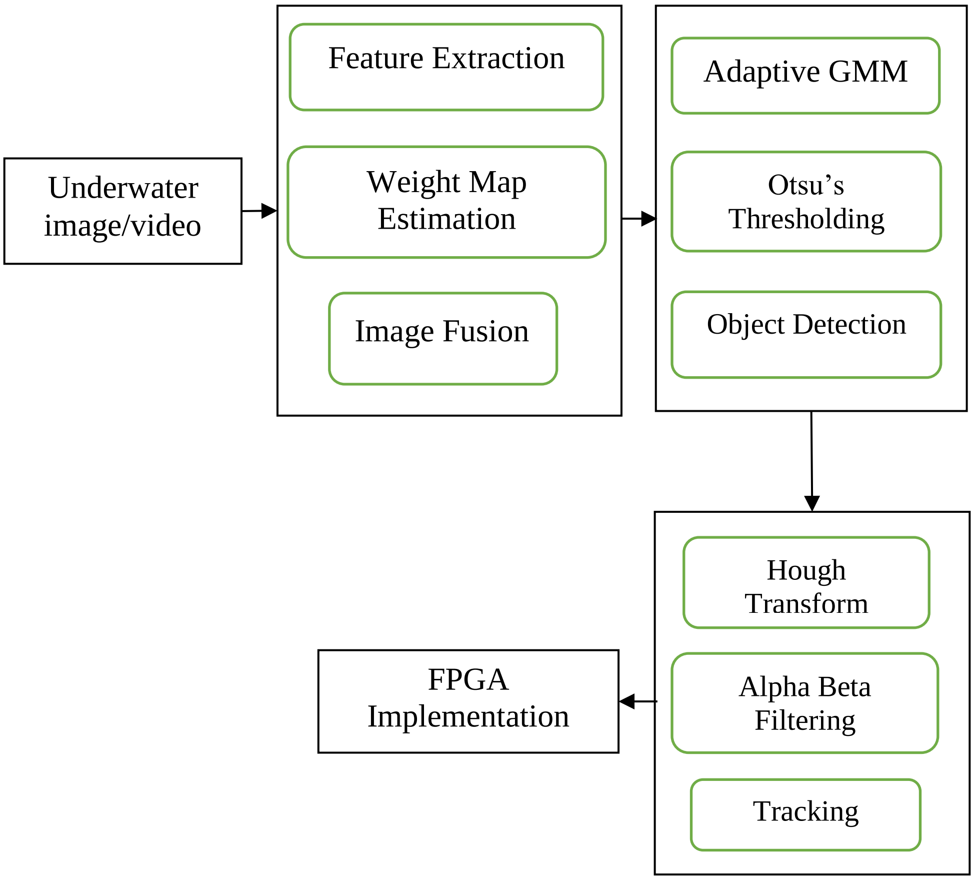


Fig. 3.1 Proposed vision system

Fig. 3.1 shows the basic model of the proposed system. Here, the underwater image is enhanced using single image based multi-scale fusion. Moving object detection is carried out to separate moving objects from the scene. It helps to avoid any moving obstacles in front of the vehicle. Then pipeline tracking is carried out using Hough transform followed by improved alpha beta filtering. A brief explanation of each block is given in the following sections.

3.3 Multi-Scale Pixel based Image Fusion

Multi-Scale Fusion technique is used to improve the visibility of poorly illuminated areas by using multi-scale fusion based strategy [22]. The fundamental principle of this method is extracting the best features from the given input image and then to apply the estimated perceptual based qualities called weight maps to them and finally fuse them together to form the enhanced output. In the fusion-based strategy, two inputs derived from the single image will be by white balancing and by median filtering followed by contrast stretching. White balancing is used to reduce colour casting. From the individual input image, estimation of four weight maps is done. One is exposedness weight map, which measures the exposedness of pixels not exposed to underwater constraints. Laplacian weight map assigns high values to textures as well as edges. Colour cast weight map, introduced newly, increases red channel value thereby reducing colour cast. Saliency weight map measures the amount of discernible concerning other pixels. These weight maps are computed and applied to both the derived inputs. This method is a per pixel implementation.

3.4 Adaptive GMM based Object Detection

The method can be used to detect moving objects while tracking pipeline and hence to remove them from the image by separating foreground and background. The foreground is the moving object, and by identifying them, one can quickly get the desired background. The raw data, visual information, collected using camera can be utilised as the input to the object detection system. Adaptive Gaussian Mixture Model [121] in an improved form is used for moving object detection and is compared with subtraction method. FPGA implementation of both the methods has been carried out.

The pre-processing stage performs the image processing tasks to make the video frames to a peculiar scheme that expose the concealed information and sharpen the object detection capability of the system. The processed input video frames are utilised to extract the background model and related normalisation parameters. For the sake of good detection attributes, the background model key metrics get updated recursively according to the available new information from each pixel in the current frame. In the moving object detection stage, a threshold based scheme separates the foreground from the background distributions efficaciously employing the parameters already derived in the previous step of the system. Post-processing stage assures the required level of visual quality of the result by thresholding and connected component analysis technique [122].

3.5 Underwater Pipeline Tracking

Automatic underwater vehicles are robots that can navigate through the marine environment without any human intervention. AUVs carry precision sensors as the payload and are used mainly for survey missions, oceanographic mapping, and infrastructure inspection. As the duration of mission increases, more advanced navigation systems need to be used [123].

Navigation is the process of determining one's position, i.e. localisation and then discovering a path to advance further from the existing location. The AUV used for the survey missions in an underwater environment requires highly advanced precision navigation systems. Navigation usually requires high speed and high accuracy computation. It is done through different techniques of which the vision-based navigation is the cheapest, as it needs only a single camera.

Underwater navigation poses further challenges due to the highly dynamic and complex environment. One of the most critical aspects of navigation is the knowledge of the situation and to get the knowledge of the situation, different types of sensing techniques are used [36] as explained below.

- Inertial navigation is mainly based on motion sensors or accelerometers and rotation sensors or gyroscopes. Using these sensors the position velocity and orientation of AUV can be calculated via dead reckoning [124].

- Acoustic navigation is primarily based on acoustic transponder beacons. During pipeline inspection, the acoustic sensors that act as microphones listen for sound generated when fluids under high-pressure leak.

- Geophysical navigation is based on estimating the location of the AUV using the environment, surrounding the AUV [125].

Side scan sonar, magnetometers, sub-bottom profilers are some of the sensors used for pipeline/cable inspection in the underwater environment [34]. For example, a magnetometer that can sense ferrous materials can be used to detect the existence of a pipeline. Side scan sonar together with a video camera can be used to inspect pipes that usually project out of the ocean bed, and they can make a proper detection even when water is not clear.

Once a pipeline/cable is detected, different algorithms and high-intensity computation are employed to calculate the orientation of the pipeline/cable and navigate along it. Hence, the quality of navigation depends heavily on the accuracy of these sensors. Magnetometers have the disadvantage that they can detect only ferrous materials. Also, they cannot give an idea of whether the object is buried or projecting out. Using side scan sonar is a costly option as the use of sonar means the mandatory use of companion equipment like transponders. Another condition is that these sensors should be able to adapt to the highly dynamic marine environment.

Vision sensors and the visual data captured by them provide a new way for autonomous underwater gadgets to perceive surroundings. Vision sensor data analysed using specific computer vision algorithms can give highly accurate detailing about the surrounding environment.

Advantages of using 3D imaging technique are that they can be used to detect and track pipelines/cable which are buried or projecting out of the seabed [83]. However, these techniques are expensive. A cheaper and more accessible solution is to use a camera. Cameras are attractive in the sense that they offer more precious information about the surrounding environment even though the detection range is limited in the poor visibility condition of the underwater environment.

For vision-based tracking, a novel alpha-beta filter based tracking algorithm is employed and is explained below.

3.5.1 Alpha Beta Filter for Pipeline Tracking

The commonly used tracking filters are Kalman filter, extended Kalman filter, particle filter etc. [126]. But the computational complexity of these filters are very high, and in most of the real-time cases, they are not preferred [127]. α- β filters, a particular case of Kalman filter, are simple to implement as they are computationally simple. They assume constant acceleration in between sampling interval [128]. α- β filters are based on position and velocity measurements.

3.6 Field Programmable Gate Array

Many image-processing applications require that several operations need to be performed on each pixel in the image resulting in a large number of operations per second. Continual growth in the size and functionality of FPGAs over recent years have led to an increasing interest in their use as implementation platforms for image processing applications, particularly real-time video processing [129]. FPGA is a collection of logic elements that can be electrically rewired. FPGA implements an application by developing separate hardware for each function, and hence such designs are inherently parallel. Each instruction that the programmer enters will be mapped into a different hardware component. Thus, such a configuration is suitable in those image-processing algorithms, which has a significant amount of parallelism in them.

An FPGA based circuit implements several operations in one clock cycle simultaneously. This allows clock speed to be lowered significantly. Reduction in clock speed corresponds to a reduction in dynamic power consumption of the system. Thus FPGA based design facilitates a low power design.

FPGAs offer the following advantages that can be referred as the reason for the increased acceptance that gained in last few years [92],

- High Speed: Look up tables shares an essential role in the swift execution of FPGA as it is mainly based on it. This enables FPGAs to have prompt implementation related to ASIC technology.

- Low Cost: FPGA is economical, for the same reason it is designer-friendly. As FPGA architecture primarily based on lookup tables, its energy consumption is so minimal.

- Fast prototyping and turn-around time: Process of fabricating a real circuit to a theoretical layout to check its operation and to debug the core on a physical platform, is called prototyping. Turn around is duration from starting of an operation to its completion. Interconnects present in FPGA can be configured to get the desired output. Thus, time is saved related to any manual design or application specific integrated chips.

The hardware description languages which include VHDL and Verilog, are frequently used for FPGA programming [130]. Xilinx FPGA is chosen to program the algorithm. As Xilinx FPGA is used, Xilinx ISE Design Suite is used as the software platform for implementation. System generator and Altium designer are used as front end tools in association with Xilinx design suite. As system generator allows only one dimensional data processing, programming is tedious for complex algorithms. Hence Altium designer is also used. Xilinx Virtex 4 is the target FPGA for the implementation of all designs irrespective of the front end tools.

3.7 Summary

The basic model of the proposed thesis work is presented in this chapter. The proposed method used for underwater image enhancement is described briefly. A brief introduction of the algorithm used for moving object detection and tracking is also explained. An introduction to FPGA is also narrated.

CHAPTER 4

MULTI-SCALE PIXEL BASED IMAGE FUSION

The wavelength of light rays are considerably smaller than the diameter of water droplets and other suspended particles. Hence the light rays are scattered and even absorbed by these particles, which in turn reduces a number of light rays striking on the object [131]. This poor illumination causes low visibility and results in degraded underwater images taken from the underwater vision system. Improved multiscale image fusion is used for image enhancement. New weight maps, suitable for underwater image enhancement, are incorporated for better visual appearance.

4.1 Introduction

Underwater video sequences are being used to monitor marine species, tracking pipeline, etc. Manual monitoring of underwater video presents labour-intensive processing techniques. Therefore a high speed automatic skilled system for data manipulation is needed, which includes pre-processing of underwater images. Underwater vision has less visibility with low contrast and colour disparity. Earlier, due to limitation for visible light in the underwater scenario, other techniques such as sonar ranging was widely used. But, recently, with the development of high quality underwater cameras, vision-based navigation plays an important role in ocean exploration [132].

For researchers, as the available underwater video data is increased, the methods in handling data are tiresome. That is, monitoring of these video sequences requires an expert to watch and mark interpretations, and then have to enter the annotations into a database. Data manipulated in a period will be less. Labour-intensive process of huge volume data is unmanageable, resulted in need of high-performance automatic system.

4.2 Underwater Vision System

The underwater vision system includes integrating as well as automating a process that is used for visual sensitivity. It assists in reinventing the furthermost needed skills of a visual system, making use of digital hardware. Earlier work on image pre-processing has given less attention to underwater images [133].

Underwater images can be enhanced using methods such as nonlinear filtering, single scale retinex method, adaptive histogram equalisation, multi-scale image fusion etc.

4.3 Multi-Scale Fusion Technique

Image fusion is the process of combining relevant information from two or more images into a single image. It is possible to derive more than one input from a single image and to perform fusion. The process of image fusion can be performed at three different levels of information representation, such as pixel, region or decision-level[134][135].

4.3.1 Pixel-Level Image Fusion

The combination of information at the lowest level represents the pixel level information, since every pixel in the fused image is determined by a set of pixels in the source images. Usually, this set consists of a single pixel or comprises of all pixels within a small window, typically of size 3x3 or 5x5. The advantage of pixel-level fusion, apart from its easy and time-efficient implementation, is that the resulting image comprises the actual information from the sources. However, since pixel-level fusion methods are very sensitive to miss registration, co-registered images at sub-pixel accuracy are required.

4.3.2 Region-Level Image Fusion

Region-level fusion method typically starts by extracting all salient features from the various input images. Salient features are extracted by applying an appropriate segmentation algorithm which identifies all salient features within the input images concerning certain properties such as size, shape, contrast, texture or grey-level. Based on this segmentation, a region map can be created which will link each pixel to a corresponding feature. Consequently, the fusion process is performed on the extracted regions. Though the final fusion performance heavily depends on the quality of segmentation, it has certain advantages such as low blurring effects, high sensitivity to noise. Degradation in the fused image will occur if over or under-segmentation is done.

4.3.3 Decision-Level Image Fusion

Fusion at decision-level provides the highest level of abstraction by allowing the information from multiple sensors to be effectively combined. Here, a decision map is built for each source image by performing a decision (labelling) procedure on all input pixels. Finally, a fused decision map is constructed based on the individual decision maps. For this purpose, decision rules are used which reinforces common interpretation and can resolve differences between the individual decision maps.

The appropriate level decisions depend on different criteria such as the underlying application, the characteristics of the physical sources as well as factors like execution time and the available tools. However, there exists a strong inter-linkage between the different levels of image fusion. Many fusion rules that are used to determine the individual pixels in the composite image at pixel-level can, for instance, also be used at region-level to fuse the extracted features. Furthermore, decision-level fusion often resorts to the segmentation map created at region-level to aid the decision-making application fields. Image fusion has attracted a great deal of attention in a wide variety of application areas in the last decades. All imaging applications that require the analysis of more than one image can benefit from image fusion.

4.4 Proposed Multi-Scale Fusion Method

Underwater images corrupted by lighting conditions may lose the visibility of the scene considerably. Enhancing approach is not available to remove entire haze effects in degraded images. The proposed algorithm includes deriving two inputs from the single degraded image and it recovers colour and visibility of the entire image. Colour correction is applied to the image after the fusion process.

In the proposed algorithm, two inputs are derived from a single degraded input. The first derived input is obtained by white balancing the input. This step aims at removing chromatic casts in the image. More attention is given to red channel of the image as it attenuates more in underwater. The second derived input I_2 is capable of enhancing those regions having low contrast. It is obtained by contrast stretching the median filtered input image. Therefore, the first derived input I_1 avoids colour casts and the second derived input improves the uneven contrast.

The important features of these derived inputs are computed using different weight maps. The different weight maps derived in the proposed model are exposedness, saliency, Laplacian and colour cast. The weight maps are normalised to avoid artefacts. After normalising the weight maps, the resultant images are fused to form a single image. Flowchart of the improved multi-scale fusion method is shown in fig. 4.1.

4.4.1 White Balancing

White balancing is the method of removing additional and unrealistic colour casts produced by the environment. In underwater environment due to scattering and wavelength selective absorption, additional colour casts are introduced. To remove these effects, white balancing method is used. The most common and widely used white balancing algorithm is grey world theory [136].

4.4.1.1 Grey World (GW) Algorithm

The GW algorithm is based on the property that average colour of the image is grey. This condition do not hold by the underwater images, because as the depth of the ocean increases the red channel of the image attenuates faster than the other colour channels. To obtain the colour cast free underwater images, grey world algorithm is modified.

The first step of the GW algorithm is calculating the averages of R, G and B channel, i.e., R_{avg}, G_{avg}, and B_{avg} as in eqn. 4.1.

$$\begin{cases} R_{avg} = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} I_r(x,y) \\ G_{avg} = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} I_g(x,y) \\ B_{avg} = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} I_b(x,y) \end{cases} \tag{4.1}$$

where, $I_r(x,y), I_g\,(x,y)\ and\ I_b(x,\,y)$ are the intensity values of red, green and blue channel of the image respectively. After finding out the individual channel average values, the average of these three can be found out.

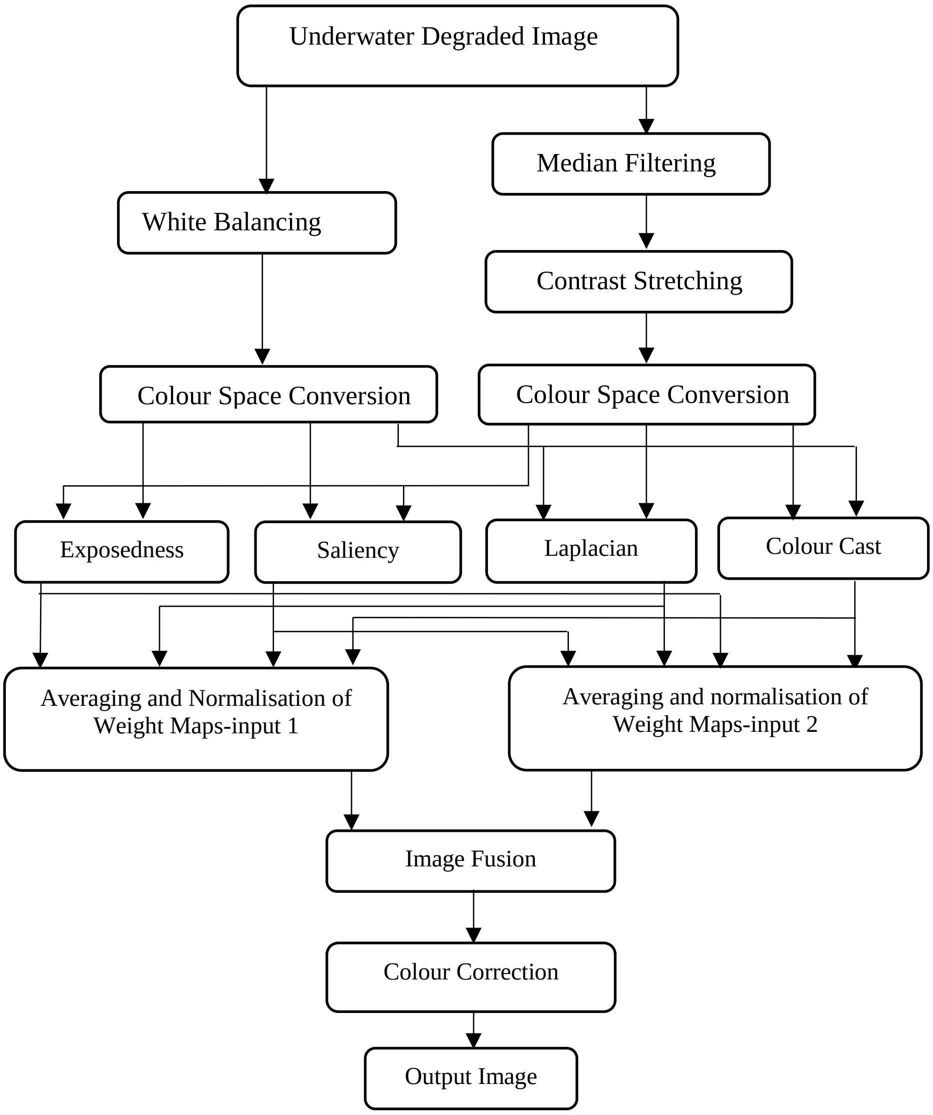


Fig. 4.1 Flow chart of proposed multi-scale image fusion

The average value A_{avg} is calculated by averaging the values obtained from eqn. 4.1 and is given in eqn. 4.2.

$$A_{avg} = (R_{avg} + G_{avg} + B_{avg})/3 \tag{4.2}$$

The individual colour value of each pixel is adjusted as per eqn. 4.3.

$$\begin{cases} I'_r = I_r(x,y)A_{avg}/R_{avg} \\ I'_g = I_g(x,y)A_{avg}/G_{avg} \\ I'_b = I_b(x,y)A_{avg}/B_{avg} \end{cases} \tag{4.3}$$

In eqn. 4.3, $I_r, I_g \; and \; I_b$ are the original pixel values and $I_r{}', I_g{}' \; and \; I_b{}'$ are the adjusted values by the grey world method.

4.4.1.2 Modified Grey World Algorithm

As the R channel value in the underwater image is very low, colour correction becomes erroneous. In this thesis, a novel approach of augmenting the red channel value as per eqn. 4.4 is introduced. This is done by adding 45% of the mean of green and blue channel values to the mean of red channel value. 45% is the optimum value chosen based on trial and error method and is shown in eqn. 4.4.

$$R_{avg} = R_{avg} + 0.45 \times \frac{(G_{avg}+B_{avg})}{2}$$

(4.4)

To equalise the channels, the grey world algorithm is modified as in eqn. 4.5 which gives better results over the existing algorithm.

$$\begin{cases} I'_r = I_r(x,y) + \left(A_{avg} - R_{avg}\right) + \; (I_{avg} - I_r(x,y)) \\ I'_g = I_g(x,y) + \left(A_{avg} - G_{avg}\right) + \; (I_{avg} - I_g(x,y)) \\ I'_b = I_b(x,y) + \left(A_{avg} - B_{avg}\right) + \; (I_{avg} - I_b(x,y)) \end{cases} \tag{4. 5}$$

where $I_{avg} = \; (I_r + \; I_g + \; I_b)/3$, the average intensity value.

Though white balancing removes additional and unrealistic colour casts, it alone cannot solve the problem. Hence colour correction is required after fusion. The white balanced output, which is the first derived input is then converted to YC_bC_r Colour space model for applying various weight maps to the interested colour channel as per the design described in the coming sections.

