Underwater Object Detection Using Image Processing

Syed Safdar Hussain,

National University of Sciences and Technology,

Islamabad,

Pakistan.

November 2014

Contents

List of Figures

List of Tables

Abstract

In this research, An algorithm has been developed and implemented in embedded system, which can detect underwater water man made object using prediction technique. For this purpose strength lines are detected using edge detection technique. After detection of lines Prediction algorithm using Gaussian Mixture Model predict the appropriate location of underwater object. Two different embedded systems PSoC and FPGA are also test for the implementation of the detection and prediction technique. Cypress PSoC 3, Cypress PSoC 5, Xilinx Virtex 6 are tested in this research. PSoC can only perform well for image enhancement technique but for rest of the algorithm it fails as it has very limited resources and it is not able to preform complex multiplication and exponential operation. By Using line detection kernel, detection and prediction algorithms are implemented in Virtex 6 FPGA. FPGA based algorithm predict the presence of underwater object by ratio proportion technique. Whole algorithm from edge detection to object prediction are implemented in FPGA by using Xilinx system development for DSP.

Chapter 1

Introduction

Underwater surveillance and navigation is normally based on sonar technology [1], but in many cases image processing is needed rather then sonar. Underwater remote operating vehicles such as PAP and ROV are using image processing based system [2–4]. The dynamics of underwater vehicles required some embedded system, which will able to control and operate vehicle with some algorithm and technique. This research is focused on the implementation of algorithm and technique on embedded system, which can detect some man made object in under water environment. For detection underwater man made object line detection technique is used for the development of under water object prediction algorithm. For the implementation of prediction algorithm two different embedded system Xilinx Field Programmable Gate Array (FPGA) and Cypress Programmable System on Chip (PSoC) has been tested.

1.1 Research Aim and Objectives

1.1.1 Aims

This research is focus on the development of technique which not only deals with the under water noisy image, but also try to detect man-made object and able to predict the most probable position of object. Secondly, this is also aim of research to develop such type of algorithm which is able to implement is embedded system.

1.1.2 Objective

Objective of this research is based on three different category.

- Develop some mathematical model which can prediction or detection underwater man made object.
- For system implementation, optimize mathematical model up to a level so that it can be implement in some embedded system.
- Implement algorithm in PSoC or FPGA.

1.2 Contributions per Chapters

1.2.1 Contribution of Chapter 4

Chapter 4 is based on proposed technique which is able to detect man-made object in underwater environment. Comparison between different preprocessing filters and object detection technique is discussed. It is also mention that, how prediction is more effective than detection in underwater object classification.

1.2.2 Contribution of Chapter 5

Chapter 5 describe how image enhancement algorithms can be implemented in reconfigurable hardware. It is also describe how affectively algorithm can be implemented by efficient utilization of logical blocks. This chapter describe how much Cypress PSoC is able to deal with image processing in terms of reconfigurable computing.

1.2.3 Contribution of Chapter 6

Chapter 6 describe the implementation of image processing technique, which predict the presence of underwater man-made object in embedded system FPGA. This chapter describe the efficient way of implementation of image processing technique by using Xilinx system generator for DSP by utilization of reconfigurable hardware FPGA.

1.3 Organization of Thesis

In this thesis, chapter 2 covers the literature reviews based on the concept of different object detection technique. Chapter 3 covers the basis detail of embedded system used for implementation of image processing technique. Chapter 4 is describe the simulation based implementation of underwater man made object detection technique. Chapter 5 and 6 describe the different options for the implementation of technique discussed in chapter 3 in embedded systems.

Chapter 2

Literature Review

2.1 De-Blurring Technique

2.1.1 Wiener Filter

Wiener filter [5] is use to reduce blurriness and noise up to minimal level, it is done by minimizing error function by estimating $\hat{f}$ by using equation

$$e^2 = E\{(f - \hat{f})^2\} \tag{2.1}$$

where E, is expected value operator and f is un-degraded image. $f_r(x, y)$ is formed from degraded image $g(x, y)$ by using wiener filter [5]. Computation of wiener filter is depend on power spectral density of image and noise.

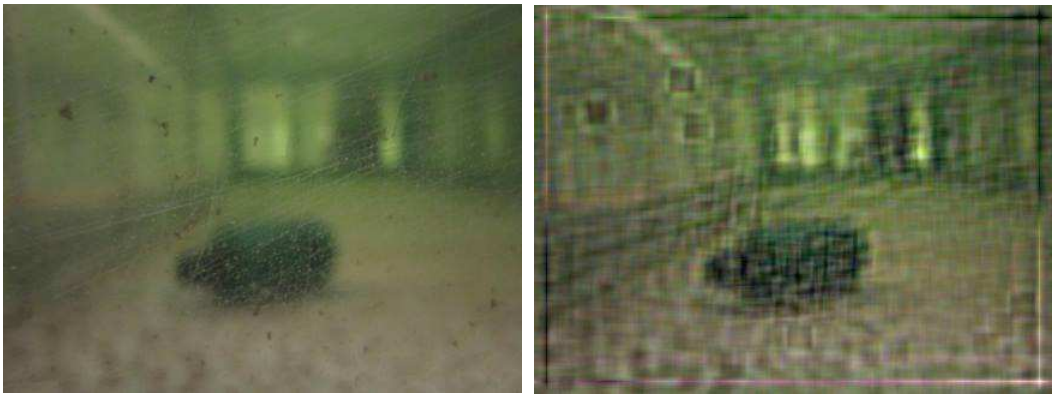
$$H_W(u, v) = \frac{S_{aa}(u, v)}{S_{aa}(u, v) + S_{nn}(u, v)} \tag{2.2}$$

where $S_{aa}(u, v)$ is power spectral density of image and $S_{nn}(u, v)$ is power spectral density of noise, also u and v are coordinate value of pixels. During de-

convolution, finding out appropriate signal to noise ratio is important, signal to noise ratio is calculated by using equation

$$SNR_E = \frac{NoiseVar_E}{ImageVar} \tag{2.3}$$

Where SNR_E is estimated signal to noise ratio and $NoiseVar_E$ is estimated noise variance.



(a) Original underwater image (b) Image after Wiener filtering

Figure 2.1: Wiener Filtering using Gaussian kernel as PSF

2.1.2 LucyRichardson deconvolution

This technique is based on iterative method [6, 7]. According to its algorithm, if p_{ij} is point spread function,where j is light producing location and i is observed location at image, then observed value d_i at location of pixel i can calculated as

$$d_i = \sum_j p_{ij} u_j \tag{2.4}$$

By using Poisson distribution u_j can calculated, it is value of pixel at location j. Iteratively u_j can obtain by

$$u_j^{(t+1)} = u_j^{(t)} \sum_i \frac{d_i}{c_i} pij \tag{2.5}$$

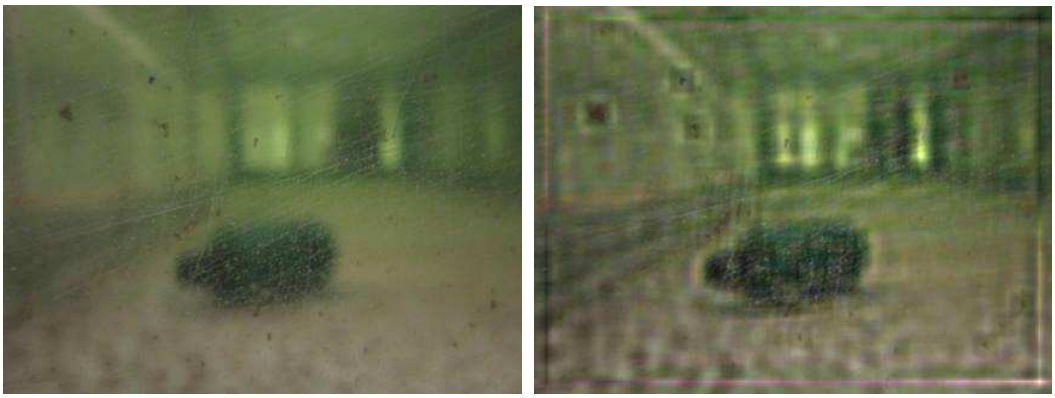
$$c_i = \sum_j p_{ij} u_j^{(t)} \tag{2.6}$$

For more general problem u_j can be written in term of convolution

$$u^{(t+1)} = u^{(t)}. \left(\frac{d}{u^(t) * p} * \hat{p} \right) \tag{2.7}$$

$\hat{p}$ is flipped point spread function and can expressed as

$$\hat{p}_{nm} = p_{(i-m)(j-m)}, 0 \leq n, m \leq i, j \tag{2.8}$$



(a) Original underwater image (b) Image after LR deconvolution

Figure 2.2: LR deconvolution using Gaussian kernel as PSF

2.1.3 Blind deconvolution

There are several method of blind deconvolution technique [8] which include, Iterative blind deconvolution method, Simulated Annealing method and NSA-RIF

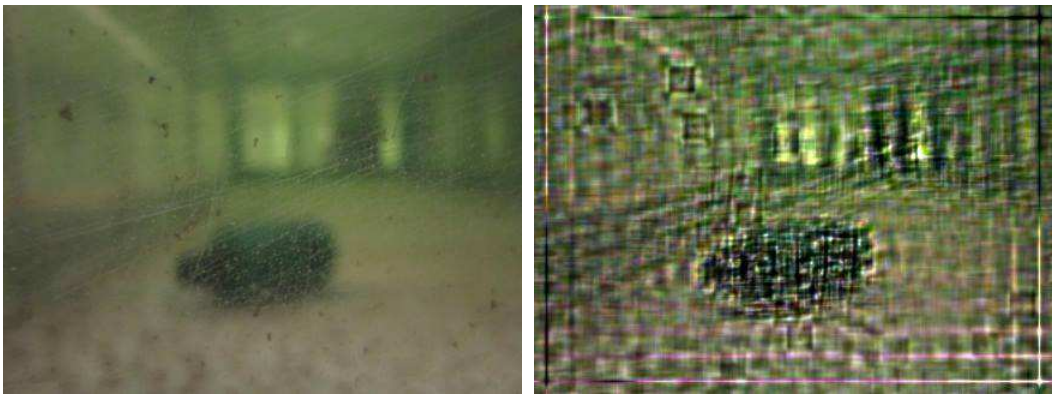
method [9]. For Iterative method Wiener like system response is use for reduce blurriness in image [9], so in frequency domain it can be achieve by $G(u,v) = F(u,v).H(u,v)$ where PSF in frequency domain can be estimated as

$$H_k(u,v) = \frac{G(u,v)F^*_{k-1}(u,v)}{|F_{k-1}(u,v)|^2 + \alpha/|H_{k-1}(u,v)|^2} \tag{2.9}$$

here α is representing noise level parameter. Similarly for $F_k(u,v)$

$$F_k(u,v) = \frac{G(u,v)H^*_{k-1}(u,v)}{|H_{k-1}(u,v)|^2 + \alpha/|F_{k-1}(u,v)|^2} \tag{2.10}$$

In Simulated Annealing method [10], De blurring is achieving by minimizing the cost function. In non negativity and support constraints recursive inverse filtering (NSA-RIF) [11], iterative updating of coefficient of inverse filter is use to achieving result close to original image.



(a) Original underwater image (b) Image after blind deconvolution

Figure 2.3: Blind deconvolution using Gaussian kernel as PSF

2.1.4 Histogram Equalization

For enhancement of object shape in image, contrast adjustment is consider to be more process effective method. It can be done by histogram equalization of

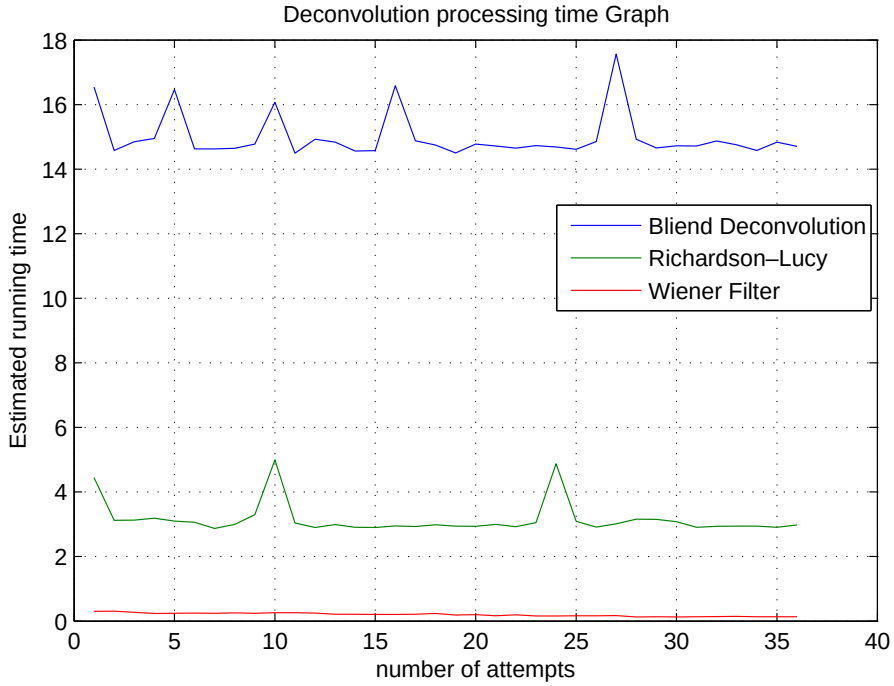


Figure 2.4: Graph show time taken in second by different de-blurring techniques in 36 attempts

image [12]. This process improves average intensity and contrast of image. If p_x is consider as probability of occurrence of any gray level pixel then

$$p_x(i) = p(x = i) = \frac{n_i}{n}, 0 \leq i < L \tag{2.11}$$

where L is highest value of pixel, and for this research it's value is 255 and n is total number of values. Cumulative density function is calculated by equation

$$cdf_x(i) = \sum_{j=0}^{i} p_x(i) \tag{2.12}$$

Then histogram equalization of an image is calculated by generalize equation using cfd of each particular values

$$h(v) = \lfloor \frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1) \rfloor \tag{2.13}$$

Figure 2.5 shows histogram equalization of underwater image. This technique is more computationally feasible then wiener filter as mentioned in Figure 2.6

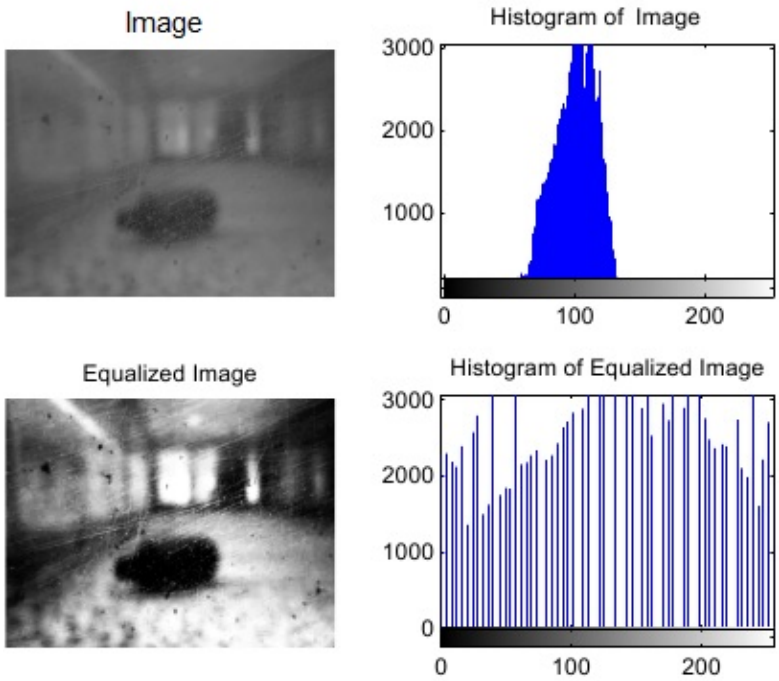


Figure 2.5: Histogram equalization of gray level image

2.2 Object Detection

2.2.1 SIFT

SIFT is develop by David G. Lowe for image feature generation [13]. SIFT detect object in targeted image by taking four basic step, these are Scale-space extrema detection, Keypoint localization, Orientation assignment, and Keypoint descriptor [14]. SIFT extract distinctive invariant feature from images, these feature are use for object recognition.

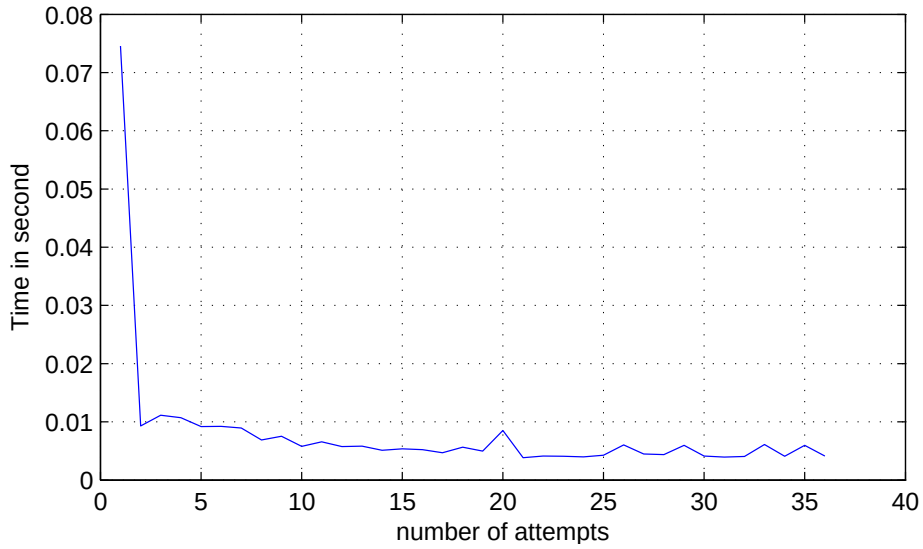


Figure 2.6: Graph show time taken in second by histogram equalization in 36 attempts

Scale-space extrema detection

In this technique Difference of Gaussian is taken for identifying interest point in image, those points are invariant to scale and orientation [14]. Initially image is convolved with Gaussian function.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.14}$$

where Gaussian function with scale σ is given by following equation

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{2.15}$$

so difference of Gaussian of image can be calculated by equation

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{2.16}$$

Keypoint localization

In this step contrast points are rejected and edge responses are eliminated, principle curvature is also computed by using Hessian matrix [14].

Orientation assignment

Invariance of image rotation is achieved by orientation assignment. 36 bins orientation histogram was form to covering 360 degree range of orientation. Gradient magnitude $m(x, y)$, and orientation $\theta(x, y)$ is precomputed for each image sample $L(x, y)$ [14].

$$Q_1 = L(x + 1, y) - L(x - 1, y) \tag{2.17}$$

$$Q_2 = L(x, y + 1) - L(x, y - 1) \tag{2.18}$$

$$m(x, y) = \sqrt{Q_1^2 + Q_2^2} \tag{2.19}$$

$$\theta(x, y) = tan^{-1} \frac{L(x + 1, y) - L(x - 1, y)}{L(x, y + 1) - L(x, y - 1)} \tag{2.20}$$

Keypoint descriptor

It is based on 16 * 16 patches with 4 * 4 subregions, each subregion has 8 bins, so there are 4*4*8 =128 dimension in total [14].

2.2.2 SURF

SURF creates stack and filter, SURF use box filter approximation of second-order Gaussian partial derivatives [15] rather then using DoG as used in SIFT. If $\mathbf{x} = (x, y)$ is a point in an image $\mathcal{I}$, then Hessian matrix $\mathcal{H}$ ($\mathbf{x}$,σ) in $\mathbf{x}$ at scale σ is

define as follows

$$\mathcal{H}(\mathbf{x},\sigma) = \begin{bmatrix} L_{xx}(\mathbf{x},\sigma) & L_{xy}(\mathbf{x},\sigma) \\ L_{xy}(\mathbf{x},\sigma) & L_{yx}(\mathbf{x},\sigma) \end{bmatrix} \tag{2.21}$$

In above matrix $L_{xx}(\mathbf{x},\sigma)$ is convolution sum of second-order Gaussian partial derivatives with image $\mathcal{I}$

$$L_{xx}(\mathbf{x},\sigma) = \frac{\partial^{\mathbf{2}}}{\partial \mathbf{x^2}} \mathcal{G}(\sigma) * \mathcal{I} \tag{2.22}$$

In general SURF depends upon interest points, feature vector and descriptor matched between different images [15].

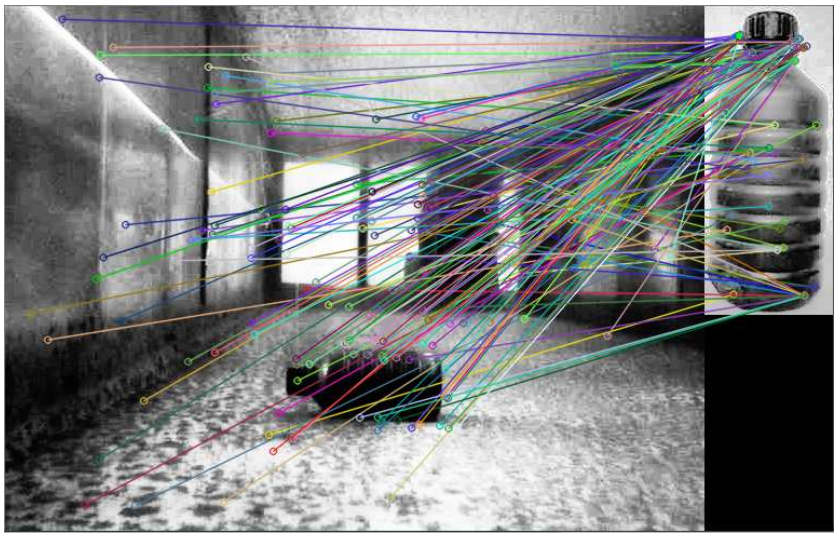


Figure 2.7: SIFT Key flase match problem distract the prediction of object in image

SIFT and SURF generate false matches if number of feature keys are high, either image is matched with right or wrong object as show in Figure 2.7. These technique are totally depend upon database of images of object.

2.3 Contour Detection

Contour can be detected using Canny method [16].In First step of canny method, image $f_r(x, y)$ is smoothed using Gaussian kernel, Secondly Sobel filter is used to find gradient components in image by using given equation.

$$\frac{\partial f_r(x,y)}{\partial x} = \Delta x = \frac{\partial f_r(x+dx,y) - f_r(x,y)}{\partial x} \tag{2.23}$$

$$\frac{\partial f_r(x,y)}{\partial y} = \Delta y = \frac{\partial f_r(x,y+dy) - f_r(x,y)}{\partial y} \tag{2.24}$$

Edge strength can is calculated by $|G| = |G_x| + |G_y|$ and direction is computed by $\theta = tan^{-1}(\frac{G_y}{G_x})$ [16]. In Figure 2.8b boundaries of geometrical shaped objects are visible, by using this image area of different shape can be calculated. By detecting the regularity of close boundaries, it can also be predicted that image contain human made object. By calculating scaling factor of detected object distance can also be measured. Absolute distance can be measured by using given formula.

$$d_a(x,y) = \sum_{i=1}^{N} |x_i - y_i| \tag{2.25}$$

Euclidean distance can be measured by given equation

$$d_E(x,y) = \sum_{i=1}^{N} \sqrt{{x_i}^2 - {y_i}^2} \tag{2.26}$$

2.4 Boosting

In supervise machine learning classifier has to learn a functional relation F : $f(x) = y$ among input argument x and output of function y [17]. Boosting is

(a) Natural object

(b) Man made object

Figure 2.8: Contour Detection, contour in man made object is based on lines and arcs while there is irregularity in natural object.

supervise machine learning technique in which many weak classifier are combine to perform powerful task [18]. As week classifier is computationally inexpensive but, smart integration of some week classifier may perform batter then many monolithic classifier like artificial neural networks [18].

2.4.1 Adaptive Boosting

Adaptive Boosting (AdaBoost) is a meta Algorithm and is use in combination with other machine learning technique to improve it performance [19]. There are different form of AdaBoost implication like LogitBoost, Discrete AdaBoost, Gentle AdaBoost, and Real AdaBoost [17].

The standard two-class Discrete AdaBoost algorithm is based on N training example $(x_i, y_i)1N$ with $x_i \in R^K, y_i \in -1, +1$.