4.4.1.3 Modified Contrast Stretching

The second derived input is obtained by contrast stretching the median filtered degraded image. In colour contrast stretching model, the R, G and B channels are stretched to 0 to 255 range. As this might lead to overstretching problem [137], the formula can be modified as,

$$\begin{cases} I'_r = \frac{I_r - R_{low}}{R_{high} - R_{low}} \text{ x } A_{max} + A_{min} \\ I'_g = \frac{I_g - G_{low}}{G_{high} - G_{low}} \text{ x } A_{max} + A_{min} \\ I'_b = \frac{I_b - B_{low}}{B_{high} - B_{low}} \text{ x } A_{max} + A_{min} \end{cases} \tag{4.6}$$

where, Amax is the maximum value of R_{high}, G_{high} and B_{high} and A_{min} is the minimum value of R_{low}, G_{low} and $B_{low.}$ To avoid over stretching, maximum and minimum values of the original image are taken as the limit.

4.4.2 Median Filtering

Underwater image is noisy by nature and hence to obtain a noise free image, the noise needs to be eliminated using a suitable filter. The noise is removed by median filtering. Edges are preserved when median filter is applied to an image. This nonlinear operator arranges the intensity value of the local window image pixels in order and replaces the value in the output image pixel by the middle value.

4.4.3 Weight Maps

The weight maps play a critical role in the outcome of the final fused result. The weight maps generated should have a non-negative value. The weight maps are some measures of the input image. It represents the finer details of the image [138]. The finer details from each image have to be extracted out and fuse them together to form the enhanced image. The weight maps are to be designed carefully to extract the details. New weight maps were used which showed better results in enhancing degraded underwater images. The different weight maps are Laplacian, saliency, colour cast and exposedness.

4.4.3.1 Laplacian Weight Map

It is the measure of visibility of each pixel such that it gives higher values to high visibility areas and smaller values to the remaining areas. The main aim of this weight map is to assign a high value to edges and textures. Laplacian filter is a better choice for edge preserving. Laplacian of an image accentuates the regions having abrupt or rapid intensity change. The second derivative measurements are very delicate to noise. Since the images were noise reduced, further noise reduction is not needed. Even though it preserves edges, this weight map doesn't give the details of contrast since it can't discriminate the flat, valley or rapid regions. The weight map is computed as shown below

$$W_L^k = \sqrt{1/3((R^k - Y^k)^2 + (G^k - Y^k)^2 + (B^k - Y^k)^2)}$$

(4.7)

where Y is the luminance component of the image in YC_bC_r model.

4.4.3.2 Saliency Weight Map

One of the main challenges in underwater image enhancement is that the objects within the image lose their visibility and hence discrimination of the objects from the background scene becomes difficult. The quality that makes the object distinctive relative to its neighbours is known as saliency. It is based on the concept of centre-surround contrast wherein a saliency map is developed so that the contrast of the main object of interest is enhanced. Another main advantage of this approach is that the mid values of the image are not affected while applying this method.

Saliency weight map can be expressed as in eqn. 4.8.

$$W_s(x,y) = \|Y_\mu - Y_{\omega hc}(x,y)\| \tag{4.8}$$

where Y_μ is the mean luminance value of image and $\mathrm{Y}_{\omega\mathrm{hc}}$ is the Gaussian blurred version of the luminance channel that aims to remove high frequency noise. A 5x5 separable kernel is used for image blurring.

4.4.3.3 Colour Cast Weight Map

A novel weight map, colour cast weight map, is introduced to reduce the colour cast present in the underwater image. To find the colour cast weight map, chrominance-red *(C_r)* channel of the image is first found out from the YC_bC_r colour space. Then the standard deviation between C_r value of each pixel and the average value of the surrounding region is computed. Eqn. 4.29 represents the equation for computing colour cast weight map, where Cr^k represents the chrominance-red channel of the image and $\mathrm{Cr}^k{}_{avg}$ represents the low pass filtered chrominance-red channel.

$$W_C(x,y) = \|\mathrm{Cr}^k(x,y) - \mathrm{Cr}^k{}_{avg}\ )\|$$

(4.9)

4.4.3.4 Exposedness Weight Map

In an underwater image, all pixels will not be exposed. Pixels are commonly better exposed when they have normalised values close to the average value of 0.5. This weight avoids an over or underexposed look by constraining the result to match the average luminance. Mertens et al.

[139], employed a similar weight in the context of tone mapping. The exposedness weight map is expressed as a Gaussian-modelled distance to the average normalised range value (0.5) and is shown in eqn. 4.10.

$$W_E(x,y) = \exp\left(-\frac{(Y^k(x,y)-0.5)^2}{2\sigma^2}\right) \tag{4.10}$$

where Y^k *(x, y)* represents the luminance value of the pixel at location *(x, y)* of the input image, while the standard deviation is set to $\sigma = 0.35$.

4.4.4 Fusion

Image Fusion is a process of combining the relevant information from a set of input images into a single image, where the resultant fused image will be more informative and complete than the input images. In this work, there are three weight maps based on luminance channel and one based on chrominance-red channel. The effective weight map from the three weight maps, W_L, W_S and W_E are combined by averaging the weight map values as in eqn. 4.11. As a result of this operation, two weight maps are obtained for each derived input. Two colour cast weight maps from the derived inputs, W_C, are considered separately. Then these weight maps are normalised as shown in eqn. 4.12. W^k_{Normy} and W^k_{Normc} are the two normalised weight maps for luminance and chrominance-red channel respectively. Finally they are fused using the eqn. 4.13 for luminance channel and using eqn. 4.14 for chrominance-red channel to obtain the fused image.

$$W^k_{Avg} = \frac{1}{N}\sum_{i=1} W^k_i \tag{4.11}$$

$$W^k_{Norm} = \frac{W^k}{\sum^2_{i=1} W^k} \tag{4.12}$$

$$Y(x,y) = \sum_{k=1}^{2} W_{Normy}^{k}(x,y)Y^{k}(x,y) \tag{4.13}$$

$$Cr(x,y) = \sum_{k=1}^{2} W_{Normc}^{k}(x,y)C^{k}(x,y) \tag{4.14}$$

4.4.5 Colour Correction

Many qualitative techniques based on imaging have been proposed for colour correction in the literature [140][141]. They are based on image restoration techniques or subjective techniques such as image enhancement methods. They require some assumptions or knowledge of physical media. In this work, $l\alpha\beta$ colour space based method is used for colour correction which do not require any knowledge of physical media.

4.4.5.1 lαβ Colour Space

The l$\alpha\beta$ colour space was proposed by Ruderman et al. [142] by studying the human eye perception of natural images. The $l\alpha\beta$ colour space consists of two opponent colour components α and β, (which encode yellow-blue and red-green chromaticity) and a luminance component l (achromatic). This can be used as reference for hue or for colour correction [143]. In the human colour visual mechanism, this colour space is used for colour discrimination. With the help of $l\alpha\beta$ colour space, the chromatic components are relatively altered by displacing their distributions closer to the white point (white balancing). Histogram cut off and stretching of the luminance component can be performed to improve the contrast of the image. So, by performing a colour correction similar to that in the human visual system colour constancy in underwater images can be achieved [144]. In the image formation model, an underwater image $\mathrm{f_i(m,n)}$ can be expressed by,

$$f_i(m,n) = g_i(m,n)\, r_i(m,n) \tag{4.15}$$

In eqn. 4.15, $g_i(m,n)$ is the illuminant component and $r_i(m,n)$ is the reflectance component. Here, (m,n) represent image coordinates and i Є {1, 2, 3} indicate the colour components R, G, and B of the image. Colour correction is achieved by computing the illuminant g_i and removing its chromatic components from f_i.

4.4.5.2 Steps to Colour Correction

The model for colour correction is shown in fig. 4.2. The various steps to colour correction include gamma correction and conversion to XYZ, LMS and $l\alpha\beta$ colour space. `

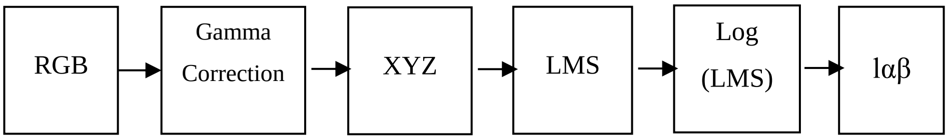


Fig. 4.2 Model for colour correction

The non-linear RGB image is corrected by the non-linearity correction (gamma correction), to get linear RGB coordinates for further processing. Next step involves conversion to XYZ tristimulus values that is obtained by multiplying the linear RGB coordinates $\mathrm{f_i(m,n)}$ with $T_{xyz,ij}$ matrix. From this device-independent XYZ space, the image is converted to LMS space using the $T_{lms,ij}$ matrix. Now, transform the data into logarithmic space, and finally de-correlate these axes by utilising Principal Component Analysis (PCA), that means multiply $l_{log,i}(m,n)$ vector by the decorrelation matrix $T_{pca,ij}$. Thus, vector coordinates of image in $l\alpha\beta$ space are obtained. The three resulting principal axes are orthogonal, where one axis *l* represents an achromatic channel, while the other two (α and β)

channels are chromatic yellow-blue and red-green opponent channels [145]. The various steps are given in eqn. 4.16 to eqn. 4.21.

- Conversion from RGB to XYZ space

$$x_j(m,n) = T_{xyz,ij}\, f_i(m,n) \tag{4.16}$$

$$T_{xyz,ij}\, f_i(m,n) = \begin{bmatrix} 0.5141 & 0.3239 & 0.1604 \\ 0.2651 & 0.6702 & 0.0641 \\ 0.0241 & 0.1228 & 0.8444 \end{bmatrix} \tag{4.17}$$

- Conversion from XYZ space to LMS space

$$l_j(m,n) = T_{lms,ij}\, x_i(m,n) \tag{4.18}$$

$$T_{lms,ij}\, x_i(m,n) = \begin{bmatrix} 0.3897 & 0.6890 & 0.0787 \\ -0.2298 & 1.1834 & 0.0464 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \tag{4.19}$$

- Conversion from log (LMS) space to $l\alpha\beta$ space

$$l_{l\alpha\beta,j}(m,n) = T_{pca,ij}\, l_{log,i}(m,n) \tag{4.20}$$

$$T_{pca,ij} = \frac{1}{\sqrt{6}} \begin{bmatrix} \sqrt{2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \sqrt{3} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \tag{4.21}$$

4.5 Other Underwater Image Enhancement Methods

Various underwater image enhancement algorithms, suitable for underwater vision systems, are presented in open literature [13]. The commonly used enhancement methods, which were implemented in FPGA, for enhancing the visual quality of underwater images, are described in this section.

4.5.1 Homomorphic Filtering

Traditional image enhancement techniques like histogram equalisation, are spatially invariant. Therefore, enhancement is performed on the entire image devoid of taking the object's distance within the image. Problem occurs when the visibility of object depends on the distance from the camera. Homomorphic filtering is a nonlinear filtering technique that reduces the non-uniform illumination and enhances the contrast. It is a generalised technique for image enhancement and/or correction that simultaneously normalises the brightness across an image and increases its contrast. This frequency based nonlinear filtering method can be used for correcting the non-uniform illumination.

In the illumination reflectance model, an image can be represented as a multiplication of illumination and reflection component as in eqn. 4.22.

$$f(x,y) = i\,(x,y)\ r(x,y) \tag{4.22}$$

In eqn. 4.22, f(x, y) is the captured image, i(x, y) is the illumination multiplicative factor and r(x, y) is the reflectance function. The illumination element signifies the low frequency that changes slowly through the field. Reflectance component denotes high frequency components. On multiplying with high pass filter, which removes low frequencies, it results in the reduction of non-uniform illumination within the image [14][13]. Eqn. 4.22 cannot be used directly to operate separately on the frequency components of illumination and reflectance because the Fourier transform of the product of two functions is not separable. Instead the function can be represented as a logarithmic function wherein the product of the Fourier transform can be represented as the sum of the illumination and reflectance components as shown below.

Taking logarithm of eqn. 4.22 to convert multiplicative into additive,

$$g\ (x, y) = \ In\left(f(x, y)\right)$$

$$= \ ln\ (i\ (x, y))\ +\ In(r(x, y)) \qquad (4.23)$$

Fourier Transform of eqn. 4.23 is taken,

$$G\ (k, l) = \ I\ (k, l) + \ R\ (k, l) \qquad (4.24)$$

Applying a filter, *H (k, l)* to the image in eqn. 4.24,

$$S\ (k, l) = \ G\ (k, l).H\ (k, l)$$

ie., $S\ (k, l) = \ [(\text{-I}\ (\text{k}, \text{l}).\text{H}\ (\text{k}, \text{l}))\ + (\text{-R}\ (\text{k}, \text{l}).\text{H}\ (\text{k}, \text{l}))]$

(4.25)

Representing eqn. 4.25 in spatial domain,

$$s\ (x, y) = \ \left[\left(F^{-1}\left(I\ (k, l).H\ (k, l)\right)\right) + \left(F^{-1}\left(R\ (k, l).H\ (k, l)\right)\right)\right]$$

(4.26)

The exponential of *s(x, y)* is taken to obtain an enhanced image [146].

4.5.2 Adaptive Histogram Equalisation

Histogram equalisation is a technique which is used to improve the quality of an image captured under poor lighting conditions. The standard procedure in this case, is to re-map greyscales of the image so that the resultant histogram approximates that of uniform distribution. This procedure is based on the assumption that the image quality is uniform over all areas and one unique greyscale mapping provides similar enhancement for all regions of the image [113]. However, when the distributions of greyscale change from one region to another, this assumption is not valid.

In this case, an adaptive histogram equalisation technique can significantly outperform the standard approach [147].

The main idea in adaptive histogram equalisation is to find the mapping for each pixel based on a local (neighbourhood) greyscale distribution. Here contrast enhancement mapping is applied to a pixel, which is a function of the intensity values of surrounding pixels. Hence, the number of times that this calculation is repeated is same as the number of pixels in the image. This gives rise to an extensive computation requirement, which even with some modifications cannot be used for real time image enhancement. So another form of adaptive histogram equalisation is used which is a compromise between global histogram equalisation and fully adaptive histogram equalisation i.e., regional histogram equalisation [148]. Here the image is divided into limited number of regions and the same histogram equalisation technique is applied to the pixels in each region. Hence, computation is reduced considerably and the method can be used for real time image processing.

In this thesis, YC_bC_r colour space is used for implementing AHE. Here, the luminance channel is separated from the chrominance channel. Then it is divided into sub images and histogram equalisation for each sub image is carried out. The new histogram equalised Y component is superimposed with the old C_b and C_r components to get the histogram equalised colour image as per the eqn. 4.27 to eqn. 4.29.

$$R = Ynew + 1.402Cr \tag{4.27}$$

$$G = Ynew - 0.344Cb - 0.714Cr \tag{4.28}$$

$$B = Ynew + 1.772Cb \tag{4.29}$$

The histogram equalised values are mapped to the original intensity values.

4.5.3 Retinex Algorithm

The idea of retinex was introduced in 1986 by Edward Land [149], to evaluate the inconsistency between the human and machine vision systems. The recognition of objects under various lighting conditions has an effect on colour perception. Separate inconsistency exists between the human vision and machine vision, as the former is adjustable with severe dissimilarity in illumination and latter includes range of view taken by the camera that is much less than the vision range of humans. It results in loss of essential features captured by a camera.

The analysis of retinex theory is based on human brain reactions to the colour perception by changing the illumination. A human-oriented image-processing procedure provides colour constancy and dynamic range compression. Spectral shifts have a strong influence on colour in the recorded images of the entire spectrum. Reliability in colour images to human observation requires dynamic range compression, colour constancy and colour and lightness rendition [150]. It requires wide dynamic range colour imaging systems.

The single scale retinex representation is shown by eqn. (4.30)

$$R_i\,(x,y) \;=\; log\, I_i\,(x,y) - log\,[I_i\,(x,y)\; * \;F(x,y)] \qquad (4.30)$$

where $i \in R,G,B$, *F(x, y)* is a Gaussian surround function.

The multi-scale retinex is a weighted summation of single scale retinex outputs and it is mathematically expressed as in eqn. 4.31.

$$RMSRi \ = \sum_{n=1} wn\ Rni$$

(4.31)

where n is the number of scale and wn is the different weights. The choice of scale is chosen empirically.

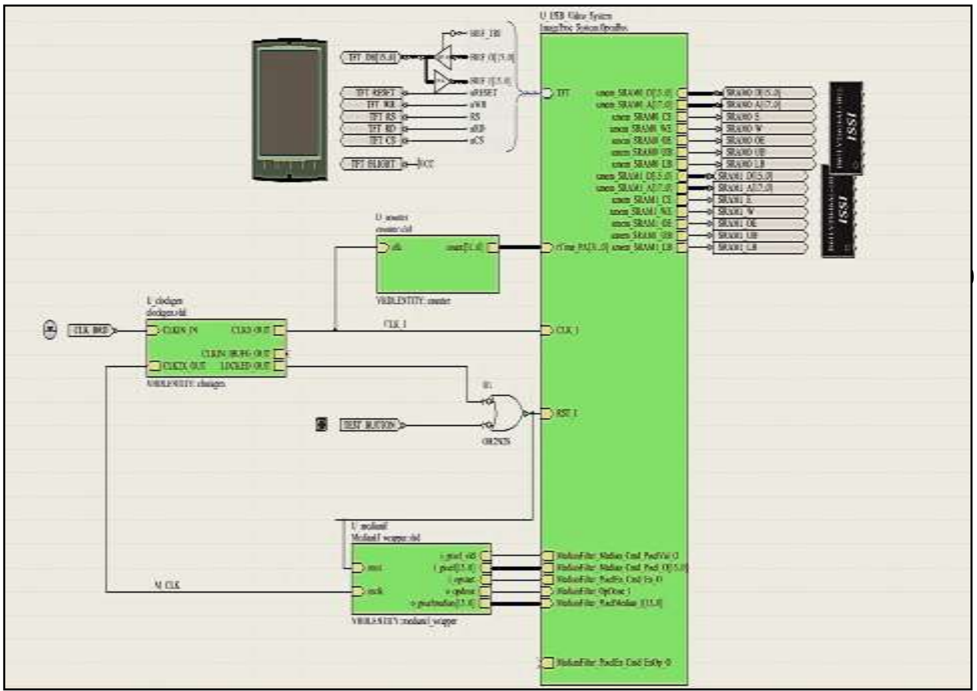
In this work, the input RGB image is converted to YC_bC_r model. Red colour needs to be enhanced more, compared to the other two. Hence, the component C_b is kept constant and the other two components, Y and C_r are enhanced by applying retinex algorithm and finally the enhanced components and the original C_b components are combined. Contrast stretching is also applied to this output to form the final output image.

4.6 Implementation in FPGA

Underwater image enhancement methods like homomorphic filtering, multi-scale retinex algorithm and AHE were successfully implemented in FPGA. The proposed method of fusion technique was also implemented. The device utilisation were computed. The implementation of these algorithms in FPGA are discussed in the following sub sections.

4.6.1 Multi-scale Fusion Method

The work was implemented in Xilinx Virtex 4 FPGA. Altium Designer was used as software development platform. Schematic document, which occurs in the top level, consists of schematic diagram that is a representation of the elements of the system using graphic symbols and



all these elements were wired together by means of buses, wires or signal harness units. A schematic document was added to the FPGA project and the open bus sheet symbol and interface circuitry are placed and interconnected. The schematic of the design is shown in fig. 4.3.

Fig. 4.3 Schematic diagram

In the schematic diagram, shown in fig. 4.3, VHDL entity block was used to indicate the input and output ports of the system. The output was displayed on the display unit. The main block was used to connect all peripheral devices that were used in the design.

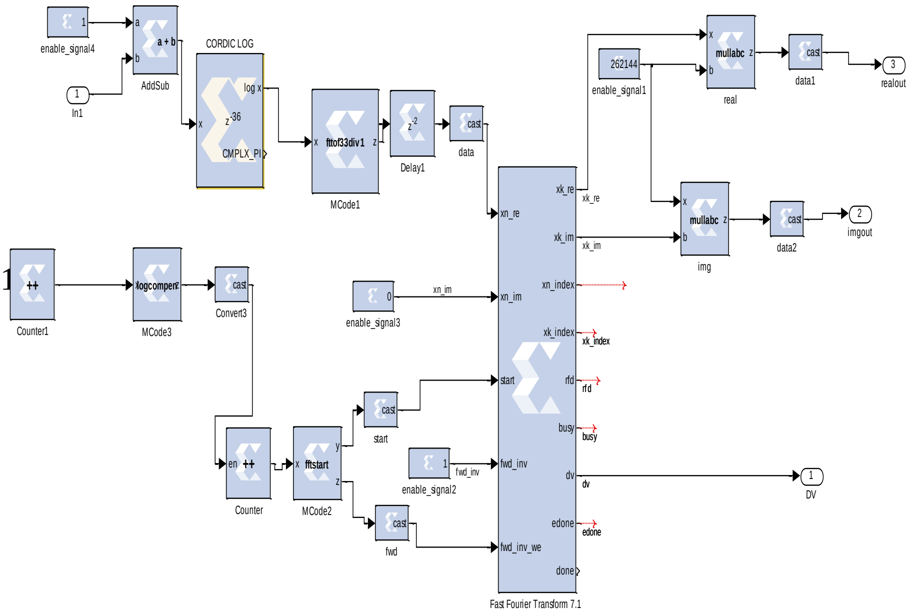
4.6.2 Homomorphic Filtering

The implementation of homomorphic filter, which is a nonlinear filter, on system generator platform was performed. Calculation of logarithmic and antilogarithmic value was done by means of eqn. 4.32 as direct blocks are not available,

$$log\ x = 1 + x^2/2 + \dots\dots$$

(4.32)

System generator allows one-dimensional (1D) data processing, so 2D FFT calculation was tiresome. Row wise and column wise FFT were calculated separately and combined to form 2D FFT.