In first step weight are assigned

$$W_i = 1/N \tag{2.27}$$

$i = 1, ..., N$. Secondly those weight W_i on training data, are used to fit the classifier

$$f_m(x)\epsilon - 1, 1 \tag{2.28}$$

$m = 1, ..., m$. Error err are is computed as

$$err_m = E_w[1_{(y \neq f_m(x))}] \tag{2.29}$$

$$c_m = log((1 - err_m)/err_m) \tag{2.30}$$

Set weight W_i as

$$W_i \Leftarrow W_i exp[c_m 1_{(y \neq f_(x_i))}] \tag{2.31}$$

$i = 1, 2, ..., N$, and it will be normalize by $\sum iw_i = 1$. Finally new sample x is classify as

$$sing(\sum m = 1 M c_m f_m(x)) \tag{2.32}$$

The implication of other three boosting techniques (LogitBoost, Gentle AdaBoost, and Real AdaBoost) are also same structure as Discrete AdaBoost as discussed above.

2.4.2 Haar Feature-based Cascade Classifier for Object Detection

One of the novel example of haar training is “Rapid object detection using a boosted cascade of simple features”.This technique is develop by Viola and Jones

[20], and it is further enhance by Rainer Lienhart [21] . It is a machine learning approach for robust and rapid object detection. This technique is based on three key contribution.

- Introduction of image using "$IntegralImage$"for rapid feature computation using detector.
- "$AdaBoost$"base learning algorithm for selection of small number of visual features.
- Cascaded structure of classifier

The cascaded classifier is combination of several classifiers, first it train with desire shapes of object, called positive example and then with scaled size of object called negative example. Once classifier is trained it will be apply in image, it returns 1 if region has object, and return 0 otherwise. The technique is to move classifier search window to image in every location. Another aspect of classifier is, it can resize it self according to the scale of the object, rather then resizing image [21].

Chapter 3

Proposed Embedded System for Image Processing

3.1 Introduction

For utilizing mathematical equation in real life environment, it is important to implement it in some system. For dynamics of this research, embedded systems are used to implement algorithm. This chapter describe the basic introduction and working of two different types of embedded systems which are consider for this research.

3.2 Cypress Programable System on Chip(PSoC)

PSoC not only provides digital configurable blocks but also CPU core and analog configurable blocks [22], which increase the flexibility of implementation of algorithm. Because amount of digital blocks is limited, it is required smartly

distribution of arithmetic and logical operation among core and digital block to achieve maximum utilization of resources [23].

Programmable System on Chip (PSoC) is embedded system develop by Cypress [24]. PSoC is integration of electronic component including programmable logic device(PLD) based logic elements, memory, 8051 or ARM micro controller [25], and also configurable analog component in a single embedded chip [22]. There are variants of PSoC, which include PSoC1, PSoC 3, PSoC4, PSoC5, and PSoC5LP [24].

3.2.1 PSoC 3, 5, and 5LP

PSoC is based on MCU subsystem, programmable routing and interconnects, and digital and analog configurable blocks [22].

MCU subsystem

MCU is based on SRAM, EEPROM, flash memory, interrupt controller, switch mode oscillator, and CPU Core [25]. Basic different between PSoC 3 and PSoC 5 is core processor. PSoC 3 is based on 8051 micro controller which is work on single cycle per instruction [25]. PSoC5 and PSoC5LP has ARM Cortex-M3 based CPU core [22]. For communication PSoC also provides some communication interfaces which includes I^2C, CAN 2.0, serial wire debug (SWD), USB 2.0, and JTAG [25]. All communication interfaces are part of MCU subsystem.

Programmable Routing and Interconnects

With the help of programmable routing interconnects facility input and output devices can be connected with any of the provided pins. Unidirectional or bidirectional configuration can be set with the help of programming to any pin.

Digital and Analog Configurable block

The most distinguish facility provided in PSoC and configurable blocks [25]. Digital configurable blocks contain PLD based units called UDB [24]. These blocks can utilize for system on chip programming. Digital block also provide facility of UART, PWM, and SPI [22]. For analog system implementation analog configurable block can be use. Analog block contain switch capacitor, DAC, ADC, digital filter, and operational amplifier blocks [24].

3.3 Field Programmable Gate Array ($FPGA$)

FPGA is the combination of Logical units (LUT) and reconfigurable interconnection for dynamic routing. For this research Xlinix Spartan3, Spartan6 and Virtex6 FPGA are tested. For the implementation of line detection and object prediction algorithm Virtex6 is considered because of it provides more DSP slices then Spartan 3 and 6. DSP slices are utilize for complicated signal and image processing algorithm for example convolution.

3.3.1 Xilinx System Generator for DSP

All image processing and object prediction algorithms are implemented in FPGA by using Xilinx System Generator for DSP. With the help of this facility developer not only program FPGA with the help of Matlab and Verilog but also design algorithm module by module in Matlab simulinks. Xilinx provide many predefined blocks of complex mathematical and logical operations which and place and route in simulinks for the the development of algorithms.

3.3.2 Conclusion

The detail of implementation and testing of algorithm in embedded systems are discuss in Chapter 5 and 6. PSoC and not so much affective for image processing technique but FPGA perform very well and in some cases it perform better than system simulation.

Chapter 4

Prediction of Underwater Object using Line Detection

Under water imaging is primarily focused on search and rescue, underwater mine detection, underwater cable and pipeline overhauling and underwater geological survey. Main challenge in underwater imaging is blurriness, Underwater blurriness is caused by many factor which includes microscopic organism, impurities and density of water which effects refractive index of water, and bokeh which is blur effect on those region of image that are out of focus in range. Picture of moving object also have blur effect, reason is motion blur. To detected object in underwater image, integration of different image processing technique has been made. It includes preprocessing to reduced blurriness and noise effect in image and Euclidean shape prediction by detecting lines in image. This research also discuss computationally feasible technique which is not only independent of image data bank but also less time consuming to process.

4.1 Introduction

This research is focused on the implementation of proper technique to predict the presence of regular shape object in underwater environment. For this task different preprocessing and object detection technique has been tested.Underwater environments are very hazy, so iterative and adaptive filter based technique for example blind deconvolution, Richardson-Lucy and wiener filter is failed to properly deblur image. It is also observed that, detection of object in blur image with the help of available object detection technique like scale invariant feature transformation SIFT [14] and speeded up robust feature SURF [15] is not much effective because feature information is diminished because of blurriness, as a result probability of false features matching is increase.

Blurriness can be remove by taking convolution of effected image with appropriate PSF. But in uncertain condition like underwater environment, prediction of efficient PSF is ambiguous and convolution of image with obtainable PSF is not completely removed blurriness and noise. At this point contrast adjustment technique like histogram equalization [26] is quite effective but the chance of false feature matching is still there. To prevent from false matching localized rather than generalized method is more effective. Rather then implements feature-matching technique, regular shape detection in preprocessed image has done in two different steps. Edges of regular shape object have detected in first step, in second step classifier predict the presence of close form object shape with the help of lines detection.

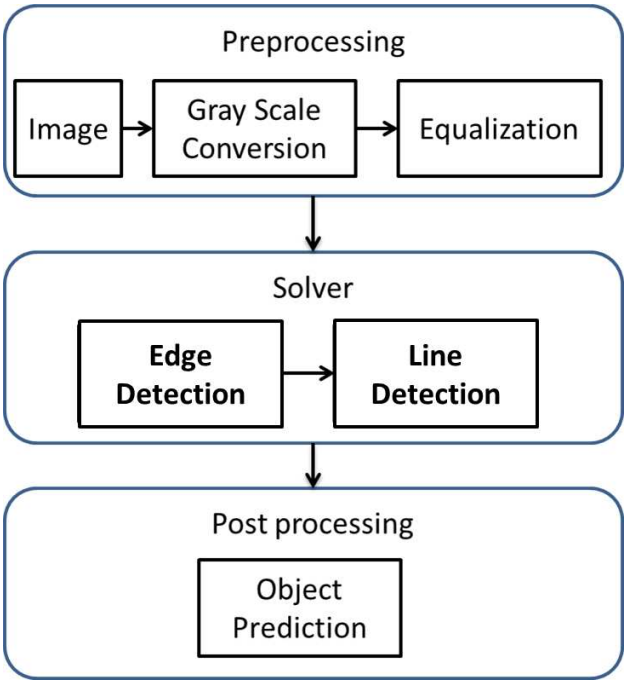


Figure 4.1: State diagram

4.2 Preprocessing

Blurriness can be modeled by convolution of original image $f(x;y)$ with some specific system response $h(x;y)$ called point spread function (PSF) [27]. To exhibit image according to real environment, addition of noise $n(x;y)$ is another important factor in the blur image.

$$g(x;y) = f(x;y) * h(x;y) + n(x;y) \tag{4.1}$$

Blurriness hide key information of features in image. Blurriness can be reduced by convolving blurred image $g(x,y)$ with PSF $h(x,y)$, as a result image $f_r(x,y)$ with close resemblance to original image $f(x,y)$ can be achieved by equation $f_r(x,y) = g(x,y) * h(x,y)$. By taken Fourier transformation of equation $F_R(u,v)$

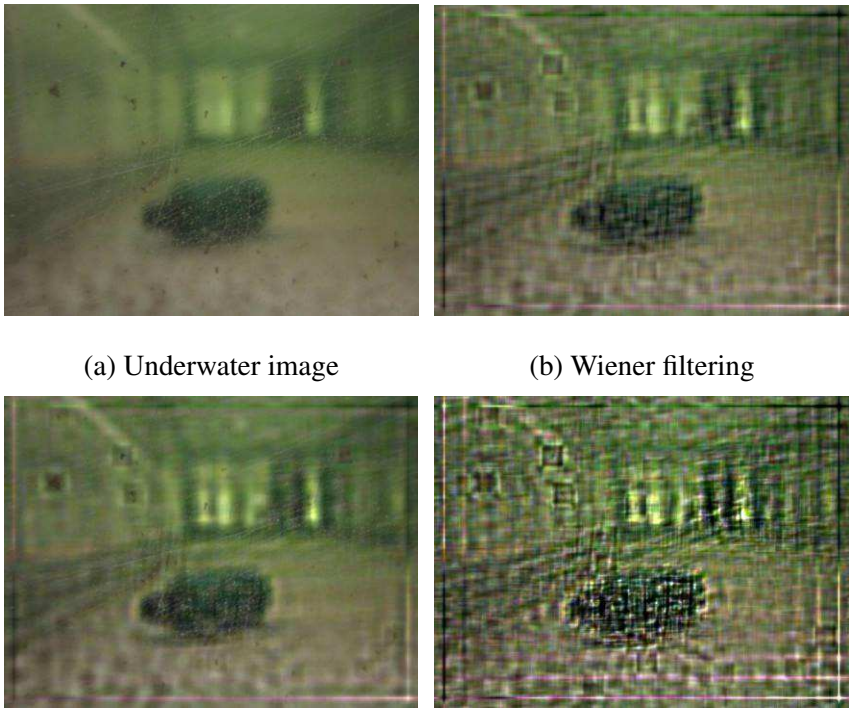
can be computed.

$$F_R(u, v) = G(u, v).H(u, v) \tag{4.2}$$

where $H(u, v)$ is modulation transfer function (MTF) [28]. Noise is second factor which can be remove with the help of filter.

4.2.1 De-Blurring Using Filter

By using Wiener filter [5], blurriness and noise is reduced up to minimal level,



(a) Underwater image (b) Wiener filtering

(c) L-R deconvolution (d) blind deconvolution

Figure 4.2: Images After different deconvolution using Gaussian kernel as PSF

4.2.2 Histogram Equalization

For enhancement of object shape in image, contrast adjustment is consider to be more process effective method. It can be done by histogram equalization of image [12]. This process improves average intensity and contrast of image.

(a) Blur image (b) Gray Image (c) Equalized Image

Figure 4.3: Histogram Equalization of Underwater image is more effected then other de-blurring technique if PSF is unknown

4.3 Edge and Line Detection Detection

In this paper prediction rather then detection technique is considered. Detection of object like SIFT and SURF is depend upon image of targeted object, SIFT is develop by David G. Lowe for image feature generation [14]. SIFT detect object in targeted image by taking four basic step, these are Scale-space extrema detection, Keypoint localization, Orientation assignment, and Keypoint descriptor [14]. SIFT extract distinctive invariant feature from images, these feature are use for object recognition. SURF creates stack and filter, SURF use box filter approximation of second-order Gaussian partial derivatives [15] rather then using DoG as used in SIFT. SIFT and SURF generate false matches if number of feature

keys are high, either image is matched with right or wrong object these technique are totally depend upon database of images of object. Man made closed form object has regular shapes, by detecting basic elements of those shape which are based on line [29], presence of objects can be predicted

4.3.1 Edge Detection

Edge can be detected by $2D$ convolution of image with Sobel, Robert Cross or Prewitt Operator. These technique is also used in Canny edge detection [16] technique, but underwater image is already blur so canny which use Gaussian filter, makes image more hazy and it also use more computational time, so best choice is Sobel Operator Figure 4.4 show difference in edge detection between canny and Sobel. It is first order kernel use for edge detection, it can be done by convolving image with Sobel kernel. For horizontal operation $K1$ kernel is use, similarly for vertical operation transpose of $K1$ that is $K2$ kernel is use . After convolving Image with both kernels, approximation of gradient of image is computed.

$$K1 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad K2 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Robert Cross Operator (modified sobel operator): Robert Cross modified sobel operator is relatively more sensitive to noise and in Embedded system it take less computational cycle for multiplication operating then original sobel operator. Also it output processed image more sharp then conventional sobel. Image generated by Robert Cross operator is more close to binary matrix, so less computation

will applied for binary matrix generation.

$$K1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \; K2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Convolved image with $Fx[x, y]$ can be formed by convolving image $I[x, y]$ with sobel or Robert Cross kernel $K1[x, y]$ for vertical operation as show in equation 4.3, similarly $Fy[x, y]$ is formed by convolving image $I[x, y]$ with $K2[x, y]$ for horizontal operation show in equation 4.4.

$$F_x = K_1[x, y] * I[x, y] = \sum_{i=0}^{m} \sum_{j=0}^{n} K_1[i, j] I[x - i, y - j] \tag{4.3}$$

$$F_y = K_2[x, y] * I[x, y] = \sum_{i=0}^{m} \sum_{j=0}^{n} K_2[i, j] I[x - i, y - j] \tag{4.4}$$

Sum of squares of Fx and Fy will formed final image having sharp edge information.

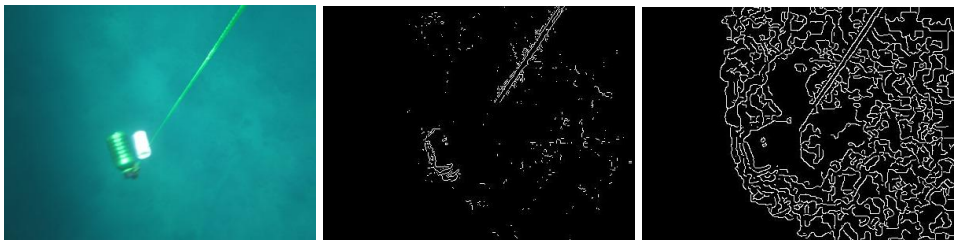
$$S[x, y] = \sqrt{(F_x[x, y])^2 + (F_y[x, y])^2} \tag{4.5}$$

4.3.2 Line Detection

Hough transform [30] are widely use for detecting line. Once edge has been found using edge detection technique next step is to detect line. Concentration of line will decide either there is a chance of presence of regular shaped objects in an image. To represent line two parameter are used , let a and b are parameters for line the line can be represent by

$$y = ax + b \tag{4.6}$$

(a) Image 1 (b) After Sobel (c) After Canny



(d) Image 2 (e) After Sobel (f) After Canny

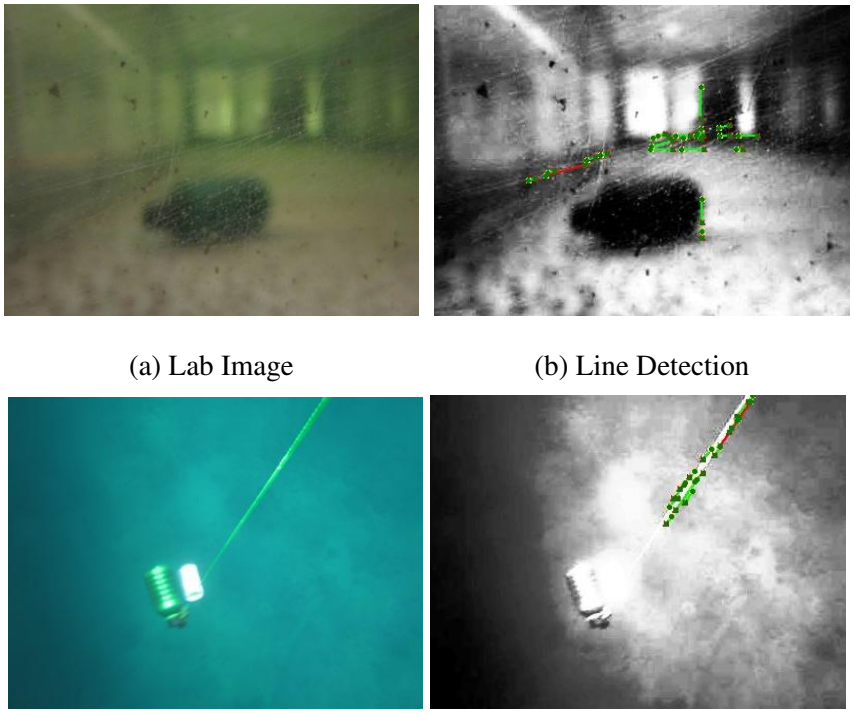
Figure 4.4: Results shows Canny operation provides extra and non linear edges then Sobel kernel

But above equation can not valid for line with any angle so for general representation if angle θ and distance r is also consider then

$$r = x.cos(\theta) + y.sin(\theta) \tag{4.7}$$

$$y = \frac{r}{sin(\theta)} - \frac{cos(\theta)}{sin(\theta)}.x \tag{4.8}$$

where $r \geq 0$ and $0 \leq \theta \leq 360$. θ and r are dimension of hough space. With the help of these dimension single line can be represent by a point in hough space. In this research hough transform is use for line detection. As much line are detected in image, probability of presence of close form object is also increase. Figure 4.5 show line are detected around close form object in lab based and ocean based environment.



(a) Lab Image (b) Line Detection

(c) Ocean Image (d) Line Detection

Figure 4.5: Line detection of underwater image in lab and ocean environment

4.4 Object Prediction

Once lines are detected, Coordinated value of the end point of each lines will be use for the prediction purpose. As lines are the basic entity of geometrical shaped object, so it coordinate value will be use to locate the position of object.

4.4.1 Averaging

Very basic way to fine out position of object using coordinated values is averaging. This method will gives the single coordinate value (x_o, y_o) pointing towards the center of whole population and can be calculated as follow.

$$x_o = \frac{1}{n}\sum_{i=0}^{n} x_i \tag{4.9}$$

$$y_o = \frac{1}{m}\sum_{j=0}^{m} y_j \tag{4.10}$$

where n is total number of x coordinate values and m is total number of y coordinate values. Red circle in figure 4.7 is showing the average position of underwater object, it is calculated by using data of coordinate values show in figure in 4.6a.

4.4.2 Gaussian Mixture Model (GMM)

After detection the coordinate values of two end points of all lines present in image, GMM is used to predict the position of possible presence of men made object in underwater image. It is done by training of data using iterative Expectation-Maximization (EM) [31] to calculate the parameter of GMM. GMM can be esti-

mated by sum of all weighted Gaussian densities components,

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{M} w_i g(\mathbf{x}|\mu_i, \sum_i) \tag{4.11}$$

where $\mathbf{x}$ is n-dimensional data and in this research $n = 2$, M is total number of densities components . n-variate Gaussian function can be formed by

$$g(\mathbf{x}|\mu_i, \sum_i) = \frac{1}{(2\pi)^{\frac{n}{2}} |\sum_i|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(\mathbf{x} - \mu_i)' \sum_i^{-1} (\mathbf{x} - \mu_i)\} \tag{4.12}$$

where GMM parameters are vector of mean μ_i, covariance matrix $\sum_i$ and mixture weights w_i. Collectively parameters can be express as $\lambda = \{w_i, \mu_i, \sum_i\} \quad i = 1, 2, ..., M$. It is also satisfies that

$$\sum_{i=1}^{M} w_i = 1 \tag{4.13}$$

There are several method to compute the GMM parameter [32]. One of them is Maximum Likelihood(ML) method [31], which can use for estimation of GMM parameterλ.

$$p(\mathbf{X}|\lambda) = \prod_{t=1}^{T} p(\mathbf{x}_t|\lambda) \tag{4.14}$$

where T is total number of sequence and X is vector of training samples $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$ ML can be obtain by multiple iteration of EM algorithm [31].

Now after detecting edge using Sobel and line using Hough transformation as show in figure 5.13b, coordinates values of end points of all lines are used by GMM to predict the position of object in image. Figure 4.6a shows the coordinates values of lines detected from underwater image given in figure 5.13a. Prediction can be done by using mean vector μ_i, which is generated during the process of EM in GMM. Figures 4.6b and 4.6c shows the detected points for most probable

locations of object. Green circles in figure 4.7 are showing locations predicted by algorithm. Log likelihood during the prediction process using image given in figure 5.13a for 9,67, and 3 iterations are -369.548, -393.608, and -369.965 respectively.

4.4.3 Averaging of Gaussian Mixture Model

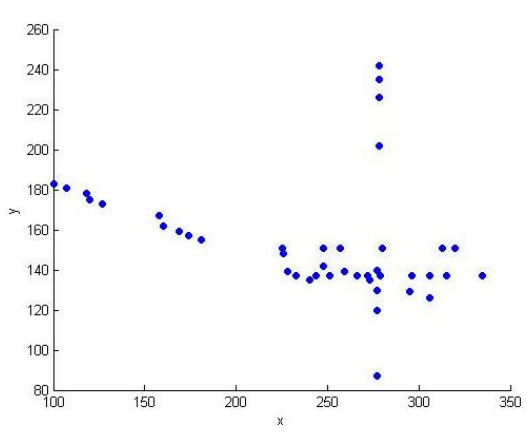
It can be computed by taken the average of coordinate values of position of object predicted by GMM.

$$\phi(x_p, y_p) = \frac{1}{n} \sum_{i=1}^{n} \phi(x_i.y_i). \quad n = 1, ..., N. \tag{4.15}$$

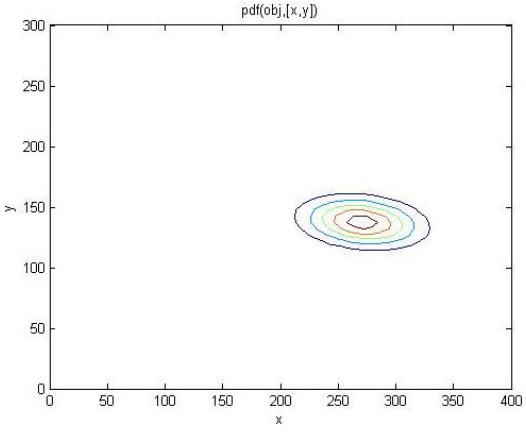
Here N = 3.It is also observed that if $N \rightarrow$ to ∞ then $\phi(x_p, y_p) \rightarrow$ to $Average(x_o, y_o)$. Blue circle in Figure 4.7 pointing the position of object detected by average GMM.