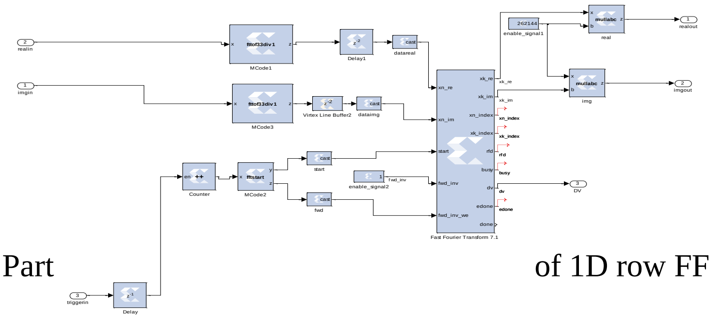


Fig. 4.4 Part of 1D row FFT model

Fig. 4.5 Taking column wise FFT

Fig. 4.4 and fig. 4.5 show various implementation blocks. Appropriate coefficients and multipliers were included in the block diagram to realise the algorithm. After the multiplication of the filter coefficients, row wise and column wise, its inverse Fast Fourier Transform (IFFT) was calculated. Finally, antilog value was calculated for obtaining the enhanced image.

4.6.3 Adaptive Histogram Equalisation

Xilinx system generator model for AHE is shown in fig. 4.6.

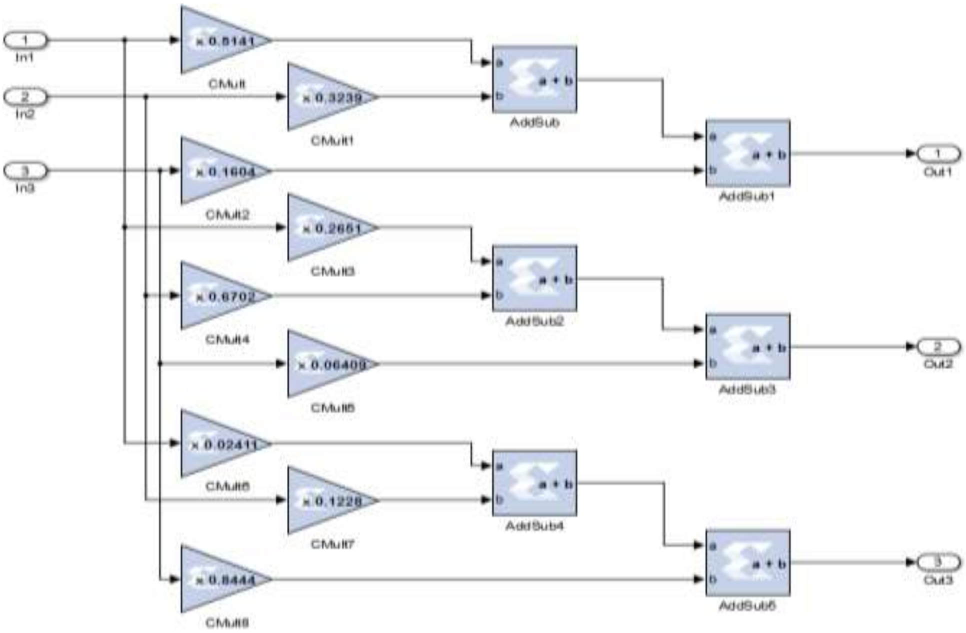


Fig. 4.6 Final block (part) of AHE method

The various system generator blocks were connected together to complete the design model. In mcode block, user defined functions for the proposed algorithm were written. As 2D image data was processed as a sequential 1D data, delay elements were given to access them in a later stage.

4.6.4 Multi-scale Retinex Algorithm

Fig. 4.7 shows part of the Xilinx system generator model file used to implement the algorithm in Virtex 4 FPGA development board.

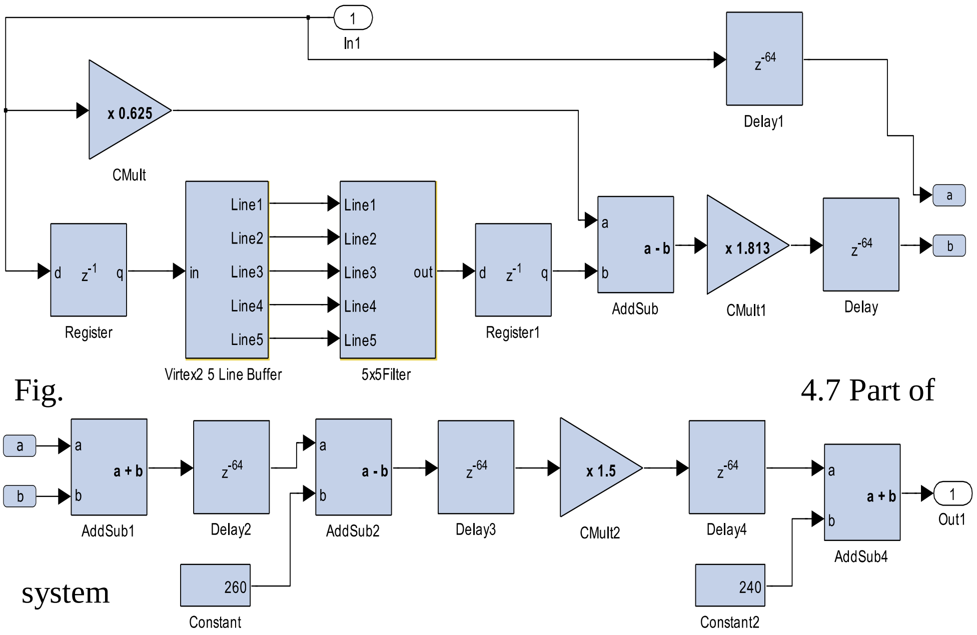


Fig. 4.7 Part of system generator model for multi-scale retinex algorithm.

4.7 Results and Discussions

Underwater image enhancement techniques like homomorphic filtering, AHE, MSR and proposed Multi-scale pixel based image fusion were implemented in FPGA and the results were compared using quality metrics. The results are shown in the following sub sections. It is clear from visual as well as qualitative analysis that the proposed model enhanced underwater degraded images more effectively. In the following sub section the results of each method are explained.

4.7.1 Homomorphic Filtering

Hardware co-simulation using Xilinx ISE design suite 11.3 system was carried out. The input images suffering from non-uniform illumination were enhanced by homomorphic filtering and are displayed in figures 4.9 and 4.10.

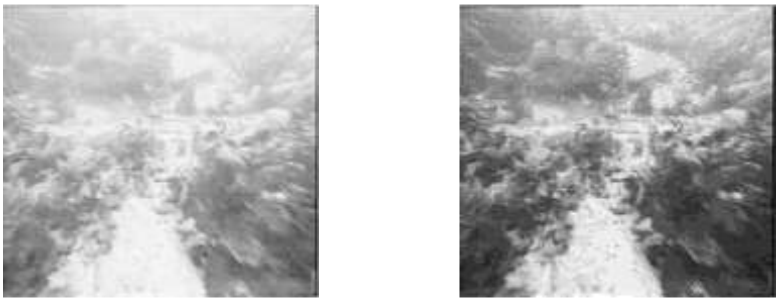


Fig. 4.8 Input image1 and output mage1 (Homomorphic filter)

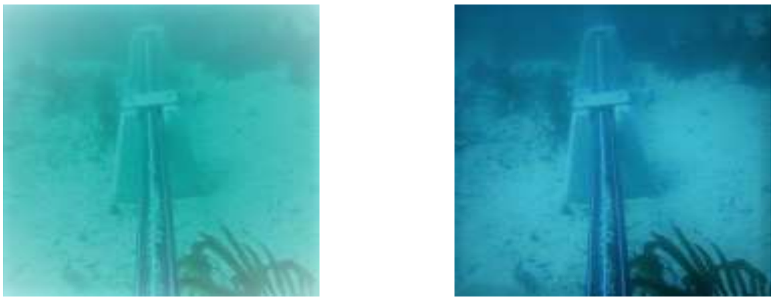


Fig. 4.9 Input image2 and output image2 (Homomorphic filter)

The device utilisation by FPGA is displayed in table 4.1. In this table, usage of various hardware blocks in Virtex 4, used by this design is shown. Only less than 30% of the resources of the Virtex 4 have been utilised for the design. Out of 35,840 slices present in Virtex4 XC4VLX80, only 4367 was used. Similarly, only 6943 LUTs, from the available 71,680 LUTs, were used. As the resources used are low, the power dissipation also reduces proportionally.

Table 4.1 Device utilisation – Homomorphic filter

Resources	Quantity
Slices	4367
FFs	4509
BRAMs	4
LUTs	6943
Mults/DSP48s	12

4.7.2 Adaptive Histogram Equalisation

Adaptive histogram equalisation was implemented in FPGA and the results are shown in fig. 4.10 and 4.11. The results show that visual quality of image is improved. The histogram of enhanced image shows a uniform distribution indicating that the brightness of enhanced image is uniformly distributed.

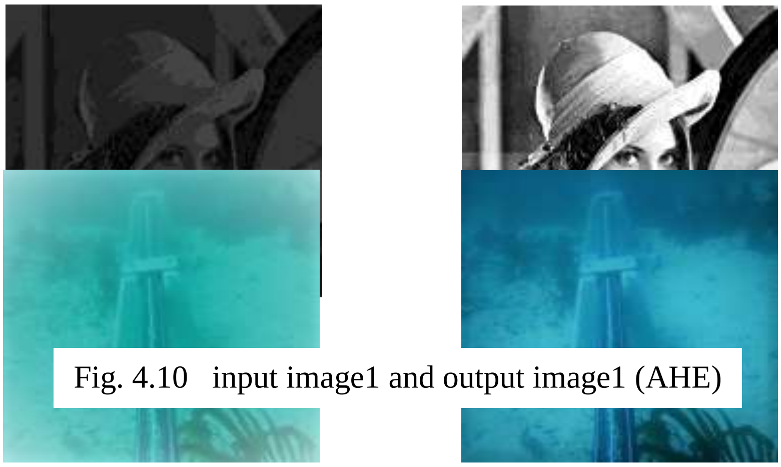


Fig. 4.10 input image1 and output image1 (AHE)

Fig. 4.11 Input image2 and output image2 (AHE)

Table 4.2

Device

Resources	Quantity
Slices	7234
FFs	6328
BRAMs	5
LUTs	10244
Mults/DSP48s	14

utilisation – AHE

The device utilisation when implemented the design in Virtex 4 FPGA is shown in table 4.2. The device Utilisation was below 30%.

4.7.3 Multi-Scale Retinex Algorithm

Hardware co-simulation was carried out using Xilinx system generator. The target platform was Virtex 4 FPGA board. Fig. 4.12 and 4.13 show two different images and their enhanced output.

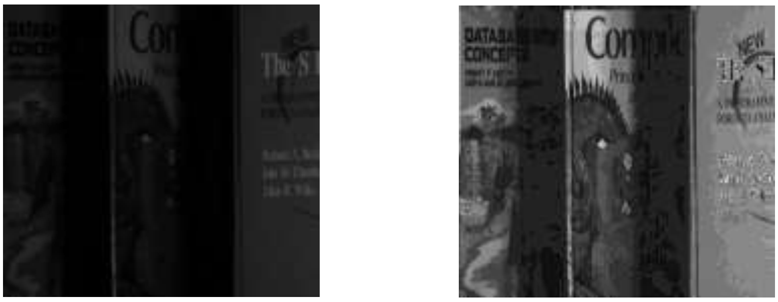


Fig. 4.12 Input image1 and output image1 (MSR)

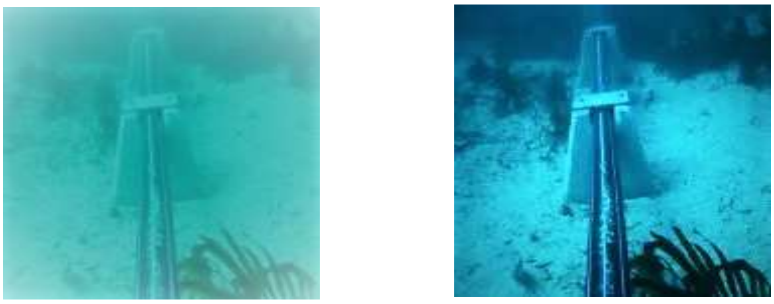


Fig. 4.13 Input image2 and output image2 (MSR)

Table 4.3 Device utilisation - MSR

Resources	Quantity
Slices	3224
FFs	3211
BRAMs	4
LUTs	3379
Mults/DSP48s	11

Table 4.3 shows the device utilisation of for the proposed multi-scale retinex algorithm. Less than 30% of the resources have been used when the design was implemented in Virtex 4 FPGA.

4.7.4 Multi-Scale Fusion

The proposed fusion design was implemented in Xilinx FPGA. The output images shown are free from any colour dominance. The processing time of each image was computed and the average time required for the test images was 37 ns. The results are shown in fig. 4.14 and fig. 4.15.

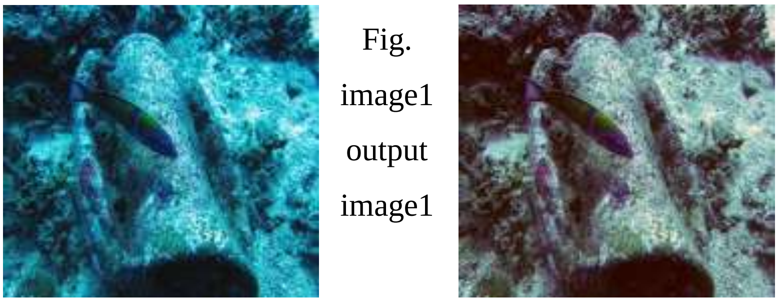


Fig. 4.14 Input image1 and output image1 (Fusion)

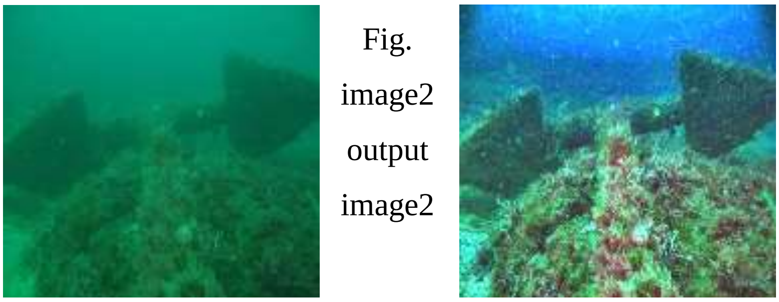


Fig. 4.15 Input image2 and output image2 (Fusion)

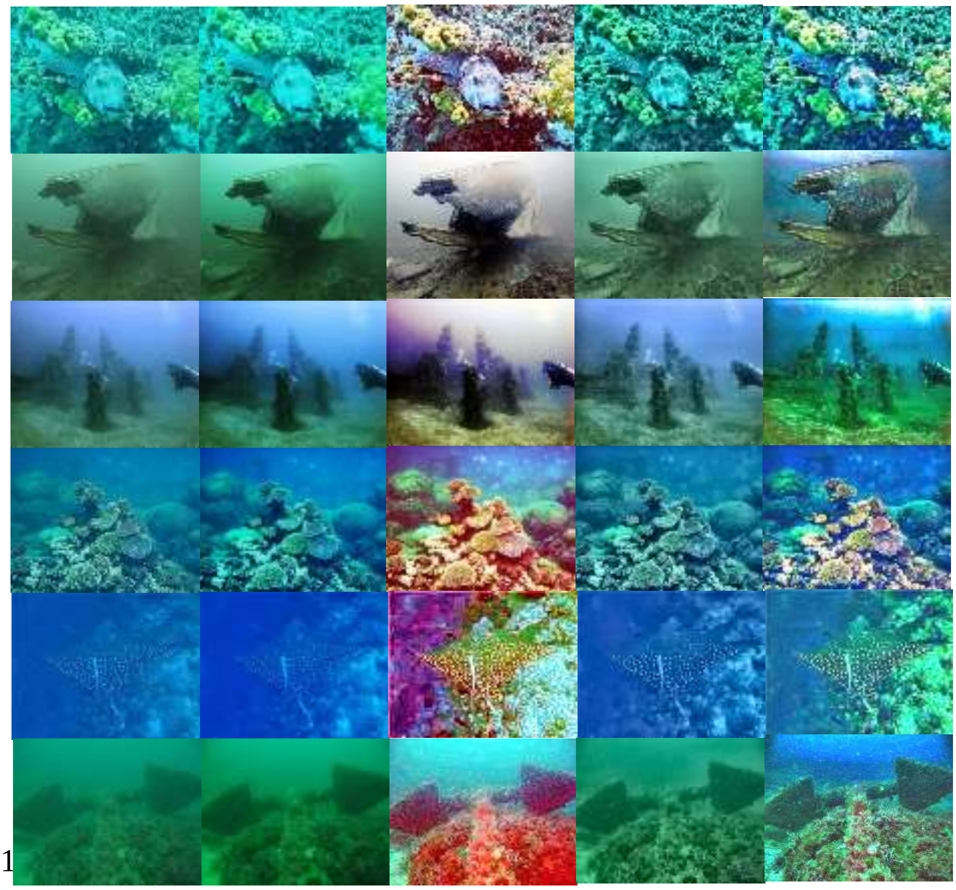
Table 4.4 describes device utilisation of the design when implemented in FPGA.

Table 4.4 Device Utilisation - Proposed Fusion Method

Resources	Quantity
Slices	6044
FFs	5296
BRAMs	6
LUTs	8467
Mults/DSP48s	14

4.7.5 Comparative analysis

The image comparison is done based on the visual quality of the original image and output images obtained using AHE, Retinex, Fusion and proposed Fusion method are shown in fig. 4.16(a-e). Homomorphic



filtering

(a) (b) (c) (d) (e)

Fig. 4.16 Visual comparison. (a) input image (b) AHE (c) retinex (d) fusion (e) proposed fusion method.

is not included as it improved only contrast of the given image. Fig. 4.16 © shows retinex output and here, red channel enhancement is more in some cases. Normal fusion method enhanced the image better than AHE, but the colour cast is still present. The proposed method shown in fig. 4.16(e) is able to enhance underwater image significantly and reduces colour cast much better than other methods.

An image signal whose quality is being evaluated can be thought of as a sum of an undistorted reference signal and an error signal. A widely adopted assumption is that the loss of perceptual quality is directly related to the visibility of the error signal. The quality metrics used for analysing different techniques are Root Mean Square Error (RMSE), Peak Signal to Noise Ratio (PSNR) and Mean Absolute Error (MAE). The MSE represents the cumulative squared error between the improved image and the original image, whereas PSNR represents a measure of the error. The enhanced image is good if it has low MSE and high PSNR values. Root Mean Square Error (RMSE) is the root of MSE. MSE is given by eqn. 4.33.

$$MSE = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[I_1(i,j)-I_2(i,j)]^2}{MN}$$

(4.33)

PSNR is given by eqn. 4.34.

$$PSNR = 10log_{10}\left(\frac{L^2}{MSE}\right) \quad (4.34)$$

where L is the number of gray levels.

Mean Absolute Error is given by eqn. 4.35.

$$MAE = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left|[I_1(i,j) - I_2(i,j)]\right|}{MN} \tag{4.35}$$

where I_1 and I_2 are the input and output images respectively. MxN is the size of the image.

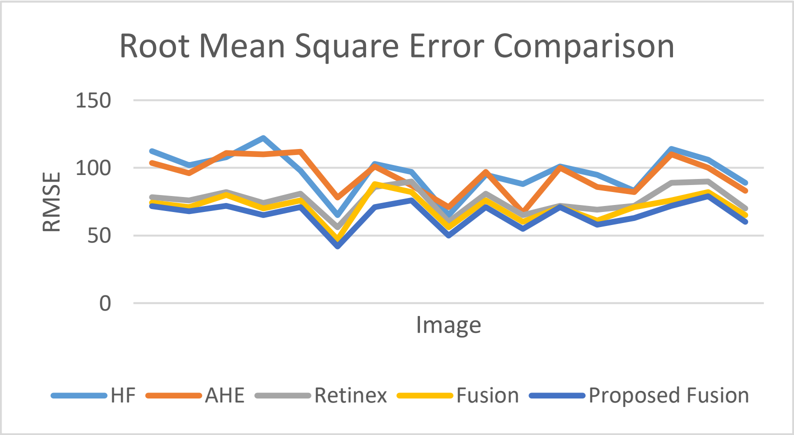


Fig. 4.17 Comparison of RMSE value for different image samples

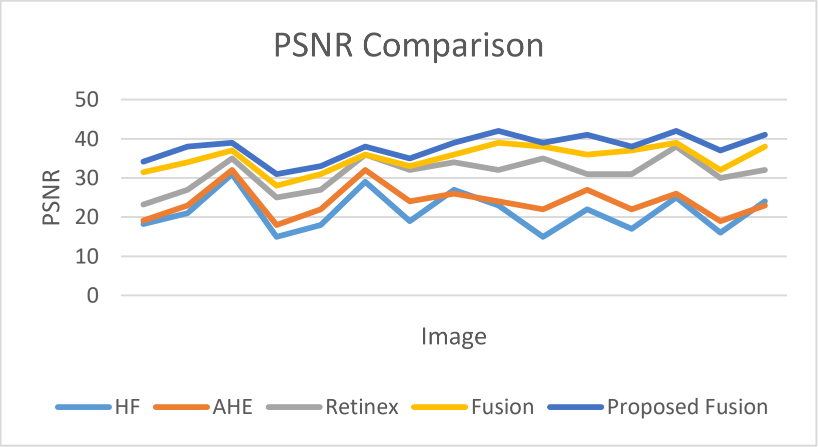


Fig. 4.18 Comparison of PSNR value for different image samples

Fig. 4.17 shows Root Mean Square Error value for image enhancement techniques like Homomorphic filtering (HF), AHE, Retinex algorithm, Fusion method and proposed Fusion method. Underwater image samples obtained from internet were used for comparative analysis. Fig. 4.18 shows PSNR comparison chart and fig. 4.19 shows MAE comparison. All these comparison charts indicated that proposed method is a better choice for enhancing underwater images. Table 4.5 shows comparative analysis of various image enhancement techniques on the basis of RMSE, PSNR, MAE, Power and Speed.