4.5 Conclusion

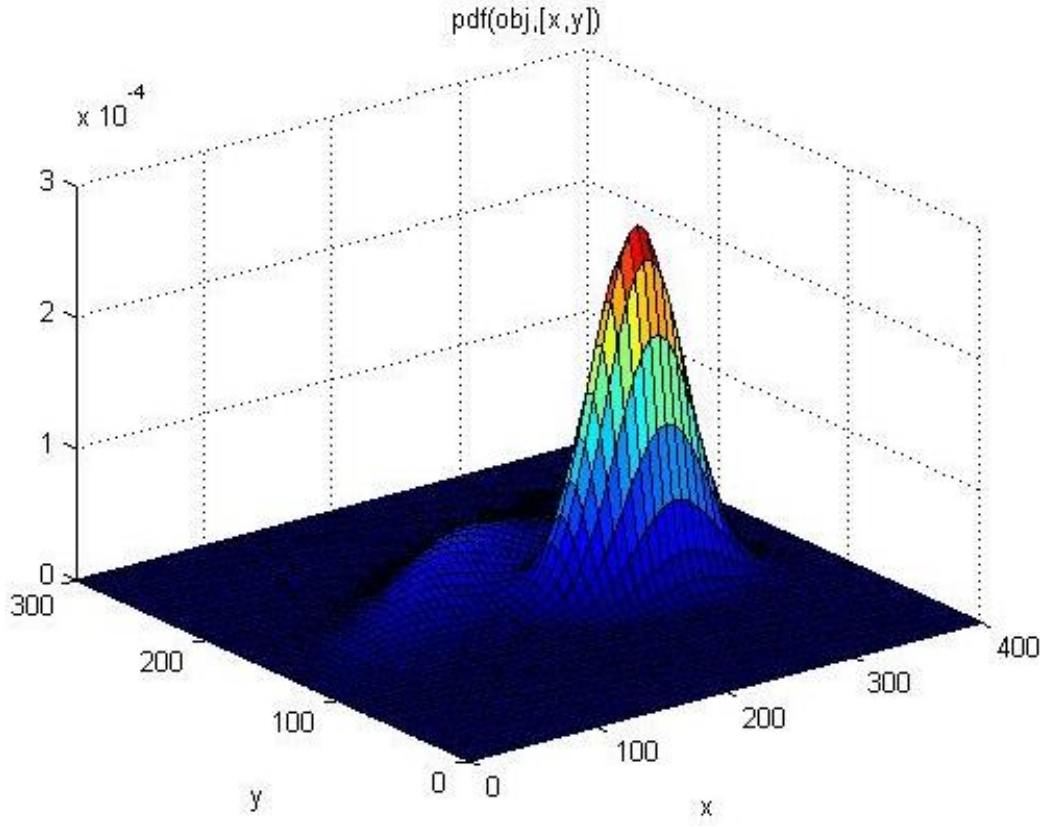
For object detection in underwater image, there are three basic challenge blurriness, false feature and distortion of shape. In this research histogram equalization is consider for preprocessing and blur removing, sobel operator is used for edge detection and hough transform is for line detection purpose. Concentration of numbers of line is bench mark for the presence of man made close form object.



(a) Coordinate value of lines



(b) Likelihood of object



(c) Likelihood of data with *pdf*

Figure 4.6: Prediction of the possible location of object in image

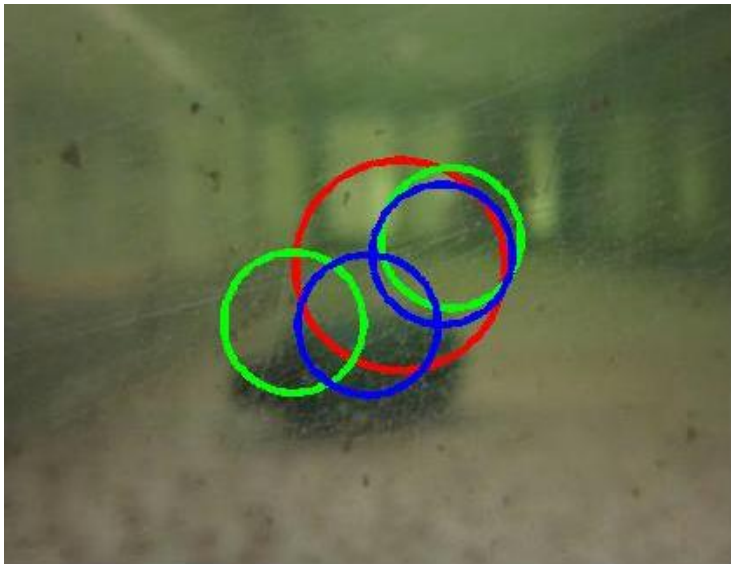


Figure 4.7: Prediction of Underwater object in Lab Environment

4.6 Results

Results of experiment are given in figure 4.8 to 4.12, which shows different step of experiments form edge detection to object prediction. Table in figure 4.1 have different number of iteration and its respected $Log - likelihood$ on the bases of that position of object is predicted.

Figure 4.8: Image Taken in Lab as wall as in Ocean Environment for Test

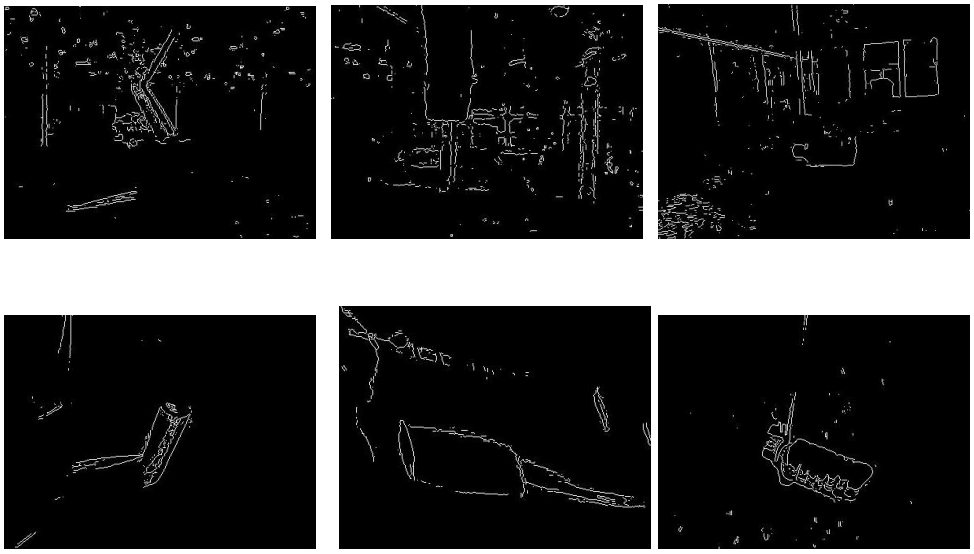


Figure 4.9: Edge detection using given images

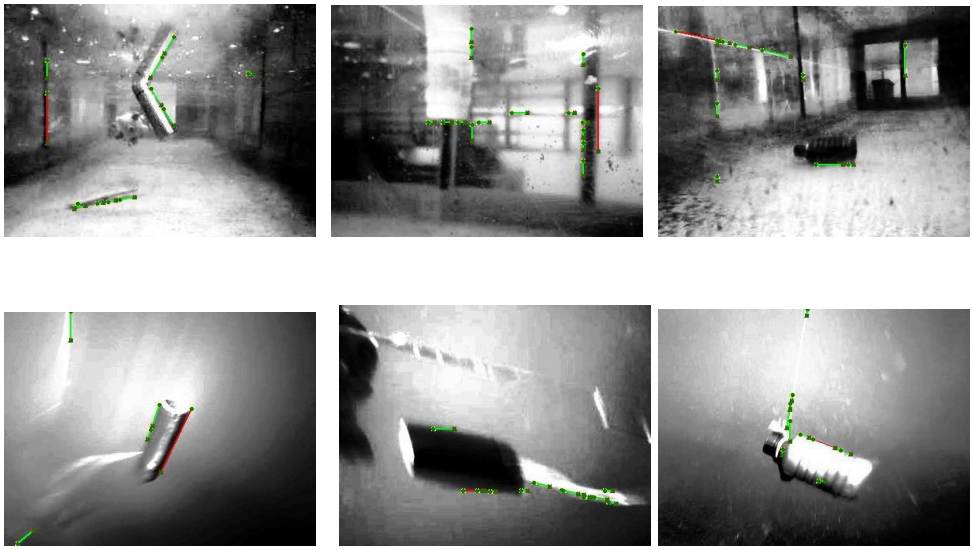


Figure 4.10: Lines detected by Hough Transformation with edge detected images

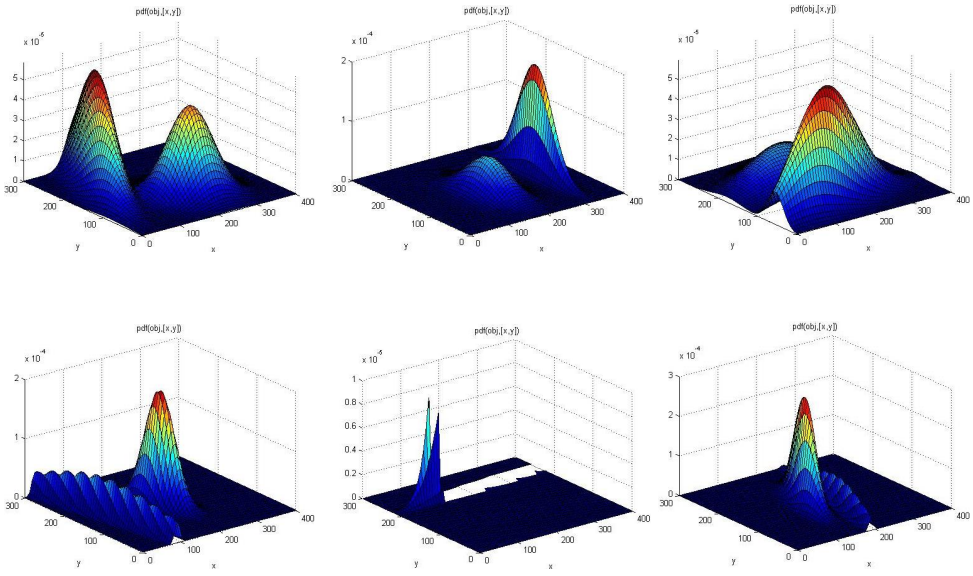


Figure 4.11: Prediction of possible position in image

$image(a)$		$image(b)$		$image(c)$	
i	L	i	L	i	L
14	-264.286	8	-317.206	27	-234.414
36	-262.265	9	-305.505	8	-266.699
21	-262.265	20	-309.498	26	-234.414
$image(d)$		$image(e)$		$image(f)$	
i	L	i	L	i	L
6	-86.3851	19	-208.823	23	-178.985
14	-94.6354	19	-208.823	22	-161.84
10	-99.6191	19	-208.823	14	-190.036

Table 4.1: I represent total number of iteration and L represent $Log-Likelihood$

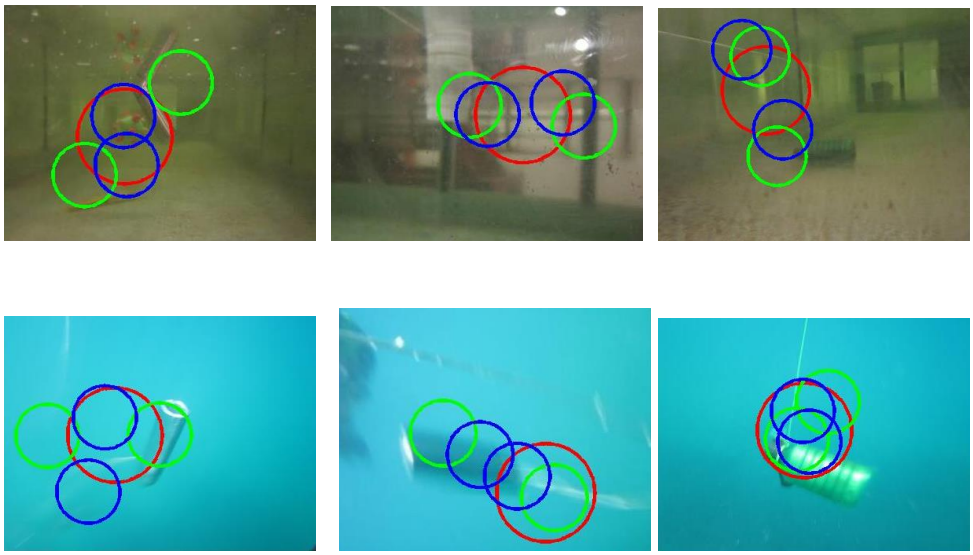


Figure 4.12: Red,Green, and Blue circles show the possibility of object in underwater image using $averaging$,GMM, and $AverageGMM$ respectively.