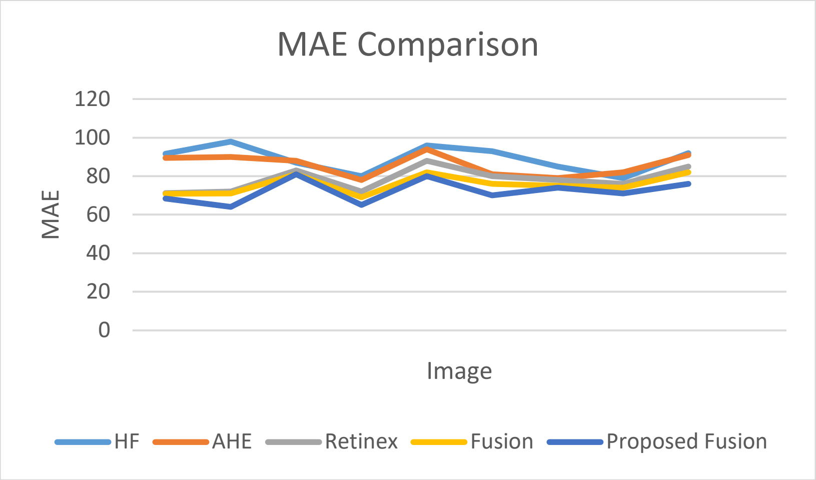


Fig. 4.19 Comparison of MAE value for different image samples

Quality Matrics	Homomorphic Filtering	AHE	Modified Multi-scale Retinex Algorithm	Classical Multi-scale Fusion from single image	Multi-scale Fusion ((Proposed)

RMSE	112.43	103.6	78.3	74.2	71.8
PSNR	18.2	19.1	23.2	31.5	34.2
MAE	91.7	89.4	71.2	70.9	68.4
Power dissipation in FPGA(in Watts)	0.237	0.215	0.214	0.226	0.205
Processing time in FPGA@100 MHz	42.18ns	34.21 ns	39.45 ns	39.40 ns	37.23 ns
Processing time in software@2.4 GHz	6.6 ms	4.9 ms	5.4 ms	5.3 ms	5.1 ms

Table 4.5 Comparative analysis

The comparative analysis presented in table 4.5, based on qualitative analysis, shows that the proposed multi-scale fusion based image enhancement outperformed other methods under the same conditions. Though the processing time is little bit higher than AHE, compared to other performance analysis, the proposed method is a better option. It has also shown that FPGA speed is comparatively higher, than executing the same algorithm in software.

4.8 Summary

A novel method of multi-scale fusion method using single image was implemented in FPGA for enhancing underwater images. It was compared with the image enhancement techniques like homomorphic filtering, adaptive histogram equalisation, multi-scale retinex algorithm and multi-scale fusion technique under the same conditions. Based on quality metrics such as RMSE, PSNR and MAE, it is found that the proposed method is a better option for enhancing underwater images that are suffered due to non-uniform illumination, absorption and scattering. The FPGA device utilisation for all these techniques were also computed and all the techniques used low percentage of available resources in the selected FPGA.

CHAPTER 5

ADAPTIVE GAUSSIAN MIXTURE MODEL BASED MOVING OBJECT DETECTION

In many computer vision applications like object tracking, human-machine interface, motion estimation etc. moving object detection is the initial and important step. The basis stage of operation for many underwater vision systems is object detection from the complex underwater environment. Many applications such as biomedical, security, surveillance and identification of threats in defence and navy make use of object detection [34]. By automating the detection system, the operations can become more efficient and accurate. Adaptive Gaussian mixture model based moving object detection was implemented in FPGA.

5.1 Introduction

The autonomous detection and tracking system could serve a crucial role in underwater surveillance. Marine researchers who involved in the underwater study can make use of the system if automation is provided for object detection and tracking. Underwater object detection is a very indispensable issue for many civilian and military applications such as hydro graphic survey used for secured navigation [151] [152].

Object detection requires typically a long time for processing and analysing by human experts. Real time object detection is very important especially when dealing with special purpose hydrographic survey operations such as dredging operations (for the purpose of deepening harbours, navigation lines and underwater mining) which may be cost-effective and time constraint operation that needs a precise and fast surveyor decision to determine the presence of objects or obstacles in underwater [153]. So the vision system should be operationally effective for better performance in real time applications. Traditionally, marine biologists determine the presence and exploration of different types of marine animals and objects using several object detection techniques.

5.2 Underwater Object Detection Techniques

Underwater video scenes have more than one of the following properties:

- Light changes: Real-time video acquisition requires object detection to be done irrespective of the lighting conditions. In fact, the video needs to be acquired for the entire day and the detection algorithm should work even when there is light transitions.
- Physical phenomena: During video acquisition, various physical phenomena affects image contrast. For instance, sea currents, storms or typhoons can easily vary the contrast and the clearness of the captured videos.
- Grades of freedom: In underwater videos, the moving objects can move in all three dimensions, while videos containing traffic images or pedestrians are virtually confined in two dimensions.
- Forming algae on camera lens: The contact of camera’s lens with seawater can lead to the quick formation of algae on top of the camera lens that will introduce level of uncertainty in captured images.
- Periodic and multimodal background: Arbitrarily moving objects (e.g. pebbles, small rocks) and periodically moving objects like tide and flood drift are common findings in the underwater setting [154].

Therefore, there should be a precise, flexible and fast underwater object detection system for use in a variety of underwater low-level images captured from underwater imaging devices operating under varying weather and bathymetric conditions. The developed system should be capable of tackling the huge complexity of the underwater environment and multidimensional motion features of the object. Analysis of underwater videos opens a new drift in biomedical and research applications. However, underwater video analysis can be used to determine the health of the ocean, endangered species, water quality, biodiversity, climatic change and impact of food chain, invasive species, ocean pollution, sustainability and biodiversity of the species surviving underwater [155]. The different algorithms used in this work for moving object detection are background subtraction method and adaptive Gaussian mixture model (GMM) method.

In background subtraction, a reference image is formed by averaging images over time at the time of initialization and is subtracted from the current image by pixel-pixel fashion, there by detecting the moving regions [156]. The difference between cumulative average of the previous values of pixel's intensity value and current pixel's intensity value is the main criteria for GMM [155].

5.3 Background Subtraction

Moving regions are detected using temporal difference method by taking pixel-by-pixel difference of consecutive frames (more than two) in a video sequence. Even though simple calculation is the advantage of frame difference method, it has a major disadvantage that it is sensitive to noise, which means that without moving the object if the background brightness is changed, the output of frame difference method will not be accurate. This problem can be overcome by using moving edge method, because the edge has no relation with the brightness. Using Sobel edge detector the edges of two continuous frames are detected and the difference between the detected edges are calculated. Then by comparing the non-zero pixels to a predefined threshold, it decides whether the region is in motion or not.

In this work, edge difference method for detecting moving object is used. Silven et al. [157] described a scheme for temporal differential method where a pixel at location *(x, y)* in the current image I_t is marked as foreground if it satisfies the eqn. 5.1 where *thr* is the threshold value.

$$|It(x,y) - Bt(x,y)| > thr \tag{5.1}$$

The background B_t is modified with the current pixel value. In this work, pixel value is taken after edge detection. The motion detection sensitivity depends on the value of the threshold. While selecting a threshold, it should be noted that a very low threshold cannot filter the noise in an image and too high threshold may result in the loss of useful changes in the image.

5.4 Proposed Adaptive GMM

The recursive approach of moving object detection system employing input video sequences having gradual error rectification is described by the block diagram shown in fig. 5.1. The system operates through different stages as in the block diagram to warrant efficient moving object detection and ultimate usage of gathered information. The underwater video sequence collected using a camera are fed to the system. Oceanographers and underwater divers now a day depending on the high-resolution images captured with cameras specially designed to operate in underwater environment. This visual data plays a key role in their work and studies. The proposed system can be incorporated with such visual information collection system so that the cost and maintenance requirement of the object detection system can be reduced.

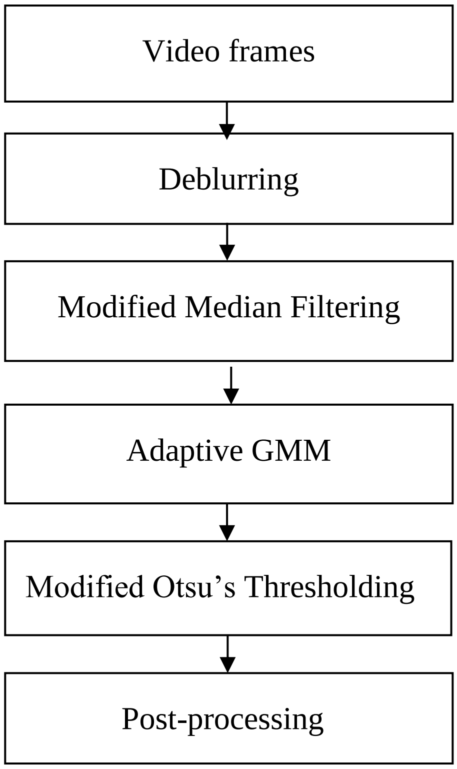


Fig. 5.1 Proposed adaptive GMM model

These video frames are deblurred to improve the sharpness of the edges. The background model and the related normalisation parameters are extracted from the frames. The background model key parameters get updated recursively according to the available new information from each pixels in the frame. In the moving object detection stage, a threshold based scheme separates the foreground from the background distributions efficaciously employing the parameters already derived in the previous stage of the system. The further post-processing stage assures the required level of visual quality of the result by thresholding and connected component analysis techniques.

5.4.1 Deblurring

Image deblurring sharpens the edges of pre-processed underwater image. The image degradation is represented by,

$$y = Sx + n$$

(5.2)

where x, y and n are the column vectors representing the original image, degraded image and additive noise. S is a square matrix used as the linear blurring operator [158]. Minimisation of the objective function is used to get the estimate of x and y,

$$C(x, S) = \|y - Sx\|_2^2 + \lambda R(F(x))$$

(5.3)

In eqn. 5.3, λ is the regularisation parameter and *R (F(x))* is the regularising term. The proposed algorithm for recovering deblurred image is given below,

- Step1: Set *S* as the identity operator.
- Step 2: Set x = *y*.
- Step 3: Set initial value to λ and prior's sparsity.
- Step 4: Find estimate of x: $x = \text{argmin}_x\, C(x, S)$ (*S* fixed).
- Step 5: Find estimate of S: $x = \text{argmin}_H\, C(x, S)$ (*x* fixed).
- Step 6: Give next value to λ and prior's sparsity.
- Step 7: Repeat step 4 to 7, if $\lambda \geq \lambda_{min}$

5.4.2 Median Filtering

The impulse-based noise can be effectively removed from images that are corrupted by distortions introduced by the medium using a median filter. It smoothens the image by replacing the pixel value with the median of the neighbouring pixel values. For that, all the pixels in the neighbourhood of the original pixel are sorted in ascending/descending order and then the middle pixel value or the median of the sorted value is used to replace the original pixel. When compared to other filters like simple mean, geometric and harmonic filters, the performance attributes of the median filter is significantly better. The median filter is a canonical image processing operation, best known for its salt and pepper noise removal abilities [114]. The mathematical expression for median filter is given in eqn. 5.4.

$$f(x,y) \ = \ median \ (g(s,t)) \tag{5.4}$$

where $(s,t) \epsilon \ S_{xy}, \ S_{xy}$ is the selected neighbourhood

A two-dimensional window of appropriate size (here 3x3) is selected and it is centred on the processed pixel *p(x, y)* in the degraded image. The pixels in the selected window are sorted in ascending/descending order to find the median pixel *Pmed*. The centre pixel of the 3x3 window is replaced with *Pmed*.

5.4.3 Adaptive Median Filter

The general output of the median filter is a specific processed pixel value placed at the location specified by the centre of the window S_{xy} under consideration. The variability of the window size is activated if certain condition levels are strictly satisfied. Level A and B are the two condition levels which should be analysed. The operation of the filter is shown in fig. 5.2. The algorithm is explained below.

- Level A:

$A1 = Zmed - Zmin$

$A2 = \mathrm{Zmed} - \mathrm{Zmax}$

$If\ A1 > 0\ and\ A2 < 0, Go\ to\ level\ B$

$else\ increase\ the\ window\ size\ to\ Sxy \leq max$

$repeat\ level\ A$

- Level B:

$B1 = Zxy.Zmin$

$B2 = Zxy.Zmin$

$If\ B1 > and\ B2 < 0, output\ is\ Zxy$

$else\ output\ is\ Zmed.$

Here Z_{min} is the minimum grey level value in S_{xy}, Z_{max} is the maximum grey level value in S_{xy}, Z_{med} is the median of grey levels in S_{xy}, Z_{xy} is the grey level at a specific coordinates and S_{max} is the maximum allowed size of S_{xy} [159]. The adaptive median filter having variable window size implements required level of smoothening for non-impulsive noises and prevents enormous distortions of data. The adaptive median filter preserves detail and smooths non-impulsive noise. It can assure better result even when the impulse noise is greater than 0.2.

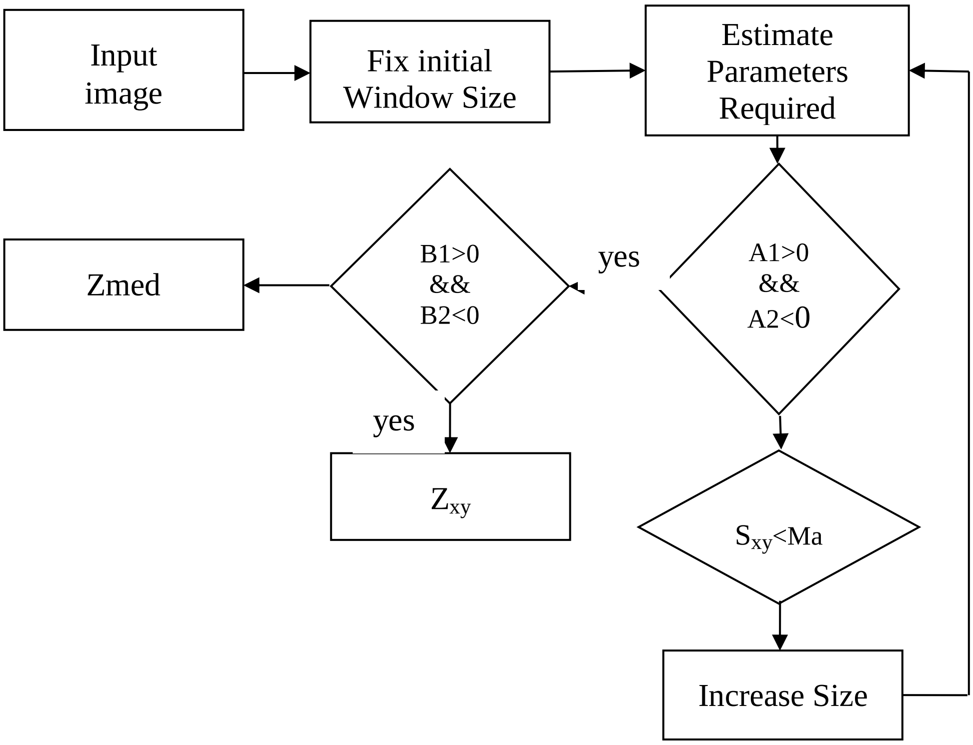


Fig. 5.2 Operation of Adaptive median filter

5.5 Background Estimation

In static environment, usually simple frame differencing approach is employed because there is no need to tackle the changing features of the background along with that of the moving objects. In uncontrolled environments such as underwater or oceanic surroundings, the simple background subtraction may fail, but this method has the benefit of very high computational speed.

The major problems involved in oceanic surroundings are varying illumination levels, temporal background clutter, occlusions, erratic movements of the objects etc. These problems are usually addressed by making the background model adaptive and flexible so that its parameters can track the dynamic complex medium [160].

Popular adaptive modelling for the background in an uncontrolled environment is GMM. To model a multimodal background image sequence, GMM uses a mixture of normal distributions. Instead of using a single distribution for all pixels in the background, multiple numbers of distributions are used in this approach. For each pixel, all the normal distributions in its background mixture relate to the probability of detecting a specific intensity or colour in the pixel. The algorithm mainly depends on the following assumptions. The background is much frequently visible when compared to foregrounds. Second, background must have least variance parameters. Therefore, the algorithm efficiently work with scenes where background clutter is formed by at most two surfaces appearing in the view of the pixel. A mixture of different Gaussian probability density functions with its own mean, standard deviation and weight constitutes a GMM [161]. The weights can be interpreted by the corresponding Gaussian model of the frequency.

The Gaussian mixture based background elimination system is defined by two parameters called learning rate α and threshold value T. The values of the pixel can be considered as a time varying process called pixel process and is a vector quantity for the images. The information regarding the history of all pixels at a particular time is available [122]. This history can be represented as in eqn. 5.5.

$$\{X_1, X_2, \ldots X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\} \tag{5.5}$$

where I denotes an image sequence. The recent history of each pixel, (X_1, X_2....X_t), is modelled by a mixture of *K* Gaussian distributions. The probability of detecting the present pixel value is given by eqn. 5.6.

$$P(X_t) = \sum_{i=1}^{k} w_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{5.6}$$

where *K* is the number of distributions, *ω* is an estimate of the weight of the ith Gaussian in the mixture at time *t*, $\mu_{i,t}$ is the mean value of the ith Gaussian in the mixture at time *t*, $\sum_{i,t}$ is the covariance matrix of the ith Gaussian in the mixture at time t, and η is a Gaussian probability density function. η is given as

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-1(X_t - \mu_t)^T \Sigma^{-1} (X_t - \mu_t)} \tag{5.7}$$

In order to reduce computational complexity, the covariance matrix is considered as a diagonal one since blue, red and green values are independent and with same variance. Hence, eqn. 5.8 should be satisfied.

$$\Sigma_{k,t} = \sigma_k^2 I \tag{5.8}$$

The developed model should be capable to handle addition or deletion of the stationary objects and the time varying illumination changes in the scene. In determining the Gaussian parameter, observations that are more recent are estimated as more significant. One of the major distributions of the mixture model is used to represent a new pixel value and then the model is updated based on this representation. The new pixel value, *Xt* is examined against the prevailing *K* Gaussian distributions in the mixture that model the background, until a match is found. Whenever the value is within 2.5 standard deviations of a distribution, a pixel is said to be matched. If the current pixel value do not match with any of the *k* distributions, then updating takes place in such a way that the minimum probable distribution is substituted with a distribution with the current value as its mean value, an initially high variance, and low prior weight. The procedure go on until the entire pixel is considered and the corresponding distribution parameters are updated accordingly.

5.6 Model Update Procedure

The Gaussian mixture model is defined by its key features mean, co- variance matrices and corresponding weights of mixed component densities. The changes arise in the key attributes, according to the motion characteristics of the pixels in the input visual information. This leads to the proper identification of the component distributions contributing the background features efficiently. So remodelling of the initially derived background model with introductory parameter values will be updated as per the information collected from the current pixel under consideration. The expectation maximisation algorithm is the preferred method to make the distribution parameters updated so that the log likelihood function is maximised.

5.6.1 Expectation Maximisation Algorithm

The GMM is a finite mixture probability distribution model. The models enable statistical modelling of environments with multimodal behaviour where simple parametric models fail to represent the characteristics of the data adequately. Expectation Maximisation (EM) algorithm is a standard approach for estimating GMM parameters. It is a repetitious parameter evaluation approach using available and estimated version of the lost information. The algorithm starts with an initial model to derive an updated one, which fit the collected data as per the condition given in eqn. 5.9.

$$p(x/\hat{\lambda}) > p(x/\lambda) \tag{5.9}$$

The derived model using the EM algorithm is considered as the primary model for the subsequent iteration. The entire procedure is repeated continuously until the required level of convergence is achieved. The two steps in EM algorithm are expectation step and maximisation step. These steps are iteratively performed until an acceptable convergence threshold is reached. Then an optimum model, fitting the input visual information, is derived.

In expectation stage of the algorithm, the obtained data set *X* is considered as incomplete and the complete one is represented as Y = (X, Z). The whole data set will be obtained by reckoning the parameters Z = Z_1, Z_2,….. , Zk. The log likelihood of the whole data Y is as shown in eqn. 5.10.

$$logp\left(\frac{Y}{\Theta}\right) = \sum_{i=1}^{N}\sum_{m=1}^{K} z^i \, log[w_m p(x^i/\theta_m)] \tag{5.10}$$

where z^i is given as

$$z^i = \log\left(m/x^i, \Theta^i\right) = \frac{w_m^t p(x^i/\theta_m^i)}{\sum_{i=1}^{K} w_i^t p(x^i/\theta_i^t)} \tag{5.11}$$

Here, Z^i is the posterior probability and Θ^t is the parameter estimate obtained after t iterations.

In the maximisation stage of the approach, the attributes of the mixed component Gaussian distributions are evaluated by iteration method. In this step, the parameters Θ^{t+1} are determined according to the estimate of the variables z_m^i. For Gaussian mixture models the corresponding parameters like mean, variance and the Gaussian weights are estimated according to the respective equations derived, by maximising the log likelihood function.

The convergence condition is called cost function. The expectation and maximisation step is repeated until the outcomes satisfy the cost function. Usually when the value is in the range of 10^5, the cost function is expressed as a threshold. After the iteration is completed, the moving object can be detected by estimating the background of the moving region. The implementation of the updating process can be achieved by the following iteration stages that is derived from the EM algorithm.

$$w_{k,t} = (1-\alpha)w_{k,t-1} + \alpha(M_{k,t}) \tag{5.12}$$

where α is the learning rate, $M_{k,t}$ is zero for unmatched models and one for matched models. The speed, at which the distribution parameter changes, is inversely proportional to the learning rate. The weight parameter $M_{k,t}$ is the averaged posterior probability that pixel values have matched model *k*, given the observations from time duration one to *t*. The key parameters μ and σ remains as such for unmatched distributions and for matched distributions it is updated according to the relation

$$\mu_t = \ (1-\rho)\mu_{t-1} + \rho X_t \tag{5.13}$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu) \tag{5.14}$$

where ρ is given as

$$\rho = \ \alpha\eta(X_t/\mu_k\,,\sigma_k)$$

(5.15)

The parameter ρ is a type of causal low-pass filter as $M_{k,t}$, except that only the data that matches the model is incorporated in the estimation.

The noticeable advantage of this process is that when a new pixel is added to the background, it does not spoil the existing model of the background. By modifying the already existing model, it is possible to get the new model. The original background colour persists in the mixture until it turns into the K^{th} most probable one and a new colour is detected. If an object is static for a long time (exceeding the limit), the object becomes part of the background. When it starts moving again, the distributions with lower weights will be instantly re-integrated into the background, but the previous background model still exists with the same μ and σ.

5.6.2 Spatial Coherence based Update

One of the main drawbacks of conventional GMM model is the limited usage and influence of spatial coherence details that exist in input frame sequences. Spatial coherence can reveal hidden relevant information in input images and is helpful to make the learning process of the Gaussian mixture model faster and efficient when compared to the conventional approaches. In the proposed method the spatial redundancy information is utilised in the updating stage if there exists any distributions matched to the current pixel attributes. In updating the distribution parameters of the components in the model, instead of current pixel based method, there exists neighbourhood pixel based updating processes with different learning rate over the neighbourhood. The varying learning rate over the neighbourhood is given as below.

$$\alpha = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

(5.16)

To separate background from foreground pixels, it is required to detect the Gaussian of the mixtures that are probably generated by background processes. When the scene includes only static objects, the variance of the distributions, which represents the background, will be very small. The new object that occludes the background will not be matched to the existing distribution easily. New objects may introduce new distributions or increase the variance of the existing distributions.

To identify the distributions which stand for the background, the parameter $\frac{\omega}{\sigma}$ has to be estimated. This parameter value increases when the σ value decreases. As variance decreases the new derived parameter magnitude falls and the distributions with highest parameter value can be considered as the most probable background distribution. So new distributions will be gradually replaced with the distributions with lower values. Then the highest B distributions are taken to represent the background model as in eqn. 5.17.

$$B = \ argmin_b(\sum_{k=1}^{b} w_k > T) \qquad (5.17)$$

where the quantity T shows the least portion of the data that should be accounted for the background. A multimodal distribution caused by a recurring background motion can be covered if the value of T is higher. The method explained here allows to extract the foreground pixels from the background pixels and to classify foreground pixels in each new frame while updating the description of each pixel's process [162].

5.7 Adaptive Threshold for Detection.

The efficacy of the GMM based underwater moving object detection technique thoroughly depends on the threshold value over which the background distributions are isolated. The conventional GMM model segregates the distribution corresponding to the background and foreground pixels adopting a static passive threshold value which is predefined. The application for which the system is designed is a real time one demanding extreme operational efficiency while processing the input. The adaptive threshold that accounts for the characteristics of the moving object has to achieve the required level of accuracy and efficiency to the system.

According to this method the input difference frame is divided into *K* numbers of equal sized blocks and they are classified as either having region of change (ROC) or not having any change [163]. Let the input frame F is divided into *K* equal blocks such that

$$F = \sum_{n=1}^{K} W_n$$

(5.18)

The pixel values in ROC will be high due to severe illumination changes and multimodal features and having low value in the background region. The regions of significant changes are obtained using first-moment m of the histogram using the given equation.

$$m = \sum_{i=1}^{V_{max}} i.H_i \qquad (5.19)$$

where V_{max} stands for the maximum pixel value in the input frame and Hi implies the frequency of the grey level $n = 1, 2, ..., V_{max}$ in the image. If the first moment exceeds the computed threshold then the image block is acknowledged as the ROC, otherwise as the fixed background region.

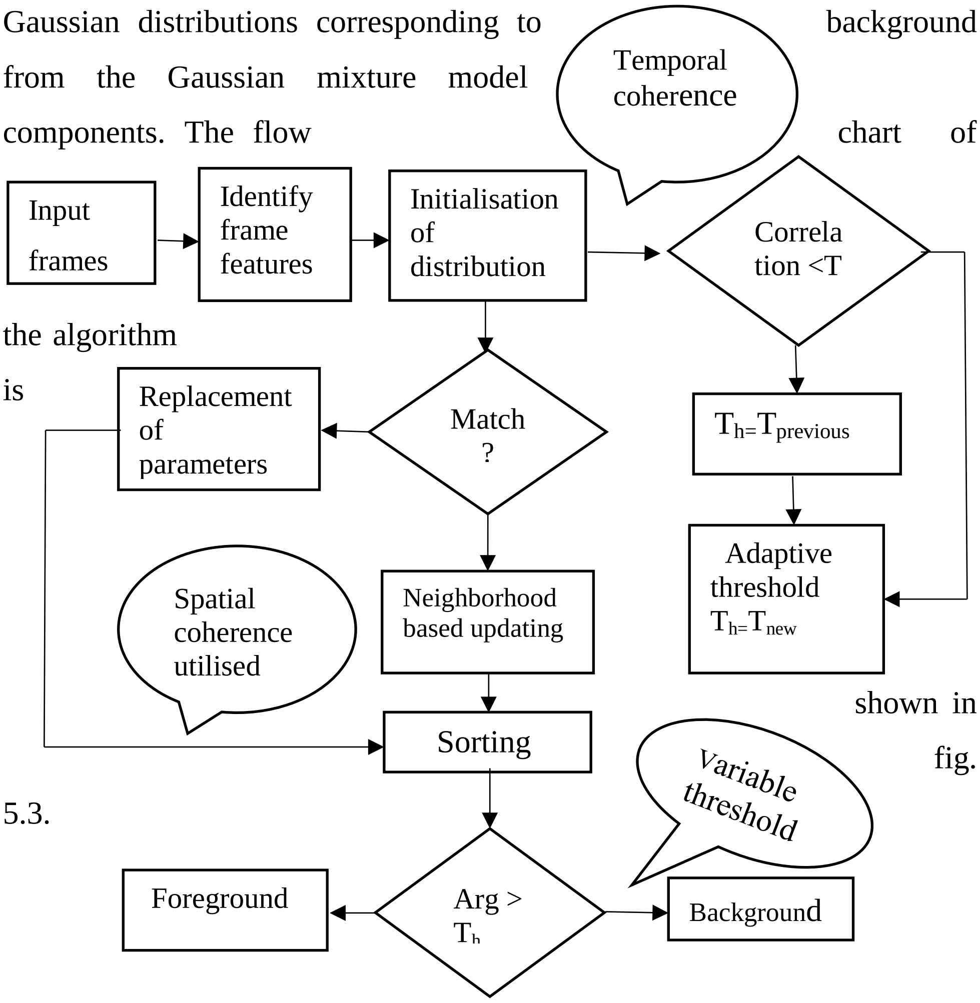
$$W_n = \begin{cases} W_n^b, & if\ m \leq T \\ W_n^r, & if\ m > T \end{cases}$$

(5.20)

The threshold value that recognises the image blocks having a region of change is obtained as described below. The first moments of all image blocks are plotted and make a line joining the moments of the first and last image blocks. The threshold T that isolates a required region of change of the image blocks is the moment having maximum perpendicular distance from the line already drawn. The corresponding two-dimensional variances of the image blocks having high motion characteristics regarding ROC are estimated using the standard relations as given below in eqn. 5.21.

$$Var(x) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{i=1}^{n}\frac{1}{2}(x_i - x_j)^2 \tag{5.21}$$

The reciprocal of the average value of the variances estimated, corresponding to the highest motion featured image block, is chosen as the new threshold value to isolate the distribution components representing background from that of the foreground. Since the estimated threshold highly depends on the pixel values of the input further, thresholding approach is advised in the proceeding stages to incorporate quality improvement. The new estimated threshold value is used to separate the Gaussian distributions corresponding to background from the Gaussian mixture model components. The flow chart of the algorithm is shown in fig. 5.3.



yes

Fig. 5.3 Flow chart of GMM Algorithm

5.7.1 Otsu's Method of Thresholding

Otsu’s method of thresholding is an efficient approach that makes clusters tight by maximising the variance between the classes by introducing an optimal threshold value. The threshold is estimated in such a way that the overlap between the clusters is minimised by spreading one of the distributions more while reducing the other one [164]. The assumptions made in this threshold method of segmentation are explained below.

- Image and the histogram are bimodal in characteristics.
- Neither the use of spatial coherence nor any other notion of object structure.
- Assumes stationary statistics, but it can be reformed to be locally adaptive.
- Since the bimodal brightness behaviour evolves only from object appearance differences, the method assumes uniform illumination.

Let *{0, 1,..., L-1}* denote L distinct intensity levels in a digital image and n_i indicate a number of pixels with intensity value *i*. Then the total probability is given as

$$\sum p_i \ = \frac{n_i}{MN}\sum_{i=0}^{L-1} \ = 1$$

(5.22)

where *MN* is the total number of pixels in the image. Here *k* is the threshold chosen to separate pixels in the image into two classes. The two class probabilities are given by the following relationship.

$$p_1(k) = \sum_{i=0}^{k} p_i$$

(5.23)

$$p_2(k) = \sum_{i=0}^{L-1} = 1 - p_1(k)$$

(5.24)

The mean intensity values of the pixels assigned to the classes are

$$m_1(k) = \frac{1}{p_1(k)}\sum_{i=0}^{k} i.p_i$$

(5.25)

$$m_2(k) = \frac{1}{p_2(k)}\sum_{i=k+1}^{L-1} i.p_i$$

(5.26)

The global mean value obtained from the class means is given by the equation,

$$m_G \ = \ p_1 m_1 + \ p_2 m_2 \qquad (5.27)$$

The between class variance which is to be maximised is given by the equation as in eqn. 5.28.

$$\sigma_B^2 = p_1(m_1 - m_G)^2 + p_2(m_2 - m_G)^2$$

(5.28)

The optimum threshold value $k*$ can be obtained by maximising σ_B^2 large as possible, as in eqn. 5.29.

$$\sigma_B^2\,(k^*) = \max_{0\leq k\leq L-1} \sigma_B^2(k)$$

$$g(x,y) = \begin{cases} 1, & if\ f(x,y) > k^* \\ 0, & if\ f(x,y) < k^* \end{cases}$$

(5.29)

The separability measurement is given by η, where the value of η is given by,

$$\eta = \frac{\sigma_B^2}{\sigma_G^2}$$

(5.30)

5.7.1.1 Modified Otsu's Thresholding

Here median based Otsu's thresholding is chosen instead of normal mean based Otsu's thresholding. Median based Otsu's thresholding is used for skewed or heavy-tailed image as the mean value gives a very robust estimate compared to the grey level value. Let m_T be the total of the grey level image with L grey levels, m_0 and m_1 are the corresponding medians of foreground and background part, m_{med} is the global median value, ω_0 and ω_1 are the corresponding probabilities of foreground and background part. Then the between-class variance is given by,

$$\sigma_B^2 = \omega_0(m_0 - m_{med})^2 + \omega_1(m_1 - m_{med})^2$$

(5.31) and the corresponding threshold is,

$$t^* = argmax(\sigma_B^2) \quad for\ 1 \leq t \leq L$$

(5.32)

5.7.2 Post Processing

In post-processing, the resultant detected frames are modified by using the two pass connected component algorithm. The foreground pixels obtained are segmented into regions by this algorithm. The post-processing step is an essential one, for the refinement of boundaries.

Connected component analysis is an iterative algorithm to distillate the connected and similar region from the input binary image and to provide with proper label to each discrete regions. So it is considered as an appropriate stage after the thresholding stage of the system.

5.8 Implementation in FPGA

The proposed GMM based algorithm and subtraction method were implemented in Xilinx Virtex 4 FPGA. The results as well as device utilisation for both algorithms are discussed in the following sections. The GMM based method is computationally complex compared to the conventional background subtraction method. To implement GMM based design using system generator is tedious. Hence, Altium designer was used.

5.8.1 Back Ground Subtraction Method

Using a 5x5 Sobel mask, edge detection was carried out and was properly delayed to subtract it from the current frame pixel. Matlab function block (mcode) of system generator was programmed to separate pixel values that are beyond a threshold value.

The above process occurred simultaneously in the R, G and B colour channels. The object in motion appeared as white and the background as black. The output of three channels, i.e., R, G and B were then combined together to obtain a colour output image that detected the target. The movement were tracked by comparing the pixel values of this video to the original video. The regions where they differed can be tracked and bound. The noise in the video frames were removed using a median filter.

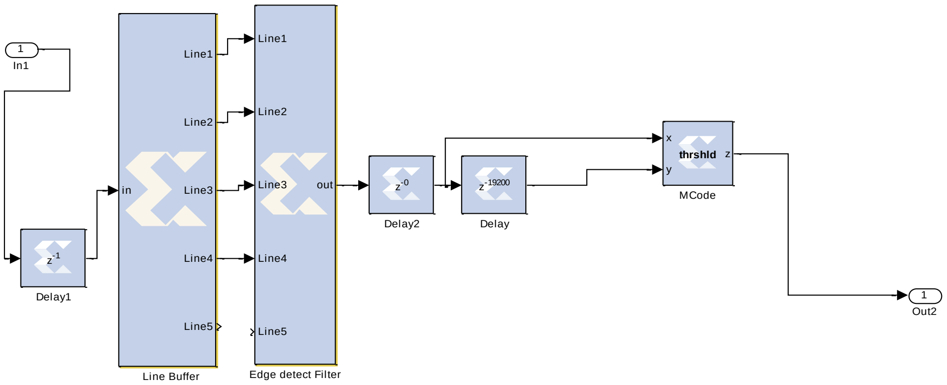


Fig. 5.4 Part of edge detection and threshold model

Threshold setting and comparison were carried out using mcode block of system generator. Fig. 5.4 shows implementation part of the algorithm in system generator. Here five inputs are processed in parallel since a 5 x 5 filter is used.

5.8.2 Proposed Adaptive GMM

The design was implemented in Virtex 4 FPGA. Using Altium designer, schematic sheet and open bus document were created for implementing the design in Xilinx Virtex 4 FPGA. The video was stored on the external RAM available in the development board. C to hardware compiler is used for the implementation. From the schematic sheet, the underlying C source file is referenced in much the same way as a HDL file. Instead of a sheet symbol, a C code Symbol primitive is used. Each C code symbol represents one top-level exported function, resident in the referenced C source file. Access to the parameters of the function is made using C code entries – placed on the symbol and functionally similar to sheet entries on a sheet symbol. These entries can be wired to other components on the schematic, allowing for transfer of data.

5.9 Results and Discussions

Object detection of moving objects from underwater video scene were done using conventional background subtraction method and proposed adaptive GMM method. The comparison is made between different methods regarding quality matrices like sensitivity, precision and accuracy. Device utilisation of FPGA is also noted. As the enhanced output after multi-scale fusion was directly given as the input of AGMM, device utilisation shown here is applicable only for the AGMM algorithm.

5.9.1 Background Subtraction Method

The two input video frames are input video farme1 and frame2 and output are video frame1 and frame2. They are shown in fig. 5.5 and in fig. 5.6. Threshold value plays an important role in detecting a moving object.

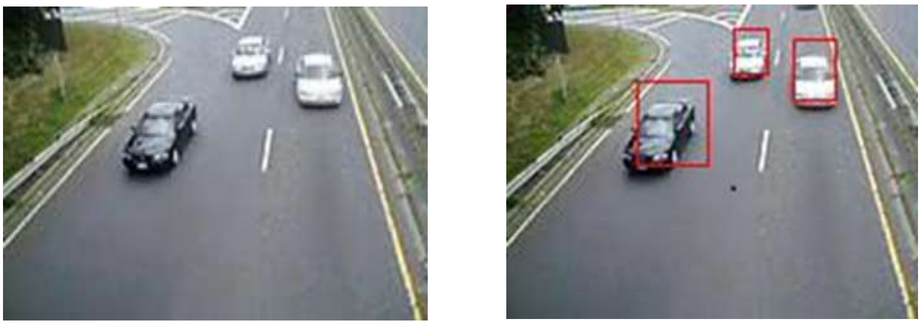


Fig. 5.5 Input video frame1 and output video frame1

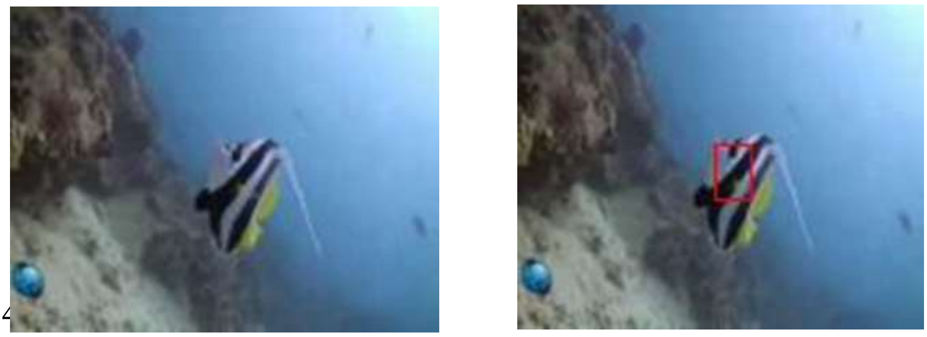


Fig. 5.6 Input video frame2 and output video frame 2

Table 5.1 Device utilisation - Background Subtraction

Resources	**Number**
Slices	14685
Flip-flop’s	28818
BRAMs	27
Look up Tables (LUT)	28347
IOBs	56
DSP48s	15

The device utilisation is shown in table 5.1. Out of 35,840 slices present in Virtex4 XC4VLX80, only 14685 was used. Similarly, only 28347 LUTs, from the available 71,680 LUTs, were used.

5.9.2 Proposed Method based on Adaptive GMM

The efficacy of the proposed underwater moving object detection method was analysed using different groups of video collected and made a comparison with the conventional approach with validated information and proved the ability of the system to tackle and separate the underwater moving objects in terms of sensitivity, precision and accuracy.

Sensitivity or True Positive Rate (TPR) is equivalent to the hit rate and is given by,

$$\text{Sensitivity} = TP/P$$

Similarly, Precision or Positive Predictive Value (PPV) and Accuracy (ACC) is given by,

$$\mathrm{Precision} = \mathrm{TP} \,/\, (\mathrm{TP} + \mathrm{FP})$$

$$\mathrm{Accuracy} = (\mathrm{TP} + \mathrm{TN}) \,/\, (\mathrm{P} + \mathrm{N}),$$

where, $\mathrm{P} = \mathrm{FN} + \mathrm{TP}$ and $\mathrm{N} = \mathrm{FP} + \mathrm{TN}$ (5.33) where, *TP* denotes the true positive, *TN* denotes the true negative, *FP* denotes the false positive and *FN* denotes false negative.

In the first video frames shown in fig. 5.7, multiple fishes in an aquarium are displayed. Even if the fishes are moving at fractions of a second, the proposed algorithm can track the moving fishes accurately than the adaptive GMM. The second video frames, shown in fig. 5.8, displays an underwater moving starfish. Table 5.2 shows device utilisation of the proposed design when implemented in Virtex 4 FPGA.

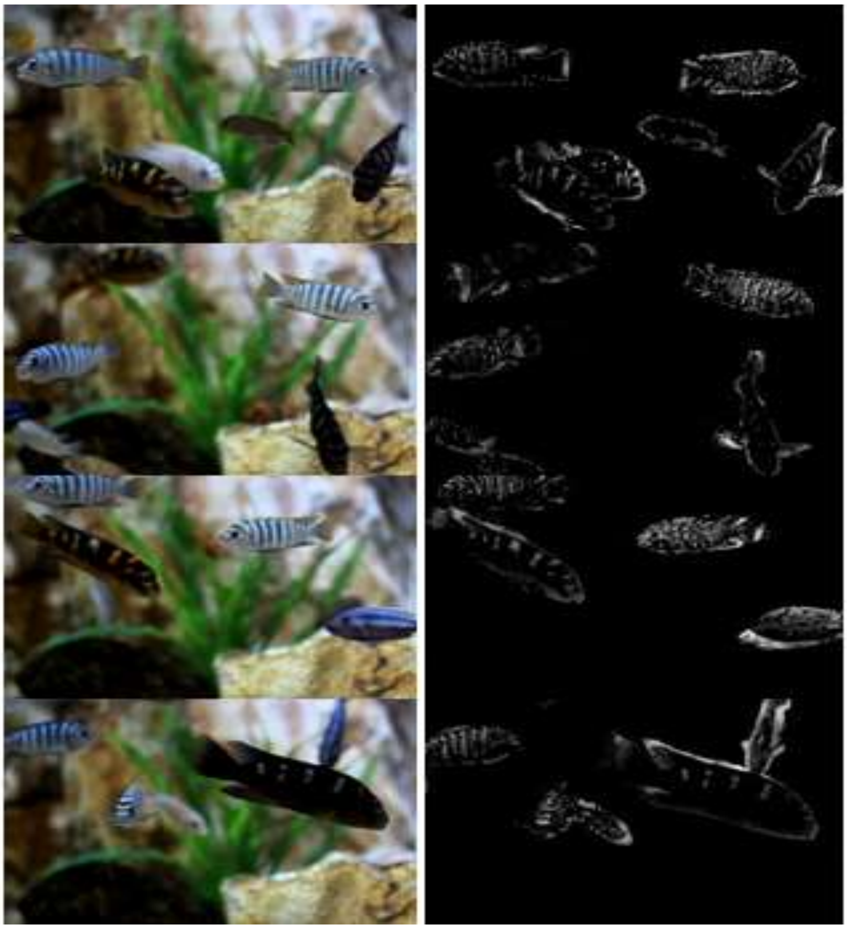


Fig. 5.7 Underwater moving object detection for video1 frames

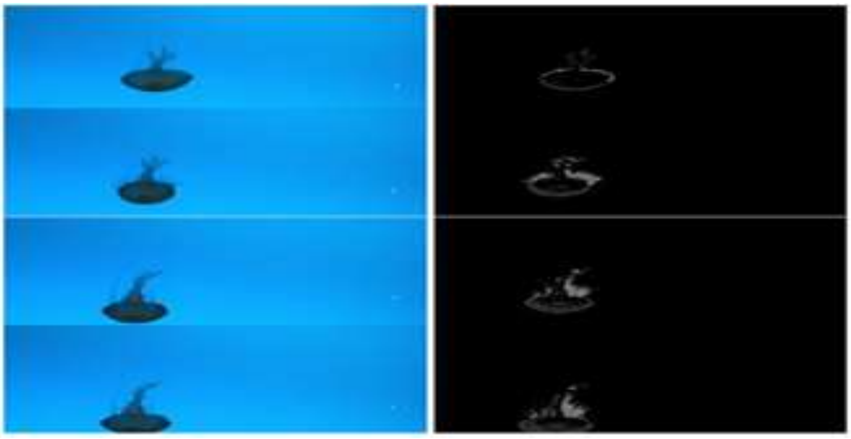


Fig. 5.8 Underwater moving object detection for video2 frames

Table 5.2 Device utilisation – proposed AGMM

Resources	**Number**
Slices	18645
Flip-flop’s	36614
BRAMs	32
Look up Tables	32436
IOBs	58
DSP48s	19

Table 5.3 Comparative analysis

	Proposed Adaptive GMM	Adaptive GMM	Subtraction Method
Sensitivity	0.92	0.89	0.73
Precision	0.84	0.82	0.69
Accuracy	0.88	0.85	0.74

Table 5.3 shows comparative analysis of the different algorithms used for detecting moving objects based on input video frame1 and was observed from the table that the proposed algorithm can tackle moving objects efficiently on the basis of sensitivity, precision and accuracy.