Chapter 5

Image Enhancement Using PSoC

5.1 Introduction

In this chapter four simple but very effective image enhancement technique is implemented in digital configurable block of PSoC5 LP. This is done by converting 2D gray image matrix to 1D vector and send it to PSoC5 LP Universal Asynchronous Receiver Transmitter (UART). UART is communicating with Verilog component which use Universal Digital Block Array (UDB) for implementing image enhancement algorithm. For implementing enhancement algorithm very simple Verilog components are develop by using Cypress PSoC Creator 2.2, It includes adder, subtracter, multiplier, 8 bit comparator, and multiplexer. Verilog components for constant values are also develop. Enhancement algorithms are implemented with the integration of arithmetic, logical and constant Verilog components.

5.2 Design flow for Image Processing with $PSoC\,5LP$

This paper is focus on the use of Universal Digital Block Array(UDB) for the implementation of four image enhancement algorithm. It is done by implementation of algorithm by using PLD based Verilog components in PSoC5LP [33]. PSoC 5LP is configurable embedded system, which has ARM Cortex base core [24]. Code written in Verilog are directly synthesized in UDB PLDs [23], while code written in C language in PSoC creator will use CPU core [34]. Figure5.1 show transmitting and receiving data to and from Verilog component to external system. UART is used to sends and receive data through serial port. UART component is provided in PSoC Creator as show in Figure 5.2. Resources used by UART is given in Table 5.1.

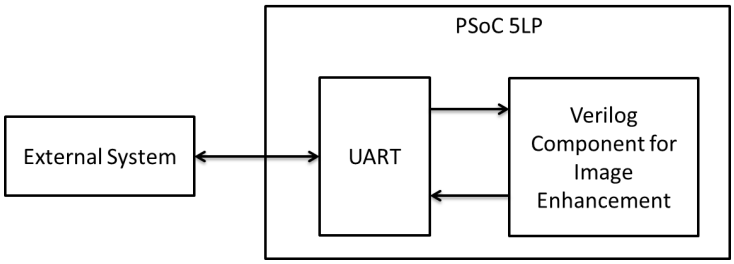


Figure 5.1: Design Flow for the implementation of Image Enhancement Algorithm in PSoC 5LP.

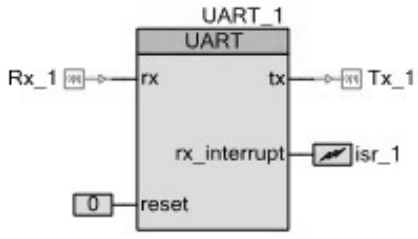


Figure 5.2: UART component provided in PSoC Creator.

Resource Type	Used	Free	Max	% Used
Digital Clock dividers	1	7	8	12.50%
Pins	5	65	70	7.14%
UDB Macrocells	25	167	192	13.02%
UDB Unique Pterms	42	342	384	10.94%
UDB Datapath Cells	3	21	24	12.05%
UDB Status Cells	2	22	24	8.33%
UDB Control Cells	1	23	24	4.17%
PLDs	12	36	48	25.00%

Table 5.1: Resources used by PSoC 5LP in the implementation of UART

5.3 Image Preprocessing and Post-processing

Before sending to PSoC5 LP, gray image has to be preprocessed. Gray image is a 2D matrix and it will have to converted into 1D vector before send to UART in PSoC 5LP. For this purpose Matlab Simulink $Convert2D\ to\ 1D$ reshape block provided by MathWorks [35] is used. Reshape block perform 2D to 1D vector transform. 1D vector is send to PSoC5 LP by using serial port, Matlab $Serial\ Send$ block is used for sending data to PSoC. Figure 6.3 show the preprocessing of input image.

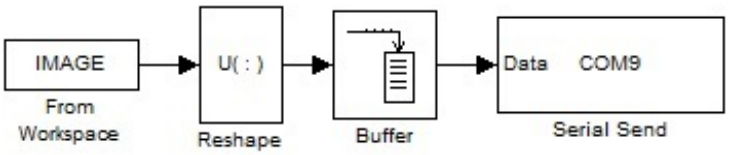


Figure 5.3: Preprocessing unit develop in Matlab Simulink.

After processed through image enhancement algorithm developed by using Verilog components in PSoC5 LP, system receive data from UART. In system, data is receive by $serial\ Receive$ block of Matlab Simulink, from this block data is converted from 1D vector to 2D matrix of gray image by using reshape $Convert\ 1D\ to\ 2D$ block and then save in directory as an image. Figure 5.4 show the post processing of ID vector receive by PSoC5 LP.

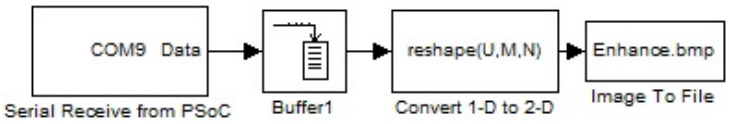


Figure 5.4: Post-processing unit develop in Matlab Simulink.

Image in Figure 5.5 are used as a input image with many other image for this research.

Figure 5.5: Input Gray Scale Image

5.4 Image Enhancement Using Digital Reconfigrable Block

Transformation means change the value of data in signal X by using some transformation function T. Image enhancement is also a form of transformation in which values in image is transform from g to f [36]. If r is any point at $g(x, y)$ then corresponding point s at $f(x, y)$ can be computed as

$$s = T(r) \tag{5.1}$$

In this paper, four simple image enhancements technique is implemented by using transformation function, which is implemented in Verilog component of PSoC 5LP.

Intensity Enhancement

Intensity of image can be enhanced by adding some value c in image so equation 5.1 will become

$$T(r) = r + c \tag{5.2}$$

for gray scale image intensity of image can be changed by using value of $c = 1, ..., 255$ [36]. Algorithm in equation 5.2 is implemented in PSoC5 LP by using add block. This add block as show in figure 5.6 is developed by using Verilog component which can add two eight bits values. In Verilog component $always$ block is used for adding values, $always$ block add values every time when next value from input a is received.

Resources used by intensity enhancement algorithm are show in Table 5.2.

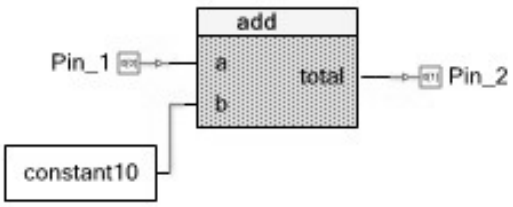


Figure 5.6: PSoC5LP components integrated for intensity enhancement.

Figure 5.7 show the intensity enhanced output after processing of input image given in Figure 5.5 by using Verilog component based intensity enhancement algorithm.

Resource Type	Used	Free	Max	% Used
Pins	5	65	70	7.14%
UDB Macrocells	8	184	192	4.17%
UDB Unique Pterms	2	382	384	0.52%
PLDs	2	46	48	4.17%

Table 5.2: Resources used by PSoC 5LP in the implementation of intensity enhancement algorithm.

Image Negative Transformation

Image negative transformation can be achieve by subtracting each values in image with largest value in image [36]. For gray image it can be done by subtracting value with 255. For implement this technique equation 5.1 will become

$$T(r) = c - r \tag{5.3}$$

Figure 5.7: Intensity enhancement of Image by using c = 50 .

where c = 255. To implement image negative algorithm in PSoC5 LP, $subtract$ block is implement as show in figure 5.8 by using Verilog component. It can subtract two eight bits values every time when new input value is receive. Resources used by this block are show in Table 5.3. Figure 5.9 show the output of image in Figure 5.5 after processed by image negative block.

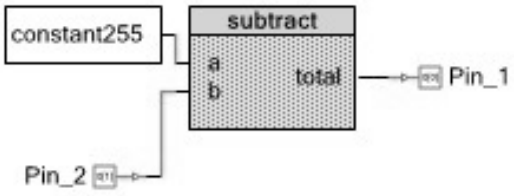


Figure 5.8: PSoC 5LP components integrated for image negative transformation.

Resource Type	Used	Free	Max	% Used
Pins	5	65	70	7.14%
UDB Macrocells	9	183	192	4.69%
UDB Unique Pterms	2	382	384	0.52%
PLDs	3	45	48	6.25%

Table 5.3: Resources used by PSoC 5LP in the implementation of image negative transformation

Figure 5.9: Image negative of input image.

contrast-stretching

For $contrast - stretching$ contrast factor γ is multiply with image value r. For implementing $contrast - stretching$ equation 5.1 will become

$$T(r) = \gamma.(r - \lfloor\frac{c}{2}\rfloor) + \lfloor\frac{c}{2}\rfloor \tag{5.4}$$

where $c = 255$ which is highest value in gray image [36]. Figure 5.10 show the implementation of $contrast - stretching$ Algorithm in PSoC 5LP by using

Verilog component for gray scale image with γ = 1,2,.....20. For implementing this algorithm add, $subtract$, and $mult$ blocks are developed and integrated by using Verilog component. Figure 5.10 shows the integration of components. Table 5.4 show large amount of UDB resources are utilized by $contrast - stretching$ block because of eight bit multiplication operation in $mult$ block. Images shown in Figure 5.11 are processed by using three different value of γ.

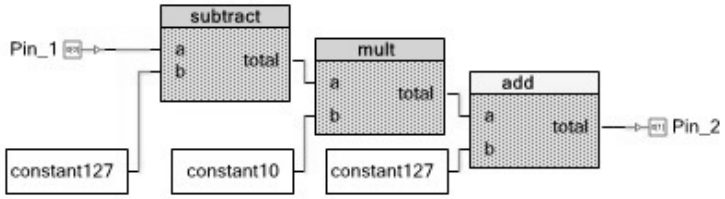


Figure 5.10: PSoC 5LP components integrated for $contrast - stretching$ transformation.

Resource Type	Used	Free	Max	% Used
Pins	5	65	70	7.14%
UDB Macrocells	97	95	192	50.52%
UDB Unique Pterms	67	317	384	17.45%
PLDs	27	21	48	56.26%

Table 5.4: Resources used by PSoC 5LP in the implementation of image negative transformation

(a) γ = 2 (b) γ = 5

(c) γ = 10

Figure 5.11: $contrast - stretching$ transformation of image with different value of γ

Image Thresholding

Image thresholding is achieve by dividing image values in two maximum and minimum points [36]. For gray scale image it is a process of developing binary image. For image thresholding Equation 5.1 will become

$$T(r) = \begin{cases} m_n & \text{for} \quad 0 \leq r \leq c \\ m_x & \text{for} \quad r > c \end{cases} \tag{5.5}$$

where m_n is minimum and m_x is maximum value, here c is decide the segment boundary. Processed threshold image is containing minimum value m_n and maximum value m_x. For implement this algorithm in PSoC 5LP, eight bits comparator and multiplexer is required. Using Verilog component having $if - else$ statement in $always$ block, $compair$ for comparator and $mux2in$ for two input multiplexer are developed. One of each unit is used for implementing binary image thresholding algorithm as shown in Figure 5.12. Table 5.5 shows, very few amount of resources are used, because logical rather then arithmetic operations are taken place in this block. Output images in Figure 5.13 show effect of value of segment boundary c.

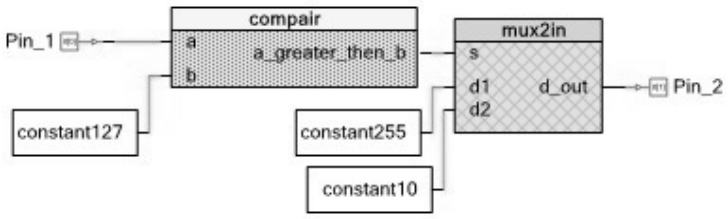


Figure 5.12: PSoC 5LP components integrated for $Image\ Thresholding$.

Resource Type	Used	Free	Max	% Used
Pins	5	65	70	7.14%
UDB Macrocells	1	191	192	0.52%
UDB Unique Pterms	0	384	384	0.00%
PLDs	1	47	48	2.08%

Table 5.5: Resources used by PSoC 5LP in the implementation of image thresholding

(a) c = 150 (b) c = 170

Figure 5.13: Thresholding of image with different value of c

5.5 Conclusion

Image processing is resource consuming task. For embedded system like PSoC5 LP implementation of complex image processing algorithm by using digital configurable block is not feasible. But simple image processing algorithm where arithmetic operation is less required, PSoC will be compatible choice.In this research different image enhancement algorithm are implemented which shows difference in resource utilization during arithmetic and logical operation. In image enhancement $contrast - stretching$ need multiplication operation, experiment show more then 50 % of UDB macrocells are utilized. In other hand in $Image - Thresholding$ task where no arithmetic but only logical operations are required, only 0.5 % of UDB macrocells are used. If arithmetic and logical operation of image processing algorithm are smartly distributed among CPU core which use C language and digital block which use Verilog components respectively then

very effective results can be achieve within the boundary of PSoC 5LP.