5.10 Summary

This chapter mainly deals with an improved adaptive Gaussian mixture model based moving object detection and its implementation in FPGA. Comparative analysis is performed based on sensitivity, precision and accuracy to validate the results obtained.

CHAPTER 6

ALPHA BETA FILTERING BASED PIPELINE TRACKING

Production of oil and gas from underwater reservoirs generates many technical and engineering challenges. The pipelines that are used for transportation of oil and gas need to be checked regularly. ROVs that are used to inspect underwater pipelines require human intervention [165]. However, this method is very risky. Hence, the solution is to support the human action by using an intellectual vision based navigation and guidance system that includes an efficient method for vision based target detection and tracking methods in underwater environment. A novel alpha beta filtering based tracking system is designed and implemented in FPGA.

6.1 Introduction

Traditionally divers, tow fish and ROVs are used to inspect pipelines. ROVs are robotic submarines used to observe and intervene underwater structures that can operate miles below the surface of water and are tethered to ship's surface with cables that provide electrical power. These cables allow commands to be transmitted and data to be received. Human pilots who sit in a command centre, observe what the ROV is seeing (along with data from other sensors), and can control them using a joystick. These expensive and time-consuming methods require specially trained persons. Video cameras mounted on ROV capture underwater images that are analysed by another human operator for any anomalies. Owing to the low quality of underwater images, this on-line image analysis provides a very challenging job for human operators [166]. By using autonomous underwater vehicles (AUVs) with minimum human interaction, the costs and time can be reduced. The AUV requires consistent localisation and vigorous tracking systems. This requires intelligent target detection and tracking algorithms considering the repetitive nature of the scene and the demand for constant monitoring for lengthy duration.

6.2 Pipeline Tracking Methods

Pipeline can be tracked either by acoustic based systems or by vision based systems. They are briefly explained in the following sections.

6.2.1 Acoustic based Method

A video camera and a side scan sonar can be used for the detection and inspection of pipelines that projects out of the seabed. A magnetometer detects the presence of a cable/pipeline. However, the information such as whether the pipeline is buried or projected out cannot be provided by it. A sub bottom profiler can be used for imaging buried objects. It is capable of finding out the presence of a buried pipeline and the type and thickness of sediment above it. A high resolution sub-bottom profiler, dual-frequency side-scan sonar, and a magnetometer allow to search, monitor and locate underwater pipelines that projects out of the seabed or buried [167]. The artificial objects on the pipeline, route and burial depth, structure and composition of superficial sediments can be detected.

6.2.2 Vision based Method

Underwater optical camera can be used efficiently for autonomous underwater navigation. Vision based systems are being used for navigations such as land, humans and space systems. Compared to acoustic sensors, cameras are considered as a cheaper solution having richer information. The major drawbacks are limited range of vision sensors and large areas of featureless regions in underwater environments. However, even with this limitation, the visual data can still play an important role in the underwater navigation.

Fig. 6.1 shows the Navigation Guidance and Control (NGC) system of an AUV with vision guided navigation system. The output of the vision system goes to the NGC system. Navigation means determining the present location, velocity and altitude while guidance refers to the selection of desired path to travel. The output of the NGC system goes to the main controller that controls the actuators.

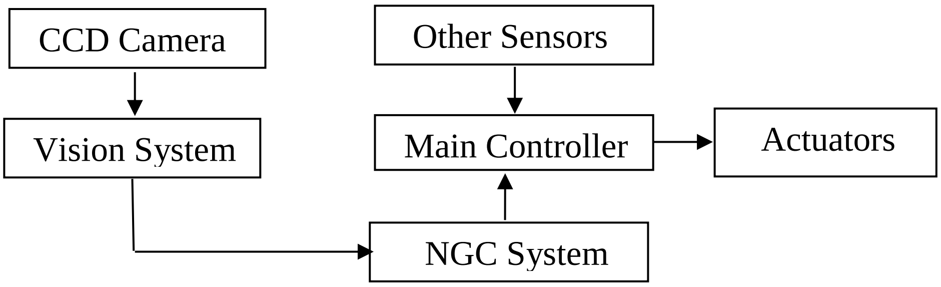


Fig. 6.1 Vision guided navigation system of an AUV

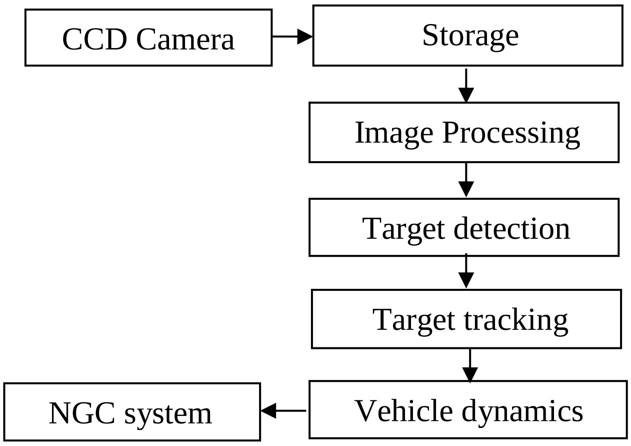


Fig. 6.2 Underwater vision System [168]

Fig. 6.2 shows the vision system in detail. Images, acquired from the camera are processed and can be given to vehicle dynamics for proper navigation.

6.3 Proposed Tracking Algorithm

Underwater image, which is pre-processed using multi-scale fusion technique, is given as an input to the tracking system. Moving object detection system makes sure that the frame is free from moving objects. Once the pre-processed image frame is obtained, the edges are detected using Sobel edge detection followed by Hough transform. Then a novel method of alpha beta tracking is applied to track the pipeline. The proposed method is shown in fig. 6.3. The steps involved in the proposed tracking algorithm are described in the following sections.

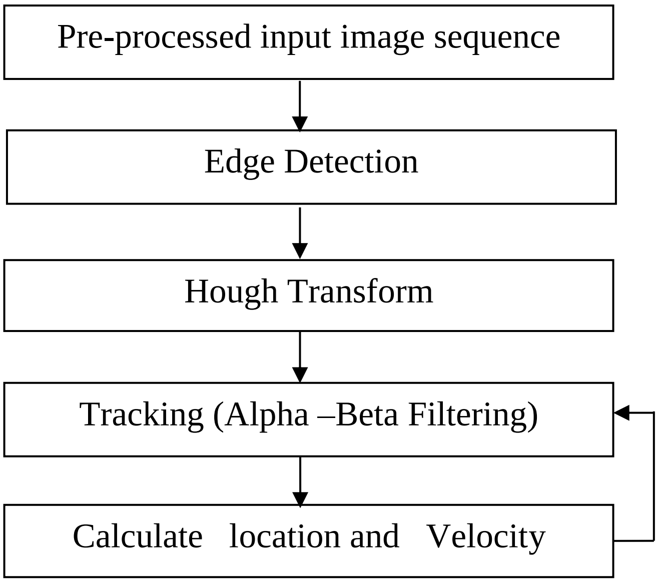

Pre-processed input image sequence
Edge Detection
Hough Transform
Tracking (Alpha –Beta Filtering)
Calculate location and Velocity

Fig. 6.3 Pipeline tracking System – proposed method

6.3.1 Edge Detection

Canny edge detection method and Sobel edge detection method are implemented in FPGA. Sobel edge detection is used for detecting edges in the proposed method due to its simplicity in computation. The main benefit of the Sobel operator lies in its ease of application. The Sobel method gives an estimate to the gradient magnitude. The added advantage of the Sobel operator is that it can identify edges and orientations [169].

The detection of edges and orientations is simple because of the gradient magnitude approximation. Even though Canny edge detection provides an optimal response than the Sobel edge detection, for real time applications, Sobel edge detection is better due to the computational complexity of the Canny edge detection method [170]. The underwater image is first convolved with Sobel masks. The direction and magnitude of the edge points are computed. After calculating the magnitude, edge pixels are identified. Then these edge pixels are subjected to thresholding operation. The thresholding will determine which edge pixel should be retained or discarded as noise.

6.3.2 Hough Transform

The Line Hough Transform (LHT) is a method to determine the value of ρ (distance from the origin to a line) and θ (the angle of the line) of lines in an image [171].

$$\rho_k = xcos\theta_k + ysin\theta_k \tag{6.1}$$

The lines pass through edge points are treated as candidates of true line in the LHT. To compute *(ρ,θ)* of all the candidates, the range of θ *$(0 - \pi)$* is divided by N, and determines the value of $(\rho_n, \mathrm{n} / \mathrm{N} \times \pi)_{\mathrm{n} = (0,\mathrm{N}-1)}$.

With larger N, more accurate ρ and θ is obtained, but it needs more amount of computation time and memory. The curve $\rho = x_1 cos\theta + y_1 sin\theta$ is plotted on *(ρ, θ)* plane, and each point on this curve shows a line in *(x, y)* plane that goes through (x_1, y_l). All points on this curve, (ρ_k, θ_k) provides the true line in the image [171].

The procedure for line detection using Hough transform is explained below.

- Quantize the parameter space *(ρ,θ)* i.e., ρ and θ are made to vary discretely. Create an accumulator array A (two-dimensional) of size ρ_{max} x θ_{max}.

- For each edge point *(x, y), θ* is varied from a minimum value to a maximum value and corresponding ρ value is calculated. The ρ value is rounded off to the nearest integer. Increment the accumulator cell corresponding to row number ρ and column number θ. Repeat this step for each edge point *(x, y).*

- Find the local maxima in the accumulator. If the local maxima is M, then it means that M points lie on the line $x\ cos\theta + y\ sin\theta = \rho$. Thus, the ρ and θ values corresponding to the accumulator cell containing the local maxima M are considered as the parameters of the line and a line is drawn in the image using these parameters.

Hough Transform is used to extract the most significant straight lines from a computer image [171] and the properties of these lines can be used to recognise the underwater track. There are some reasonable assumptions that enable Hough Transform to be used as a suitable method for underwater pipeline detection.

Assumptions used in implementing Hough transform are,

- The roll and pitch of the AUV is passively controlled and hence considered negligible.
- The vision capture device is aimed directly down along on the z-axis,
- The underwater track lies on a horizontal plane
- The underwater track is composed of straight-line segments.

Given these assumptions, a feature of an underwater track that is invariant to rotation and scaling can be defined.

6.3.3 Alpha-Beta Filtering based Tracking

The function of a tracking algorithm is to achieve recursive target state inference, which is given by state and observation equation. In state equation, recent state estimate is projected by time updates in advance of time and in observation equation, the projected estimate is adjusted by the measurement updates by the actual measurement during that time [172].

Based on the state and the observation equations, two important algorithms have been used for positioning the track of a pipeline. They are Kalman filtering (KF) algorithm and alpha beta *(α-β)* filtering algorithm. For the linear Gaussian model, the KF tracking algorithm is considered optimal, and it enhances the estimation accuracy of the system. According to classical deterministic response criteria, this algorithm is effective but its high accuracy results in huge computational complexity and its direct implementation is very tedious for practical systems [173].

Alpha-beta filter was derived as a steady state two-dimensional Kalman filter. The alpha-beta tracking filter is widely used in radar systems, as a single target tracking filter. The most important feature of this tracker is its simplicity, which provides a low computational cost allowing real-time is running. The recursive filter is compounded by two states, with one of them being obtained by the integration of the other over time.

In alpha beta filter, the system is effectively modelled using two internal states. For pipeline tracking, the two states that can be considered are velocity *v* and position *x*. Over a small interval of time *ΔT*, the velocity is considered to be constant. Then the position value is projected at the next sampling time [174].

$$\hat{x}_k = \hat{x}_{k-1} + \Delta T \, \hat{v}_{k-1} \qquad (6.2)$$

Since velocity variable *v* is taken as a constant, its projected value at the next sampling time matches with the current value.

$$\hat{v}_k = \hat{v}_{k-1} \qquad (6.3)$$

The above equation can be adapted if further details are known about how a driving function will transform the *v* state during each time interval. The output is likely to vary from the assumption if dynamic effects in the model is not considered. Based on statistical filtering interpretations, prediction error *r* is called the residual or innovation.

$$\hat{r}_k = x_k - \hat{x}_k$$

(6.4)

If the previous x estimate was low, or the previous v was low, or if both estimates were low, the alpha-beta filter uses alpha times the deviation *r* to correct the position estimate for selected values of alpha and beta [160] that are constants, and to correct the velocity estimate it uses beta times the deviation r. To normalise magnitudes of the multipliers, an extra $\varDelta T$ factor is used.

$$\hat{x}_k = \hat{x}_k + (\alpha)\hat{r}_k$$

(6.5)

$$\hat{v}_k = \hat{v}_k + \frac{\beta}{\Delta t}\hat{r}_k$$

(6.6)

The rectifications can be treated as small steps along the direction of gradient. Error in the state estimates can be lowered by experimental adjustments. The values of the alpha and beta multipliers should be positive and small for convergence and stability. The values should be,

$$0 < \alpha < 1, \;\; 0 < \beta \leq 2, \;\;\; and \; 0 < 4 - 2\alpha - \beta$$

If $0 < \beta < 1$, then noise is suppressed, else the noise is amplified. Larger alpha and beta value result in faster tracking of transient changes, whereas smaller alpha and beta value lower the level of noise in the state estimates. If there is an excellent balance between accurate tracking and noise reduction, filtered estimates are more accurate than the direct measurements and then, the algorithm is effective.

In the proposed method, it is considered that if the AUV will move faster or slower than expected, then, the observed location of the AUV will tend to move away from the expected one. So to incorporate this fact, the equation in 6.5 is modified as follows.

$$\hat{x}_{k+1} = \hat{x}_k + \Delta t \, \hat{v}_k + (\alpha)\hat{r}_k$$

(6.7)

where $\hat{x}_{k+1}$ is new location at time $k+1$ and is updated using information available at time k. Here location estimate uses speed estimate at time k, thereby the changes in the speed can also be incorporated in the model. The values of α and β were chosen experimentally.

6.4 Implementation in FPGA

The tracking system was implemented in FPGA Virtex4 FPGA. The interface circuitry designed in the schematic document includes adding the peripherals to the designed processor and wiring between them. The main processor system, designed using the open bus system, was first inserted into the schematic as a sheet symbol. The peripherals are added from the library and connections were made. C-to-Hardware Compiler (CHC) feature of Altium designer was used. The generated HDL file is integrated after compilation. The output design was instantiated on the FPGA.

Different values of velocity were given between certain numbers of frames to test the system for variable speed.

6.5 Results and Discussions

Pre-processed image frames were given as input to the vision system. Here colour correction was not performed while giving the input to the system as edge detection does not make any difference between colour corrected enhanced image and the normal enhanced image using fusion. Fig. 6.4, 6.5 and 6.6 show the output of the edge detection step for straight pipeline, bent pipeline and a partially submerged pipeline respectively.

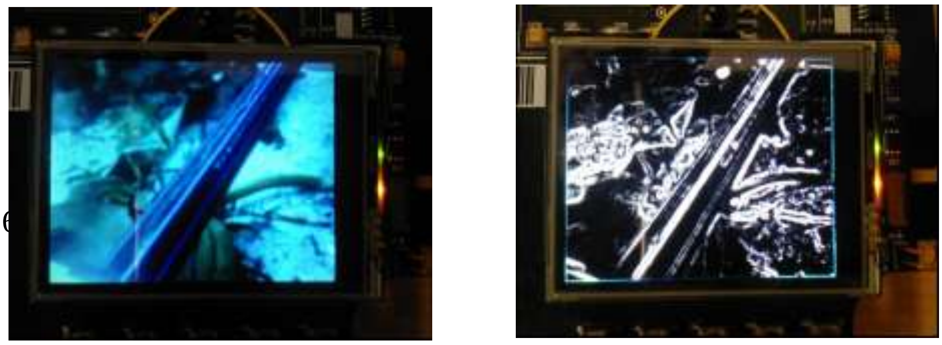


Fig. 6.4 Implementation of Edge Detection for a straight pipeline

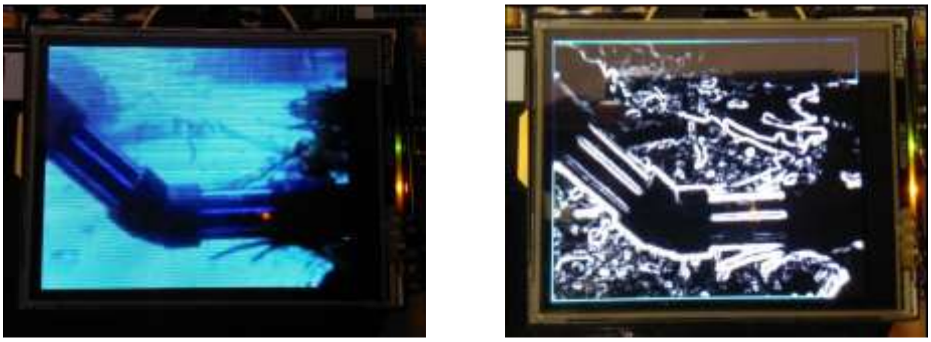


Fig. 6.5 Implementation of Edge Detection for a bent pipeline

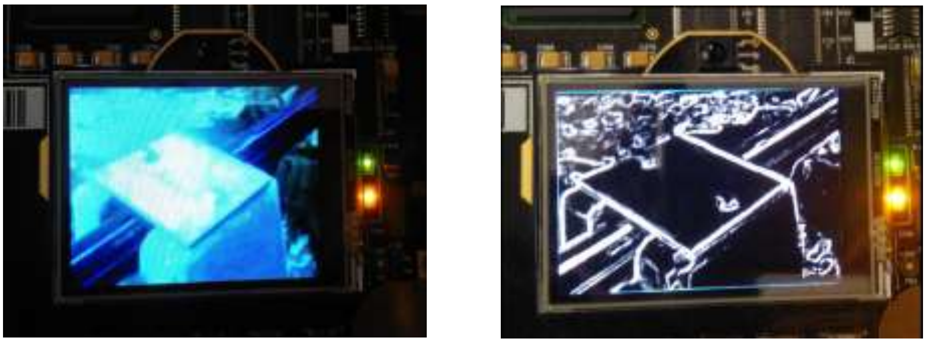


Fig. 6.6 Implementation of Edge detection for a partially submerged pipeline

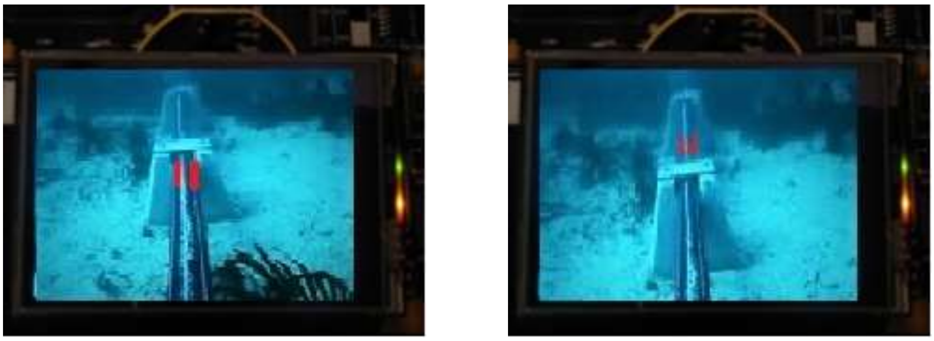
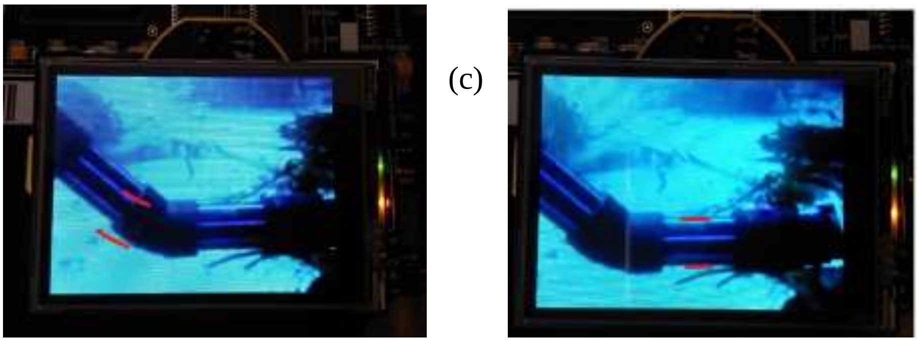


Fig. 6.7 (a) (b)



(d)

Fig. 6.7 Pipeline tracking at different frames of an underwater video (a-d)

Figures 6.7 shows the output of the alpha-beta tracking filter for different frames of an underwater video. Table 6.1 shows the device utilisation of the implementation in FPGA. An average of 58% of the total available resources have been used by this design.

Table 6.1 Device Utilisation – Alpha Beta filtering

Resources	**Number**
Slices	21632
Flip-flop’s	46618
BRAMs	35
Look up Tables	36908
IOBs	59
DSP48s	29

Table 6.2 shows the comparative analysis of tracking algorithms based on intercept location if it falls within a certain limit of pipeline. The comparison is made by simulating the algorithms with previously declared paths. For variable velocities, the proposed algorithm is found to better than the conventional $\alpha\beta$ filtering as per the results obtained.