Chapter 6

Basic System Implementation Using FPGA

Underwater imaging is use for underwater surveillance and exploration purpose. For reduce the human intervention for underwater exploration special type of vehicle are use, for example PAP and underwater ROV. The dynamics of these vehicle need sophisticated equipment for computation and processing. Image processing in other hand is depends on computationally time consuming operation for example convolution. For handling these challenges FPGA based system is proposed for prediction of man made object in underwater environment. System not only minimizes the blur effect in image but also detect the presence of closed form geometrical shaped object in haze underwater environment. For this purpose contrast equalization, edge detection and concentration of line predictor technique is implemented in FPGA using Xilinx System Generator for DSP.

6.1 Introduction

Image Processing is computationally resource-consuming task. In many environments, resource is limited and task is computationally expansive. For example, air born and underwater platform where dynamics of vehicle is very complicated and installation of effective electronic system is tricky. In such type of environment, embedded systems are very feasible choice. However, for task like machine learning and image processing, use of embedded system needs optimization of resource and compactness of algorithm. Especially in underwater environment where microscopic organism, impurities and density, which affects refractive index of water, cause blurriness in image. Bokeh is another phenomenon; it is blur effect on that region of image, which is out of focus in range. Image of moving object also have blur effect called motion blur. All of these challenges are experiences in underwater ROV during exploration and surveillance because blurriness hides feature information of object in image. Mathematically blur image can be modeled by convolving some PSF (point spired function) [27] with original image. Now consider $I(x, y)$ is original image and it is convolved with some point spired function $p(x, y)$, so blur image $\beta(x, y)$ can be modeled as

$$\beta(x, y) = I(x, y) * p(x, y) \tag{6.1}$$

But in underwater environment noise is also point of consideration because of density and refractive index of water. Another reason of noise in underwater image is microscopic organism. If $\eta(x, y)$ contain all the possible reason for noise in underwater image then by using equation 1, typical underwater image can be

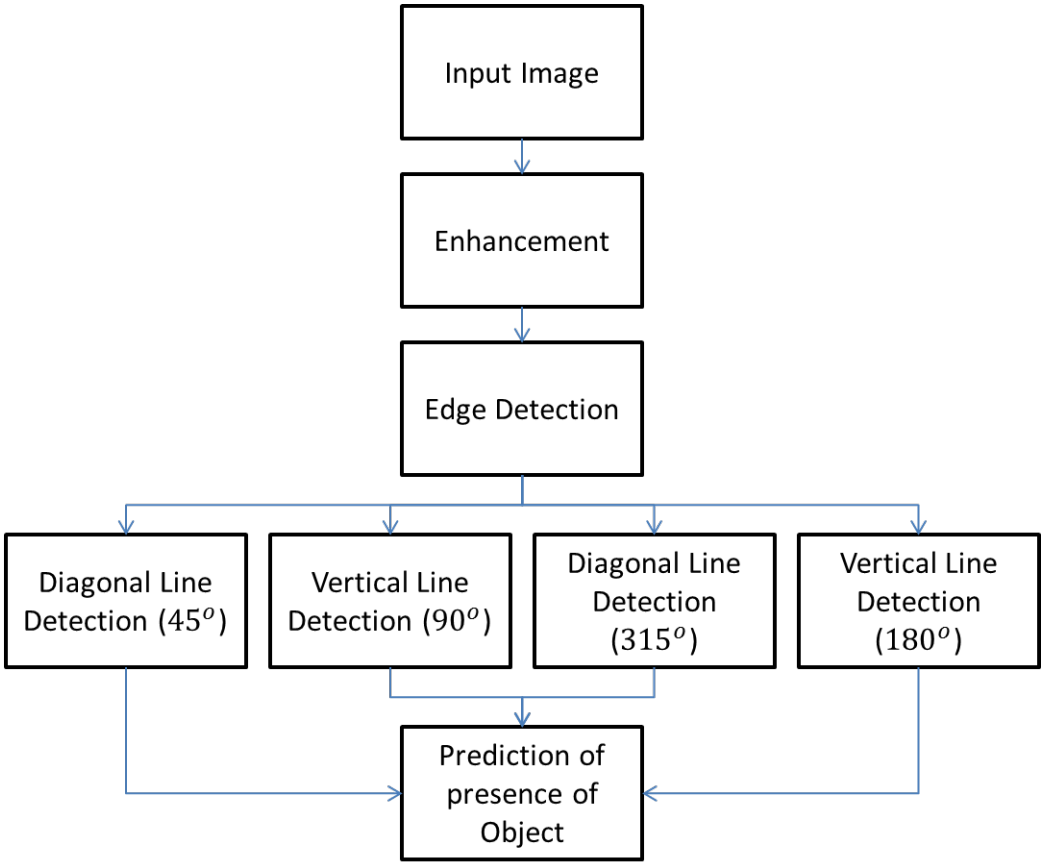


Figure 6.1: System state diagram.

formed as

$$\beta_{noise}(x, y) = \beta(x, y) + \eta(x, y) \tag{6.2}$$

If specific function of PSF and $noise$ is know then removal of blurriness can be achieve by

$$I(x, y) = (\beta(x, y) * p(x.y)) - \eta(x, y) \tag{6.3}$$

Because of randomness in underwater environment prediction of proper PSF [27] is complected. In this paper very simple but effective image enhancement technique is used which is easily implemented in FPGA and consume less resource.

Enhance image is convolved with Sobel kernel to detect edge. In next step vertical, horizontal and diagonal lines are detected. Prediction of object is made by finding out appropriate number of lines with effective size, Figure 6.1 show the state diagram of implemented system.

6.2 Preprocessing

Before sending to FPGA, gray image has to be converted into 1D. For this purpose Matlab Simulink $Convert 2D\ to\ 1D$ reshape block provided by MathWorks [35] is used. Figure 6.3 show the system implemented in Matlab for 1D vector conversion and sender to serial port.

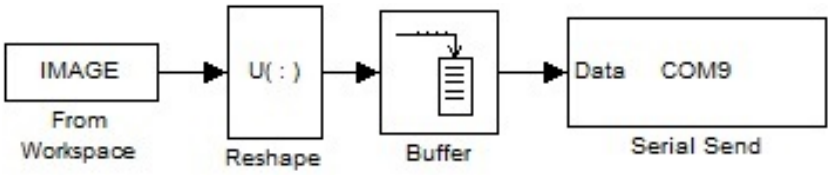


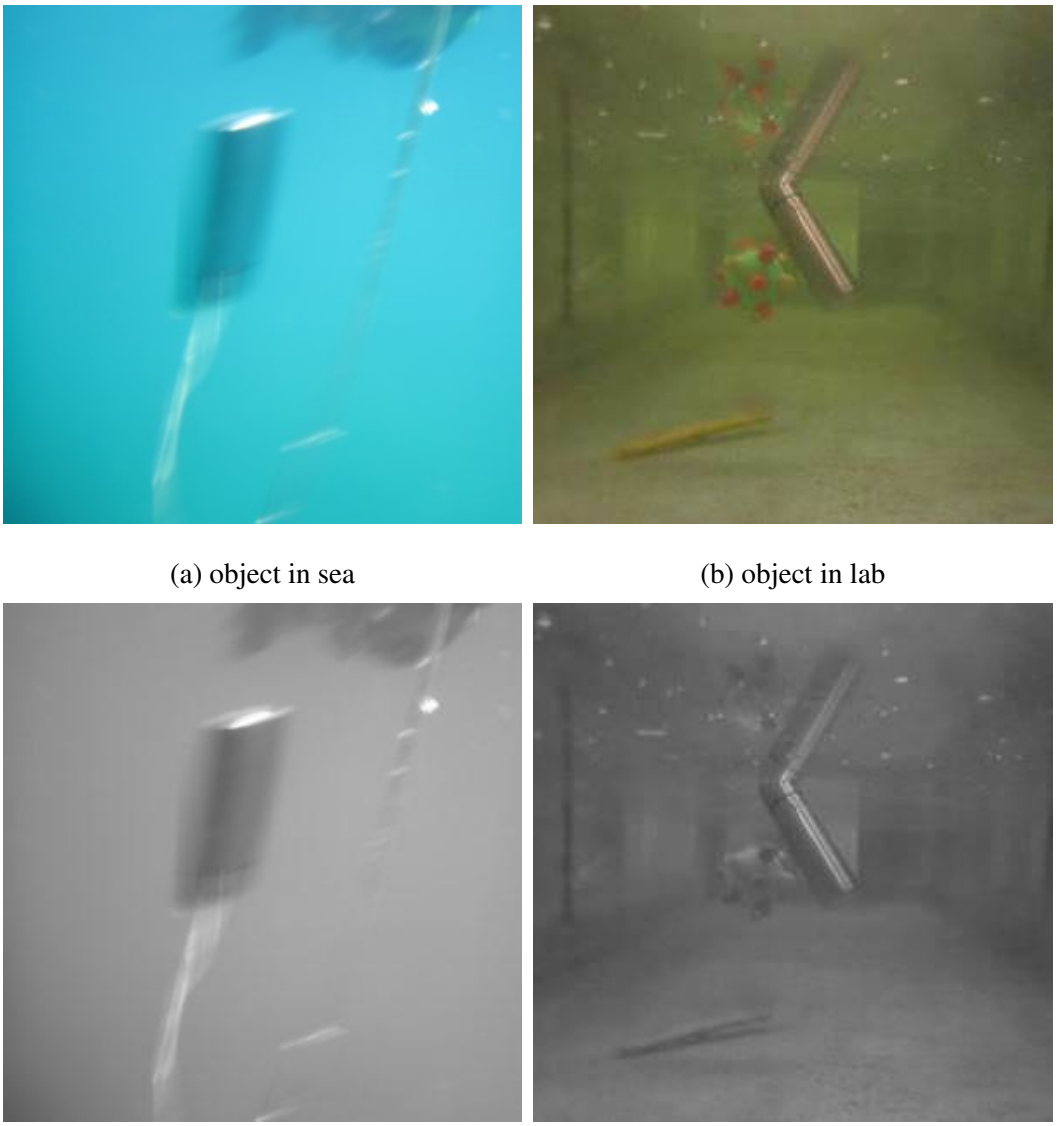
Figure 6.3: Preprocessing unit develop in Matlab Simulink.

In preprocessing unite, dull and noisy underwater image is enhanced by using contrast stretching transformation.

contrast-stretching

Image enhancement is transformation of image value from g to f [36]. If r is any point at $g(x, y)$ then crossponing point s at $f(x, y)$ can be computed as

$$s = T(r) \tag{6.4}$$



(a) object in sea

(b) object in lab

(c) Gray sea image

(d) Gray lab image

Figure 6.2: Image are taken in sea and lab environment for experiment

For this research $contrast - stretching$ is transformed by using contrast factor γ [37].

$$T(r) = \gamma.(r - \lfloor \frac{M_x}{2} \rfloor) + \lfloor \frac{M_x}{2} \rfloor \tag{6.5}$$

where M_x is largest pixel value in image. For gray scale image M_x = 255. Figure 6.4 show the implementation of $contrast - stretching$ transformation in FPGA for gray scale image by using Xilinx System Generator for DSP with γ = 12. Figure 6.5 show resultant images after processed through $contrast - stretching$ module.