Table 6.2 Comparative analysis for fixed velocity

Trajectory	α β filtering - conventional		α β filtering - proposed	
	Current position	Estimated Position	Current Position	Estimated Position
Sine	85	91	85	90
Step	92	87	92	87
Random	78	72	78	73

Table 6.3 Comparative analysis for variable velocity

Trajectory	α β filtering - conventional		α β filtering - proposed	
	Current position	Estimated Position	Current Position	Estimated Position
Sine	98	92	98	94
Step	113	101	113	104

FPGA@200 MHz, 320 x 240 image	Tracking based on α β filtering -conventional	Tracking based on α β filtering - proposed
Power Consumption	6205 mw	6228 mw
Processing time	1.21 ms	1.27 ms

Random	87	80	87	83

Table 6.4 Comparison of power consumption and speed

Table 6.4 shows comparative analysis of the conventional and the proposed tracking based on $\alpha\beta$ filtering. Processing time and power are compared for an image size of 320 x 240. From the table, it is clear that the power and processing time are slightly higher for the proposed method. This is due to the inclusion of additional parameter in the design to reduce the error while speed of the vehicle is varied.

6.6 Summary

This chapter describes alpha beta filtering based tracking method. Sobel edge detection and Hough Transform are performed, followed by a modified alpha beta filtering for tracking pipelines. The design was implemented in Virtex 4 FPGA. The results are validated based on position estimate, power and processing time.

CHAPTER 7

CONCLUSIONS

The thesis addresses implementation of vision based techniques in Field Programmable Gate Arrays that can be tested in AUVs for cable/pipeline tracking. FPGA can be used where low volume VLSI design is needed and when a prototype has to be needed before going for chip fabrication. A novel multi-scale pixel based image fusion method was used for pre-processing underwater images as those images were suffered from colour degradation and poor illumination due to ocean depth. Objects in front of the vehicles can be detected using a novel method based on adaptive Gaussian mixture model with Otsu's thresholding. Here, frames with sharp edges are extracted by deblurring and multiple moving objects were tracked by using AGMM after the noise removal by median filtering. Finally underwater pipeline tracking was carried out using an improved method based on Hough transform followed by $\alpha\beta$ filtering based tracking. The algorithms were successfully implemented in FPGA.

7.1 Image Enhancement

Underwater vision is plagued by poor visibility conditions producing images with poor contrast and colour variation. These images need to be enhanced for further applications. So many image enhancement techniques are available for which suitable ones for non-uniform illumination nature of light in the underwater were selected. Modifications in the existing algorithm were made for more visual enhancements and colour correction of these images.

Multi pixel-based image fusion method was used for enhancing underwater images as a pre-processing method. Different weight maps, suitable for underwater scenario, from a single image were developed and were properly fused to get the enhanced image. A single image based fusion algorithm was developed in such a way that it can handle underwater image degradation effectively. Underwater image enhancements techniques like homomorphic filtering, multi-scale retinex algorithm and adaptive histogram equalisation were also modified to be used in underwater environments and were successfully implemented in FPGA.

As per the qualitative analysis, it was found that the multi-scale pixel based image fusion method outperformed other underwater image enhancement techniques like homomorphic filtering, AHE and multi-scale retinex algorithm in terms of root mean square error (RMSE), peak signal to noise ratio (PSNR) and mean absolute error (MAE). The device utilisation in FPGA was slightly higher in the case of fusion technique, but since it occupied only less than 30% percentage of available resources, it makes no significant difference in power consumption.

7.2 Moving Object Detection

Due to complex and dynamic environment inside the water, fast moving objects cannot be detected accurately. Movement of object at different velocities and at different times are the most serious problems. Moving objects were tracked using AGMM. Deblurring and then using a modified median filtering improved the sharpness of the image and thereby the detection of multiple moving objects by AGMM became accurate. The accuracy was increased by incorporating modified method based on Otsu's thresholding.

Comparative analysis was carried out and the modified AGMM algorithm can tackle moving objects efficiently based on quality metrics such as sensitivity, precision and accuracy.

7.3 Pipeline Tracking

Finally, pipeline tracking algorithm was modified to estimate the position of the vehicle accurately, and the design was implemented in FPGA that can be used in the underwater scenario. Hough transform followed by alpha beta tracking was used for tracking. Variations in the speed of the tracking vehicle were incorporated in the new model. The algorithm was simulated and compared with the existing $\alpha\beta$ filtering for different trajectories like sine, step and random. The algorithm was compared for fixed velocity and variable velocity. The comparative analysis has shown that the proposed algorithm is better for tracking cables/pipelines.

7.4 Implementation

The design was targeted on Xilinx FPGA Virtex 4. Various vendors are available for manufacturing FPGAs, like Altera, Xilinx, Actel etc. For this work, Xilinx FPGAs were chosen. For implementing designs on Xilinx FPGAs, Xilinx ISE Design Suite is the software platform. In this work, System generator and Altium designer was used as front end software tools in association with Xilinx ISE design suite, targeting Virtex 4 FPGA.

7.5 Future Work

The objective of this thesis work is to develop and implement a vision system in FPGA for efficiently enhancing underwater images, detecting underwater moving objects and for tracking underwater pipeline/cable. Here, offline underwater videos and images were chosen to carry out the work. In future, this work can be tested for real time performance in underwater scenario. The output from the FPGA can be used to control AUV. Comparative analysis can also be done by implementing the algorithm in a digital signal processor. Automatic detection of cracks and other problems can also be identified during inspection. Hence, human efforts can be minimised to a great extent.

List of Publications

[1] Alex Raj S. M. Khadeeja N. and Supriya M. H., “Implementation of Histogram Based Image Fusion Technique for Underwater Image Enhancement in Reconfigurable Platform,” *Indian Journal of Science and Technology*, vol. 10, no. 26, 2017 (indexed in Scopus and Thomson Reruters(WoS)).

[2] Alex Raj S M, Sukanya L. J. and Supriya M H, “Deblurring and Tracking of Moving Objects from Low Contrast Underwater Images,” *International Journal of Control Theory and Applications*, vol. 10, no. 37, pp. 239–246, 2017 (indexed in Scopus).

[3] Alex Raj S. M., Claris Jose, And Supriya M. H., “Hardware Realization Of Canny Edge Detection Algorithm For Underwater Image Segmentation Using Field Programmable Gate Arrays,” *Journal of Engineering Science and Technology, School of Engineering*, Malaysia vol. 12, no. 9, pp. 2536–2550, 2017 (indexed in Scopus).

[4] Alex Raj S. M. and Supriya M. H, “Hardware Co-simulation of Underwater Moving Object Detection using Xilinx System Generator,” *International Journal of Oceans and Oceanography* ISSN, vol. 10, no. 1, pp. 973–2667, 2016 (indexed in Scopus).

[5] Alex Raj S.M., Abhilash S., and Supriya M.H., “A Comparative Study of Various Methods for Underwater Image Enhancement and Restoration,” *IOSR Journal of VLSI and Signal Processing*, vol. 6, no. 2, pp. 30–33, 2015.

[6] Alex Raj S. M. Rita Maria Abraham and Supriya M. H., “Spatial filtering based Boundary Extraction in Underwater Images for Pipeline Detection: FPGA Implementation,” *International Journal of Computer Science and Information Security*, USA, vol. 14, no. 9, pp. 790–794, 2016 (indexed in Thomson Reuters).

[7] Alex Raj S. M., Khadeeja N. and Supriya M. H., “Performance Evaluation of Image Processing Algorithms for Underwater Image

Enhancement in FPGA," *IOSR Journal of VLSI and Signal Processing*, vol. 5, no. 4, pp. 17–21, 2015.

[8] Alex Raj S.M. and Supriya M. H., "Reconfigurable Platform Based Design in FPGA for Underwater Image Colour Correction," *The IIOAB journal*, vol. 7, pp. 76–81, 2016 (indexed in Thomson Reuters).

[9] Alex Raj S. M. Aruna S. Rajan and Supriya M. H., "A Comparative Study of Vision Guided AUV Navigation Techniques for Pipeline/Cable Inspection," *SSRG International Journal of VLSI & Signal Processing*, vol. 2, no. 4, pp. 27-32, 2015.

[10] S. M. Alex Raj and M. H. Supriya, "FPGA Implementation of Underwater Image Enhancement using Nonlinear filtering," *Indian Journal of Science and Technology*, vol. 8, no. 35, Dec. 2015 (indexed in Scopus and Thomson Reuters(WoS)).

[11] Alex Raj S.M., Rita Maria Abraham and Supriya M. H., "Vision-Based Underwater Cable/Pipeline Tracking Algorithms in AUVs: A Comparative Study," *International Journal Of Engineering And Advanced Technology.*, vol. 5, no. 4, pp. 48–52, 2016.

[12] Alex Raj S.M. and Supriya M.H., "Underwater Image Enhancement using Single Scale Retinex on a Reconfigurable Hardware," *in International Symposium on Ocean Electronics (SYMPOL),* pp. 1–5, 2015.

[13] Alex Raj S. M., Deepa S., and Supriya M. H., "Underwater Image Enhancement using CLAHE in a Reconfigurable Platform," *in OCEANS 2016 MTS/IEEE Monterey*, USA, pp. 1–5, 2016.

List of Publications Outside the Scope of the Thesis

[1] S. M. Alex Raj, Sreelatha G., and Supriya M. H**.**, "Gesture recognition using field programmable gate arrays," *in IEEE International Conference on Devices, Circuits and Systems (ICDCS),* pp. 72–75, 2012.

References

[1] G. L. Foresti, “Visual inspection of sea bottom structures by an autonomous underwater vehicle,” *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 31, no. 5, pp. 691–705, 2001.

[2] G. Kim, D. Kim, D. Kim, H. T. Choi, D. Lee, and H. Myung, “Path Planning and T racking of an Autonomous Underwater Vehicle using Virtual Way-points,” *The Abstracts of the international conference on advanced mechatronics : toward evolutionary fusion of IT and mechatronics : ICAM*, vol. 2010.5, no. 0, pp. 118–123, 2010.

[3] M. Lanzagorta and Marco, “Quantum imaging for underwater arctic navigation,” in *Proceedings of the SPIE, Volume 10188, id. 101880G 27 pp. (2017).*, vol. 188, p. 101880G, 2017..

[4] H. Zhu, F. H. Y. Chan, and F. K. Lam, “Image Contrast Enhancement by Constrained Local Histogram Equalization,” *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 281–290, Feb. 1999.

[5] S. Aslani and H. Mahdavi-Nasab, “Optical flow based moving object detection and tracking for traffic surveillance,” *World Academy of Science, Engineering*, 2013.

[6] Q. Fu and M. Celenk, “Marked watershed and image morphology based motion detection and performance analysis,” in *2013 8th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pp. 159–164, 2013.

[7] N. Mohamed, I. Jawhar, J. Al-Jaroodi, and L. Zhang, “Sensor Network Architectures for Monitoring Underwater Pipelines,” *Sensors*, vol. 11, no. 12, pp. 10738–10764, Nov. 2011.

[8] J. Villasenor and B. Hutchings, “The flexibility of configurable computing,” *IEEE Signal Processing Magazine*, vol. 15, no. 5, pp. 67–84, 1998.

[9] Takashi Saegusa, Tsutomu Maruyama, and Yoshiki Yamaguchi, “How fast is an FPGA in image processing?,” in *2008 International Conference on Field Programmable Logic and Applications*, pp. 77–82, 2008.

[10] S. Hasan, “Performance-Aware Architectures for Parallel 4D Color fMRI Filtering Algorithm: A Complete Performance Indices Package,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 7, pp. 2116–2129, Jul. 2016.

[11] S. Boussakta, “A novel method for parallel image processing applications,” *Journal of Systems Architecture*, vol. 45, no. 10, pp. 825–839, Apr. 1999.

[12] D. M. Kocak, F. R. Dalgleish, F. M. Caimi, and Y. Y. Schechner, “A Focus on Recent Developments and Trends in Underwater Imaging,” *Marine Technology Society Journal*, vol. 42, no. 1, pp. 52–67, 2008.

[13] M. Kumar and S. Kumar Thakur, “Comparison of Filters used for Underwater Image Pre-Processing,” *IJCSNS International Journal of Computer Science and Network Security*, vol. 10, no. 1, 2010.

[14] C. J. Prabhakar and P. U. P. Kumar, “An Image Based Technique for Enhancement of Underwater Images,” *arXiv preprint arXiv:1212.0291*, Dec. 2012.

[15] K. Plakas and E. Trucco, “Uncalibrated computer vision technique in the underwater environment,” in *IEE Colloquium on Underwater Applications of Image Processing*, vol. 18, pp. 11–15, 1998.

[16] J. Y. Chiang and Ying-Ching Chen, “Underwater Image Enhancement by Wavelength Compensation and Dehazing,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1756–1769, Apr. 2012.

[17] F. Guo, Z. Cai, B. Xie, and J. Tang, “Automatic Image Haze Removal Based on Luminance Component,” in *2010 International Conference on Computational Intelligence and Software Engineering*, pp. 1–4, 2010.

[18] Kaiming He, Jian Sun, and Xiaoou Tang, “Single Image Haze Removal Using Dark Channel Prior,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2341–2353, Dec. 2011.

[19] J.-H. Kim, J.-Y. Sim, and C.-S. Kim, “Single image dehazing based on contrast enhancement,” in *2011 IEEE International Conference*

on Acoustics, Speech and Signal Processing (ICASSP), pp. 1273–1276, 2011.

[20] J. P. Oakley and H. Bu, “Correction of Simple Contrast Loss in Color Images,” *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 511–522, Feb. 2007.

[21] R. Fattal, Raanan, Fattal, and Raanan, “Single image dehazing,” *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 12-17, Aug. 2008.

[22] C. Ancuti, C. O. Ancuti, T. Haber, and P. Bekaert, “Enhancing underwater images and videos by fusion,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 81–88, 2012.

[23] N. Carlevaris-Bianco, A. Mohan, and R. M. Eustice, “Initial results in underwater single image dehazing,” in *OCEANS MTS/IEEE SEATTLE*, pp. 1–8, 2010.

[24] L. Kratz and K. Nishino, “Factorizing Scene Albedo and Depth from a Single Foggy Image,” in *IEEE 12th International Conference on Computer Vision*, pp. 1701–1708, 2009.

[25] Kaiming He, Jian Sun, and Xiaoou Tang, “Single Image Haze Removal Using Dark Channel Prior,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2341–2353, Dec. 2011.

[26] R. T. Tan, “Visibility in bad weather from a single image,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[27] H. Singh, J. Adams, D. Mindell, and B. Foley, “Imaging Underwater for Archaeology,” *Journal of Field Archaeology*, vol. 27, no. 3, pp. 319–328, Jan. 2000.

[28] J. Paik, C. P. Lee, and M. A. Abidi, “Image Processing-Based Mine Detection Techniques: A Review,” *Subsurface Sensing Technologies and Applications*, vol. 3, no. 3, pp. 153–202, 2002.

[29] A. Hétet, I. Quidu, and Y. Dupas, “Obstacle Detection and Avoidance for AUV: problem analysis and first results (Redermor),” pp. 29-32, 2006.

[30] A. Weidemann, G. R. Fournier, L. Forand, and P. Mathieu, "In Harbor Underwater Threat Detection/Identification Using Active Imaging." ICDCS, pp.12-15, 2005.

[31] Xu Wen, Wang Yuling, and Zhu Weiqing, "Sonar image processing system for an autonomous underwater vehicle (AUV)," in *"Challenges of Our Changing Global Environment". in Conference Proceedings. OCEANS '95 MTS/IEEE*, vol. 3, pp. 1883–1886, 1995.

[32] Xiaohai Yuan, Chenchang Qiu, Rongsheng Chen, Zhen Hu, and Peilin Liu, "Vision system research for autonomous underwater vehicle," in *1997 IEEE International Conference on Intelligent Processing Systems (Cat. No.97TH8335)*, vol. 2, pp. 1465–1469, 1997.

[33] T. Maki, H. Kondo, T. Ura, and T. Sakamaki, "Navigation of an Autonomous Underwater Vehicle for Photo Mosaicing of Shallow Vent Areas," in *OCEANS 2006 - Asia Pacific*, pp. 1–7, 2006.

[34] B. A. A. P. Balasuriya, M. Takai, W. C. Lam, T. Ura, and Y. Kuroda, "Vision based autonomous underwater vehicle navigation: underwater cable tracking," in *Oceans '97. MTS/IEEE Conference Proceedings*, vol. 2, pp. 1418–1424,1997.

[35] H. Wang, R. Marks, S. Rock, M. Lee, and R. Burton, "Combined camera and vehicle tracking of underwater objects," *Proceedings of ROV*, pp.67-71, 1992.

[36] and A. Z. David Wettergreen, Chris Gaskett, "Autonomous Guidance and Control for an Underwater Robotic Vehicle."International conference on robotics, pp.56-62, 2011.

[37] A. Balasuriya and T. Ura, "Underwater cable following by Twin-Burger 2," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 1, pp. 920–925, 2001.

[38] T. L. Institution of Electrical Engineers., K. S. Sim, and C. P. Tso, *Electronics letters.*, vol. 48, no. 3. [Institution of Electrical Engineers], 2012.

[39] S. P. Ehsani, H. S. Mousavi, and B. H. Khalaj, "Chromosome Image Contrast Enhancement Using Adaptive, Iterative Histogram Matching," in *2011 7th Iranian Conference on Machine Vision and*

Image Processing, pp. 1–5, 2011.

[40] M. Liu and P. Liu, “Image Enhancement Algorithm for Video Based on Multi-dimensional Biomimetic Informatics,” in *2012 International Conference on Computer Science and Electronics Engineering*, pp. 81–83, 2012.

[41] R. K. Jha, R. Chouhan, and P. K. Biswas, “Noise-induced contrast enhancement of dark images using non-dynamic stochastic resonance,” in *2012 National Conference on Communications (NCC)*, pp. 1–5, 2012.

[42] S. Parthasarathy and P. Sankaran, “Fusion based Multi Scale RETINEX with Color Restoration for image enhancement,” in *2012 International Conference on Computer Communication and Informatics*, pp. 1–7, 2012.

[43] D. Ghimire and J. Lee, “Nonlinear transfer function-based local approach for color image enhancement,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 858–865, May 2011.

[44] H. Zhang, Q. Zhao, L. Li, Y. Li, and Y. You, “Muti-scale image enhancement based on properties of human visual system,” in *2011 4th International Congress on Image and Signal Processing*, pp. 704–708, 2011.

[45] A. Ramirez Rivera, Byungyong Ryu, and O. Chae, “Content-Aware Dark Image Enhancement Through Channel Division,” *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 3967–3980, Sep. 2012.

[46] S. Bronte, L. M. Bergasa, and P. F. Alcantarilla, “Fog detection system based on computer vision techniques,” in *2009 12th International IEEE Conference on Intelligent Transportation Systems*, pp. 1–6, 2009.

[47] *Poljicak L. Proceedings of the Conference on Systems Signals and Image Processing.* IEEE, pp.25-30, 2012.

[48] K. Hasikin and N. A. M. Isa, “Enhancement of the Low Contrast Image Using Fuzzy Set Theory,” in *2012 UKSim 14th International Conference on Computer Modelling and Simulation*, pp. 371–376, 2012.

[49] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar, “Instant dehazing of images using polarization,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, p. I-325-I-332, *2001*.

[50] S. G. Narasimhan and S. K. Nayar, "Removing weather effects from monochrome images." *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Vol. 2. 2001.

[51] S. G. Narasimhan and S. K. Nayar, “Contrast restoration of weather degraded images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 713–724, Jun. 2003.

[52] A. Balasuriya and T. Ura, “Multi-sensor fusion for autonomous underwater cable tracking,” in *Oceans ’99. MTS/IEEE. Riding the Crest into the 21st Century. Conference and Exhibition. Conference Proceedings (IEEE Cat. No.99CH37008)*, vol. 1, pp. 209–215, 1999.

[53] J. Kojima, “Cable tracking by autonomous underwater vehicle,” in *2003 International Conference Physics and Control. Proceedings (Cat. No.03EX708)*, pp. 171–174, 2003.

[54] J. Antich and A. Ortiz, “Underwater Cable Tracking by Visual Feedback,” Iberian Conference on Pattern Recognition and Image Analysis, pp. 53–61, 2003.

[55] A. Ortiz, M. Simó, and G. Oliver, “A vision system for an underwater cable tracker,” *Machine Vision and Applications*, vol. 13, no. 3, pp. 129–140, Jul. 2002.

[56] S. Cowen, S. Briest, and J. Dombrowski, “Underwater docking of autonomous undersea vehicles using optical terminal guidance,” in *Oceans ’97. MTS/IEEE Conference Proceedings*, vol. 2, pp. 1143–1147,1997.

[57] D. R. Edgington, K. A. Salamy, M. Risi, R. E. Sherlock, D. Walther, and Christof Koch, “Automated event detection in underwater video,” in *Oceans 2003. (IEEE Cat. No.03CH37492)*, 2003, p. P2749–P2753 Vol.5, 2003.

[58] D. Walther, D. R. Edgington, and C. Koch, “Detection and tracking of objects in underwater video,” in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern*

Recognition, CVPR 2004., vol. 1, pp. 544–549, *2004.*

[59] E. F. Morais, M. F. M. Campos, F. L. C. Padua, and R. L. Carceroni, “Particle Filter-Based Predictive Tracking for Robust Fish Counting,” in *XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI’05)*, pp. 367–374, 2005.

[60] C. Spampinato, Y.-H. Chen-Burger, G. Nadarajan, and R. B. Fisher, “Detecting ,Tracking And Counting Fish In Low Quality Unconstrained Underwater Videos, VISAPP (2), vol.2, pp. 514-519, 2008.

[61] A. D. Sappa and F. Dornaika, “An Edge-Based Approach to Motion Detection,” Springer, Berlin, Heidelberg, 2006, pp. 563–570.

[62] S. Kumari, M. Kaur, and B. Singh, “Detection And Tracking Of Moving Object in Visual Surveillance System,” *International Journal of Advanced Research in Electrical, Electronics and Instrumentation EngineeringISO Certified Organization)*, vol. 3297, no. 8, 2007.

[63] C. Ridder, O. Munkelt, and H. Kirchner, “Adaptive Background Estimation and Foreground Detection using Kalman-Filtering” Proceedings of International Conference on recent Advances in Mechatronics (pp. 193-199), 1995.

[64] Adnan Khashman, “Automatic Detection, Extraction and Recognition of Moving Objects,” International Journal Of Systems Applications, Engineering & Development, vol. 2, no. 1, 2008.

[65] S. Vahora, N. Chauhan, and N. Prajapati, “A Robust Method for Moving Object Detection Using Modified Statistical Mean Method,” *International Journal of Advanced Information Technology (IJAIT)*, vol. 2, no. 1, 2012.

[66] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, “Background and foreground modeling using nonparametric kernel density estimation for visual surveillance,” *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1163, Jul. 2002.

[67] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, “Moving target classification and tracking from real-time video,” in *Proceedings Fourth IEEE Workshop on Applications of Computer Vision.*

WACV'98 (Cat. No.98EX201), pp. 8–14,1998.

[68] Y. Dedeoğlu, B. U. Töreyin, U. Güdükbay, and A. E. Çetin, "Silhouette-Based Method for Object Classification and Human Action Recognition in Video," *Int. Workshop on Human-Computer Interaction*, pp. 64–77, 2006.

[69] I. K. Sethi and R. Jain, "Finding Trajectories of Feature Points in a Monocular Image Sequence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 1, pp. 56–73, Jan. 1987.

[70] V. Salari and I. K. Sethi, "Feature point correspondence in the presence of occlusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 87–91, 1990.

[71] K. Rangarajan and M. Shah, "Establishing motion correspondence," in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 103–108, 1991.

[72] S. S. Intille, J. W. Davis, and A. F. Bobick, "Real-time closed-world tracking," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 697–703, 1997.

[73] C. J. Veenman, M. J. T. Reinders, and E. Backer, "Resolving motion correspondence for densely moving points," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 1, pp. 54–72, 2001.

[74] K. Shafique and M. Shah, "A noniterative greedy algorithm for multiframe point correspondence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 51–65, Jan. 2005.

[75] P. Fieguth and D. Terzopoulos, "Color-based tracking of heads and other mobile objects at video frame rates," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 21–27, 1997.

[76] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[77] D. Comaniciu, "Bayesian Kernel Tracking," Pattern Recognition, pp.

438–445, 2002.

[78] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, “Robust online appearance models for visual tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, Oct. 2003.

[79] Hai Tao, H. S. Sawhney, and R. Kumar, “Object tracking with Bayesian estimation of dynamic layer representations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 75–89, 2002.

[80] M. Isard and J. MacCormick, “BraMBLe: a Bayesian multiple-blob tracker,” in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, pp. 34–41, 2001.

[81] M. Narimani, S. Nazem, and M. Loueipour, “Robotics vision-based system for an underwater pipeline and cable tracker,” in *OCEANS 2009-EUROPE*, 2009, pp. 1–6.

[82] G. L. Foresti and S. Gentili, “A Vision Based System For Object Detection In Underwater Images,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 14, no. 2, pp. 167–188, Mar. 2000.

[83] T. Zhang, W. Zeng, L. Wan, and Z. Qin, “Vision-based system of AUV for an underwater pipeline tracker,” *China Ocean Engineering*, vol. 26, no. 3, pp. 547–554, Sep. 2012.

[84] S. Asano, T. Maruyama, and Y. Yamaguchi, “Performance comparison of FPGA, GPU and CPU in image processing,” in *2009 International Conference on Field Programmable Logic and Applications*, pp. 126–131, 2009.

[85] B. Cope, P. Y. K. Cheung, W. Luk, and L. Howes, “Performance Comparison of Graphics Processors to Reconfigurable Logic: A Case Study,” *IEEE Transactions on Computers*, vol. 59, no. 4, pp. 433–448, Apr. 2010.

[86] K. Pauwels, M. Tomasi, J. Diaz Alonso, E. Ros, and M. M. Van Hulle, “A Comparison of FPGA and GPU for Real-Time Phase-Based Optical Flow, Stereo, and Local Image Features,” *IEEE Transactions on Computers*, vol. 61, no. 7, pp. 999–1012, Jul. 2012.

[87] R. Kalarot and J. Morris, “Comparison of FPGA and GPU implementations of real-time stereo vision,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pp. 9–15, 2010.

[88] “Performance Evaluation of Vision Algorithms on FPGA - Mahendra Gunathilaka Samarawickrama - Google Books.

[89] S. Hasan, A. Yakovlev, and S. Boussakta, “Performance efficient FPGA implementation of parallel 2-D MRI image filtering algorithms using Xilinx system generator,” *7th International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP)*, p. 765–769., 2010.

[90] S. Hasan, S. Boussakta, and A. Yakovlev, “Improved parameterized efficient FPGA implementations of parallel 1-D filtering algorithms using Xilinx System Generator,” *2010 IEEE International Symposium on Signal Processing and Information Technology, ISSPIT 2010*, pp. 382–387, 2011.

[91] Z. Shanshan and W. Xiaohong, “Vehicle image edge detection algorithm hardware implementation on FPGA,” *International Conference on Computer Application and System Modeling (ICCASM)*, no. Iccasm, pp. 184–188, 2010.

[92] a. C. Atoche, M. P. Cortes, J. V. Castillo, and R. A. Ensenat, “FFT Implementation for Electronic Holograms using Field Programmable Gate Array,” *2006 International Caribbean Conference on Devices, Circuits and Systems*, no. 1, pp. 115–119, 2006.

[93] M. Kiran, K. M. War, L. M. Kuan, and L. K. Meng, “Implementing image processing algorithms using ‘Hardware in the loop’ approach for Xilinx FPGA,” *International Conference on Electronic Design (ICED 2008), Penang*, pp. 1–6, 2008.

[94] Z. Que *et al.*, “Implementing medical CT algorithms on stand-alone FPGA based systems using an efficient workflow with SysGen and Simulink,” *Proceedings - 7th IEEE International Conference on Embedded Software and Systems, ICESS-2010, ScalCom-2010*, no. Cit, pp. 2391–2396, 2010.

[95] V. Ila, R. Garcia, and F. Charot, “VLSI architecture for an Underwater Robot Vision System,” in *Europe Oceans 2005*, p. 674–679 Vol. 1, 2005.

[96] K. Nakano and E. Takamichi, “An image retrieval system using FPGAs,” *Proceedings of the 2003 conference on Asia South Pacific design automation - ASPDAC*, p. 370, 2003.

[97] S. Che, J. Li, J. W. Sheaffer, K. Skadron, and J. Lach, “Accelerating Compute-Intensive Applications with GPUs and FPGAs,” in *2008 Symposium on Application Specific Processors*, pp. 101–107, 2008.

[98] I. Ishii, T. Taniguchi, R. Sukenobe, and K. Yamamoto, “Development of high-speed and real-time vision platform, H^3 vision,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3671–3678, 2009.

[99] A. Suthar, M. Vayada, and C. Patel, “Hardware software co-simulation for image processing applications,” *IJCSI International Journal*, 2012.

[100] R. S, A. Rahim, and F. Shaik, “FPGA Based Design and Implementation of Image Edge Detection Using Xilinx System Generator,” *International Journal of Engineering Trends and Technology*, vol. 4, no. 10, 2013.

[101] V. Elamaran and G. Rajkumar, “FPGA Implementation Of Point Processes Using Xilinx System Generator,” *Journal of Theoretical and Applied Information Technology*, vol. 31, no. 412, 2012.

[102] A. Acharya, R. Mehra, and V. S. Takher, “FPGA Based Non Uniform Illumination Correction in Image Processing Applications,” *Int. J. Comp. Tech.*, vol. 2, no. 2, pp. 349–358, 2009.

[103] K. Gribbon, D. Bailey, and C. Johnston, “Design Patterns for Image Processing Algorithm Development on FPGAs,” in *TENCON 2005 - 2005 IEEE Region 10 Conference*, pp. 1–6, 2005.

[104] S. V. Devika, S. V. Devika, and S. Khumuruddeen, “Hardware implementation of Linear and Morphological Image Processing on FPGA" *International Journal of Engineering Research and Applications (IJERA)* Vol. 2, Issue 1, pp.645-650, 2012.

[105] B. A. Draper, J. R. Beveridge, A. P. W. Bohm, C. Ross, and M. Chawathe, “Accelerated image processing on FPGAs,” *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1543–1551, Dec. 2003.

[106] A. Manan, "Implementation of Image Processing Algorithm on FPGA," *Akgec Journal of Technology*, vol.2, no.1, pp. 25-31, 2006.

[107] D. C. M. Bilsby, R. L. Walke, and R. W. M. Smith, "Comparison of a programmable DSP and a FPGA for real-time multiscale convolution," in *IEE Colloquium on High Performance Architectures for Real-Time Image Processing*, vol. 19, pp. 4–9, 1998.

[108] Z. Salcic and J. Sivaswamy, "IMECO: a reconfigurable FPGA-based image enhancement co-processor framework," in *TENCON '97 Brisbane - Australia. Proceedings of IEEE TENCON '97. IEEE Region 10 Annual Conference. Speech and Image Technologies for Computing and Telecommunications (Cat. No.97CH36162)*, vol. 1, pp. 231–234,1997.

[109] M. Chandrashekar, U. Kumar, and K. Reddy, "FPGA implementation of high speed infrared image enhancement," *International Journal of Electronic Engineering Research Volume 1 Number 3, pp. 279–285, 2009*.

[110] A. M. Alsuwailem and S. A. Alshebeili, "A new approach for real-time histogram equalization using FPGA," in *2005 International Symposium on Intelligent Signal Processing and Communication Systems*, pp. 397–400, 2005.

[111] S. Sowmya and R. Paily, "FPGA implementation of image enhancement algorithms," in *2011 International Conference on Communications and Signal Processing*, pp. 584–588, 2011.

[112] T. M. Bittibssi, G. I. Salama, Y. Z. Mehaseb, and A. E. Henawy, "Image Enhancement Algorithms using FPGA." 8th International Computer Engineering Conference (ICENCO), 2012

[113] N. Sachdeva, N. Sachdeva, and T. Sachdeva, "An FPGA Based Real-time Histogram Equalization Circuit for Image Enhancement. International Journal of Electronics & Communication Technology vol. 1, issue 1, 2010.

[114] D. Dhanasekaran and K. Boopathy Bagan, "High Speed Pipelined Architecture for Adaptive Median Filter," *European Journal of Scientific Research ISSN*, vol. 145029, no. 4, pp. 454–460, 2009.

[115] T. Jain, P. Bansod, C. B. S. Kushwah, and M. Mewara,

"Reconfigurable Hardware for Median Filtering for Image Processing Applications," in *2010 3rd International Conference on Emerging Trends in Engineering and Technology*, pp. 172–175, 2010.

[116] Y. Li, Q. Yao, B. Tian, and W. Xu, "Fast double-parallel image processing based on FPGA," in *Proceedings of 2011 IEEE International Conference on Vehicular Electronics and Safety*, pp. 97–102, 2011.

[117] F. Fons, M. Fons, and E. Canto, "Approaching Fingerprint Image Enhancement through Reconfigurable Hardware Accelerators," in *2007 IEEE International Symposium on Intelligent Signal Processing*, pp. 1–6, 2007.

[118] Luxin Yan, Tianxu Zhang, and Sheng Zhong, "A DSP/FPGA - Based Parallel Architecture for Real-time Image Processing," in *2006 6th World Congress on Intelligent Control and Automation*, pp. 10022–10025, 2006.

[119] M. Alluru, I. S. Ahn, and Y. Lu, "Discrete cosine transform(DCT)-based reconfigurable system design," in *2011 IEEE International Conference On Electro/Information Technology*, pp. 1–4, 2011.

[120] H. GholamHosseini and S. Hu, "A High Speed Vision System for Robots Using FPGA Technology," in *2008 15th International Conference on Mechatronics and Machine Vision in Practice*, pp. 81–84, 2008.

[121] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, p. 28–31 Vol.2, 2004.

[122] "A self-adaptive Gaussian mixture model," *Computer Vision and Image Understanding*, vol. 122, pp. 35–46, May 2014.

[123] F. R. Dalgleish, F. M. Caimi, W. B. Britton, and C. F. Andren, "An AUV-deployable," pp. 1–5, 2007.

[124] L. Stutters, H. Liu, and C. Tiltman, "Navigation technologies for autonomous underwater vehicles," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 38.4: 581-589, 2008.*

[125] A. Jasper, “Oil/Gas Pipeline Leak Inspection and Repair in Underwater Poor Visibility Conditions: Challenges and Perspectives,” *Journal of Environmental Protection*, vol. 3, pp. 394–399, 2012.

[126] D. Kim and S. Hong, “Multiple-target tracking and track management for an FMCW radar network,” *EURASIP Journal on Advances in Signal Processing*, no. 1, pp.159, 2013.

[127] B. Ekstrand, “Some aspects on filter design for target tracking,” *Journal of Control Science and Engineering*, vol.10, 2012.

[128] K. Daniilidis, C. Krauss, M. Hansen, and G. Sommer, “Real-time tracking of moving objects with an active camera,” *Real-Time Imaging*, vol. 3, 1998.

[129] S. A. Christe, M. Vignesh, and A. Kandaswamy, “An efficient FPGA implementation of MRI image filtering and tumor characterization using Xilinx system generator,” *arXiv preprint arXiv:1201.2542* (2012) .

[130] F. Vahid, *Digital design, with RTL design, VHDL, and Verilog*. Wiley Book, 2011.

[131] V. I. Haltrin, “Absorption and scattering of light in natural waters,” in *Light Scattering Reviews*, Springer Berlin Heidelberg, pp. 445–486, 2006.

[132] A. T. Çelebi and S. Ertürk, “Visual enhancement of underwater images using Empirical Mode Decomposition,” *Expert Systems with Applications*, vol. 39, no. 1, pp. 800–805, 2012.

[133] S. Corchs and R. Schettini, “Underwater image processing: State of the art of restoration and image enhancement methods,” *Eurasip Journal on Advances in Signal Processing*, vol. 2010, 2010.

[134] Xiaoying Fang, Jingao Liu, Wenquan Gu, and Yiwen Tang, “A method to improve the image enhancement result based on image fusion,” in *2011 International Conference on Multimedia Technology*, pp. 55–58, 2011.

[135] B. Yang, Z. Jing, and H. Zhao, “Review of pixel-level image fusion,” *Journal of Shanghai Jiaotong University (Science)*, vol. 15, no. 1, pp. 6–12, Feb. 2010.

[136] H. Kawamura, S. Yonemura, J. Ohya, and A. Kojima, “Gray-world-assumption-based illuminant color estimation using color gamuts with high and low chroma,” *Color Imaging: Displaying, Processing, Hardcopy, and Applications*, vol. 8652, p. 86520C, 2013.

[137] W.-Y. Hsu and C.-Y. Chou, “Medical Image Enhancement Using Modified Color Histogram Equalization,” *Journal of Medical and Biological Engineering*, vol. 35, no. 5, pp. 580–584, Oct. 2015.

[138] C. O. Ancuti and C. Ancuti, “Single Image Dehazing by Multi-Scale Fusion,” *IEEE Transactions on Image Processing*, vol. 22, no. 8, pp. 3271–3282, Aug. 2013.

[139] T. Mertens, J. Kautz, and F. Van Reeth, “Exposure Fusion: A Simple and Practical Alternative to High Dynamic Range Photography,” *Computer Graphics Forum*, vol. 28, no. 1, pp. 161–171, Mar. 2009.

[140] A. Moreno, B. Fernando, B. Kani, S. Saha, and S. Karaoglu, “Color Correction: A Novel Weighted Von Kries Model Based on Memory Colors,”, *CCIW* 11, pp. 165–175, 2011.

[141] I. Jang, K. Park, and Y.-H. Ha, “Color Correction by Estimation of Dominant Chromaticity in Multi-Scaled Retinex,” *Journal of Imaging Science and Technology*, vol. 53, no. 5, p. 50502, 2009.

[142] D. L. Ruderman, D. L. Ruderman, T. W. Cronin, and C.-C. Chiao, “Statistics of Cone Responses to Natural Images: Implications for Visual Coding,” *Journal Of The Optical Society Of America A*, vol. 15, pp. 2036--2045, 1998.

[143] L. A. Torres-Méndez and G. Dudek, “Color Correction of Underwater Images for Aquatic Robot Inspection". EMMCVPR 2005.

[144] Y. Li, H. Lu, and S. Serikawa, “Underwater Image Devignetting and Colour Correction,” Springer, Cham, pp. 510–521, 2015.

[145] G. Bianco, M. Muzzupappa, F. Bruno, R. Garcia, and L. Neumann, “A New Color Correction Method For Underwater Imaging.” *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 40.5, 2015.

[146] A. Jain, “Fundamentals of digital image processing.” Prentice Hall, 1989.

[147] E. D. Pisano *et al.*, "Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms.," *Journal of digital imaging*, vol. 11, no. 4, pp. 193–200, Nov. 1998.

[148] M. Moniruzzaman, M. Shafuzzaman, and M. F. Hossain, "Multiple regions based histogram equalization," in *2014 International Conference on Informatics, Electronics & Vision (ICIEV)*, pp. 1–6, 2014.

[149] D. J. Jobson, Z. Rahman, and G. A. Woodell, "A multiscale retinex for bridging the gap between color images and the human observation of scenes," *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 965–976, Jul. 1997.

[150] D. H. Choi, I. H. Jang, M. H. Kim, and N. C. Kim, "Color Image Enhancement Using Single-Scale Retinex Based On An Improved Image Formation Model." *Signal Processing Conference, 2008 16th European*. IEEE, 2008.

[151] X. Wen, W. Yuling, and Z. Weiqing, "Sonar Image Processing System for an Autonomous Underwater Vehicle(Auv)," *OCEANS'95. MTS/IEEE. Challenges of Our Changing Global Environment. Conference Proceedings.*. Vol. 3. IEEE, pp. 1883–1886, 1995.

[152] S. W. Perry, "Applications of Image Processing to Mine Warfare Sonar." No. Dsto-Gd-0237. Defence Science and Technology Organisation Melbourne (Australia), 2000

[153] M. Brandner and A. Pinz, "Real-Time Tracking of Complex Objects Using Dynamic Interpretation Tree,", pp. 9–16, 2002.

[154] I. Kavasidis and S. Palazzo, "Quantitative performance analysis of object detection algorithms on underwater video footage," in *Proceedings of the 1st ACM international workshop on Multimedia analysis for ecological data - MAED '12*, p. 57, 2012.

[155] J. Yang, W. Yang, and M. Li, "An efficient moving object detection algorithm based on improved GMM and cropped frame technique," in *2012 IEEE International Conference on Mechatronics and Automation*, pp. 658–663, 2012.

[156] R. Zhang and J. Ding, "Object tracking and detecting based on

adaptive background subtraction," *Procedia Engineering*, vol. 29, pp. 1351–1355, 2012.

[157] J. Heikkilä and O. Silvén, "A real-time system for monitoring of cyclists and pedestrians," *Image and Vision Computing*, vol. 22, no. 7, pp. 563–570, Jul. 2004.

[158] M. S. C. Almeida and L. B. Almeida, "Blind and Semi-Blind Deblurring of Natural Images," *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 36–52, Jan. 2010.

[159] Z. Ye, T. Curran, and D. Lemon, "Fish detection by the acoustic scintillation technique," *ICES Journal of Marine Science*, vol. 53, no. 2, pp. 317–321, Apr. 1996.

[160] D. Mukherjee, Q. M. J. Wu, and Thanh Minh Nguyen, "Gaussian Mixture Model With Advanced Distance Measure Based on Support Weights and Histogram of Gradients for Background Suppression," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1086–1096, May 2014.

[161] M. Genovese and E. Napoli, "ASIC and FPGA Implementation of the Gaussian Mixture Model Algorithm for Real-Time Segmentation of High Definition Video," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 3, pp. 537–547, Mar. 2014.

[162] T. Long, W. Jiao, G. He, and W. Wang, "Automatic Line Segment Registration Using Gaussian Mixture Model and Expectation-Maximization Algorithm," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 5, pp. 1688–1699, May 2014.

[163] C. Su and A. Amer, "A Real-Time Adaptive Thresholding for Video Change Detection," in *2006 International Conference on Image Processing*, pp. 157–160, 2006.

[164] P. S. Shubham Lavania, "Novel Method for Weed Classification in Maize Field Using Otsu and PCA Implementation | DeepDyve," in *International Conference on Computational Intelligenece and Communication Technologies*, pp. 534–540, 2015.

[165] A. Balasuriya and T. Ura, "Autonomous target tracking by underwater robots based on vision," in *Proceedings of 1998 International Symposium on Underwater Technology*, pp. 191–197,

1998.

[166] P. Zingaretti and S. M. Zanoli, “Robust real-time detection of an underwater pipeline,” *Engineering Applications of Artificial Intelligence*, vol. 11, no. 2, pp. 257–268, Apr. 1998.

[167] W.-M. Tian, “Integrated method for the detection and location of underwater pipelines,” *Applied Acoustics*, vol. 69, no. 5, pp. 387–398, May 2008.

[168] B. T. Chacko and S. Shelly, “Real-time video filtering and overlay character generation on FPGA,” *ITC 2010 - 2010 International Conference on Recent Trends in Information, Telecommunication, and Computing*, pp. 184–189, 2010.

[169] K. Kunvar, S. A. Patil, and S. A. More, “Real time application to generate the differential time lapse video with edge detection,” in *2012 Nirma University International Conference on Engineering (NUiCONE)*, pp. 1–6, 2012.

[170] G. T. Shrivakshan and C. Chandrasekar, “A Comparison of various Edge Detection Techniques used in Image Processing.” *International Journal of Computer Science Issues* 9.5: 272-276, 2012.

[171] N. Nagata and T. Maruyama, “Real-time detection of line segments using the line Hough transform,” in *Proceedings. 2004 IEEE International Conference on Field- Programmable Technology (IEEE Cat. No.04EX921)*, pp. 89–96, 2004.

[172] chiou Y S, C.-L. Wang, and S.-C. Yeh, “Reduced-complexity scheme using alpha–beta filtering for location tracking,” *IEE proceedings. Communications.*, vol. 5, no. 13, pp. 1806–1813, Sep. 1994.

[173] M. Vinaykumar and R. K. Jatoth, “Performance evaluation of Alpha-Beta and Kalman filter for object tracking,” in *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, pp. 1369–1373, 2014.

[174] S. Sharma, S. Deshpande, and K. M. Sivalingam, “Alpha-beta filter based target tracking in clustered wireless sensor networks,” in *2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011)*, pp. 1–4, 2011.

Subject Index