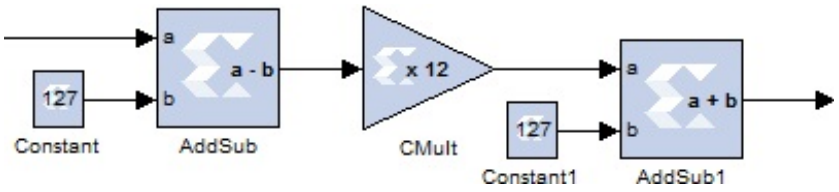


Figure 6.4: $contrast - stretching$ using Xilinx System Generation

6.3 Edge and Line Detection

6.3.1 2D Convolution

$2D$ Convolution of image can be performed by convolving original image $i(x, y)$ with some convolution kernel $k(x, y)$. Consider if image $i(x, y)$ having dimension m and n then $2D$ convolution can be obtain by equation

$$f(x, y) = K(x, y) * i(x, y) = \sum_{j=0}^{m} \sum_{l=0}^{n} K(j, l).i(x - j, y - l) \tag{6.6}$$

(a) (b)

Figure 6.5: Image after $contrast - stretching$ using Xilinx System Generation

6.3.2 Sobel Operator

For detecting edges in image kernel $K1$ and $K2$ will be convolved with image i by using Equation 6.6. In $Sobel$ F_x and F_s is computed by using , $K1$ and $K2$ kernel.

$$K1 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad K2 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Gradient megnitude is fined out by calculating root of sum of square of F_x and F_y, which generate sharp edge information.

$$S(x, y) = \sqrt{F_x(x, y)^2 + F_y(x, y)^2} \tag{6.7}$$

$Filter5x5$ block in Xilinx System Generator is used to find out edges in image by 2D convolution of input image with Sobel operator. Figure 6.6 show combination of $Virtex2\ 5\ LineBuffer$ and $Filter5x5$ blocks use for 2D convolution. Figure

6.7 show the output of image after Sobel operation from system shown in Figure 6.6.

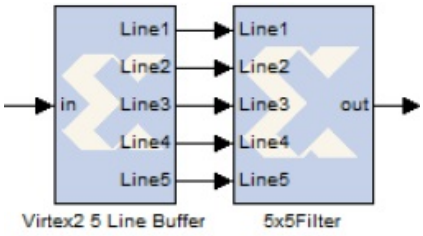


Figure 6.6: 2D convolution using Xilinx system generators Virtex5 line buffer and 5x5 filter kernel

6.3.3 Binary value generation

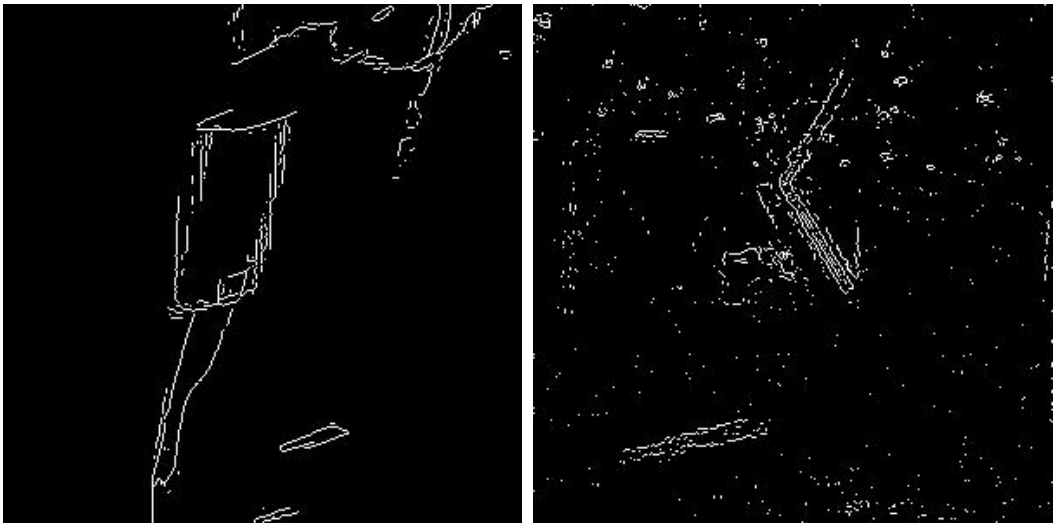
Before send data to line detection module, binary value are generated. Because After Sobel operation image has 255 and values very close to 255 in the region of sharp edges. Region other then sharp edges has values zero or very close to zero. For assigning 1 to those region where sharp edges are formed and rest of the region zero Equation6.8 is used

$$I(r) = \begin{cases} 0 & \text{for} \quad 0 \leq r \leq 255 \\ 1 & \text{for} \quad r > 255 \end{cases} \tag{6.8}$$

Implementation of Equation 6.8 in Xilinx system generator is show in Figure 6.8

6.3.4 Vertical,Horizontal and Diagonal Line Detection

For line detection four operators are used to detect lines of different orientation. Operator H is use for horizontal, V for vertical, P_d for positive diagonal(45^o) and N_d is use for negative diagonal(315^o) lines [38].



(a) (b)

Figure 6.7: Edge Detection using Xilinx System Generation $Filter5x5block$

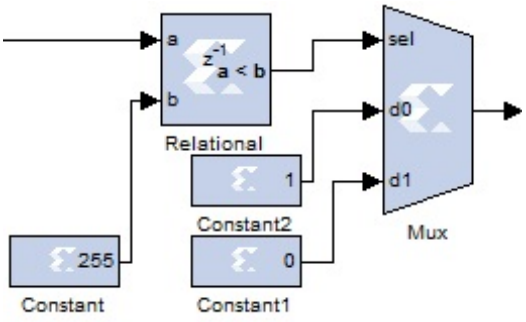


Figure 6.8: Binary Converter by using Xilinx System Generator.

$$H = \begin{bmatrix} -1 & -1 & -1 \\ -2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \qquad V = \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$$

$$P_d = \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \quad N_d = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

These operators is convolved with binary image $i_b(x, y)$ by using Equation 6.6. Four parallel combination of $Virtex2\ 5\ LineBuffer$ and $Filter5x5$ are made for detecting four different orientation of lines in parallel. Each combination perform 2D convolution with it's representative kernel with binary image $i_b(x, y)$ and generated sequence of number from set {-6,-4,-3,-2,2,3,4,6}. Continuous sequence of 3 or 6 represents the presence of straight line. 2 or 4 show the end point of lines.

6.3.5 Object Prediction

Sequence develop by lines detection kernel is processed in this portion of system. If pattern meet the threshold level then there is the chance of the particular line in image. If all four kernel give values up the threshold limit then it is said that there is the chance of men made object in the image. Figure 6.9 shows the implementation of object prediction module using line points counting technique. Module show in Figure 6.9 have four parallel block for counting number of specific integers that are 2,4,3, and 6. If line contain values of point 3 then 2 will be end point of that line similarly for line which contain 6 as a point value then 4 will be end point of line. Now by using Equation 6.9 threshold value can calculated. Last portion of module shown in Figure 6.9 is finding the ration and calculating the threshold value.

$$T_h = \frac{\sum_n L_P}{\sum_m L_{NP}} \tag{6.9}$$

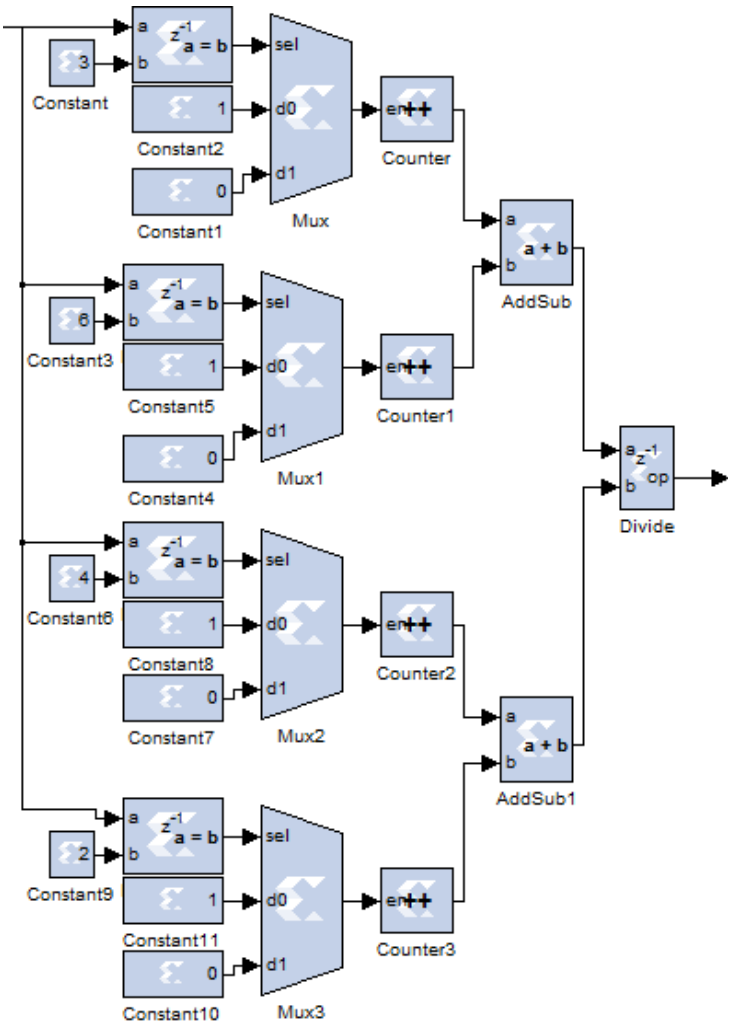


Figure 6.9: 2D convolution using Xilinx system generators Virtex5 line buffer and 5x5 filter kernel

Where T_h is threshold value, L_P is line point integer which is either 3 or 6, n total number of line point integer, L_{NP} is line end point integer which is either 2 or 4, and m is total number of line end points. If $T_h \geq 1$ then there is a chance of close form object in image.

6.4 System Implementation in Xilinx FPGA

System is implemented in Xilinx Vertix 6 FPGA platform. System utilize 1 % of register and LUTs. System also utilize 25 DSP48 blocks detail of device utilization summary is given in Table 6.1. Figure 6.10 show the RTL Schematic of whole system and Figure 6.11 show the RTL schematic of object prediction module. Data is send and receive serially through JTAG which send image data to FPGA and receive processed image and Threshold value T_h from Vertix 6 FPGA.

Slice Logic Utilization	Used	Available	Utilization
Slice Registers	3993	393600	1%
Slice LUTs	2664	196800	1%
Number used as logic	2230	196800	1%
used Memory	160	81440	1%
occupied Slices	974	49200	1%
unused Flip Flop	281	3316	8%
unused LUT	652	3316	19%
Used LUT-FF pairs	2383	3316	71%
slice register sites lost	503	393600	1%
bonded IOBs	57	600	9%
RAMB18E1/FIFO18E1s	45	1408	3%
BUFG/BUFGCTRLs	1	32	3%
DSP48E1s	25	1344	1%
STARTUPs	1	1	100%

Table 6.1: Device Utilization Summary of System implemented in Xilinx Vertix6

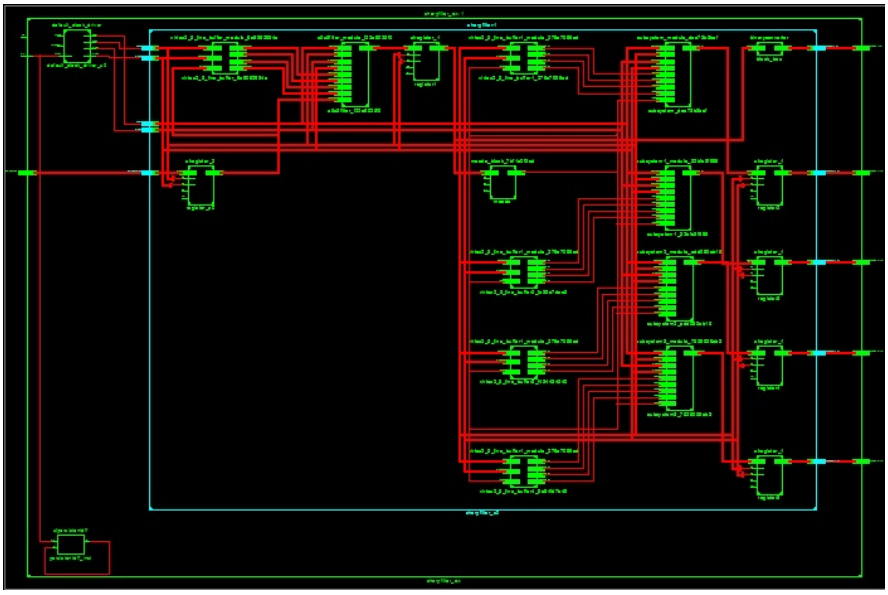


Figure 6.10: RTL Schematic of System implemented in Xilinx Vertix 6

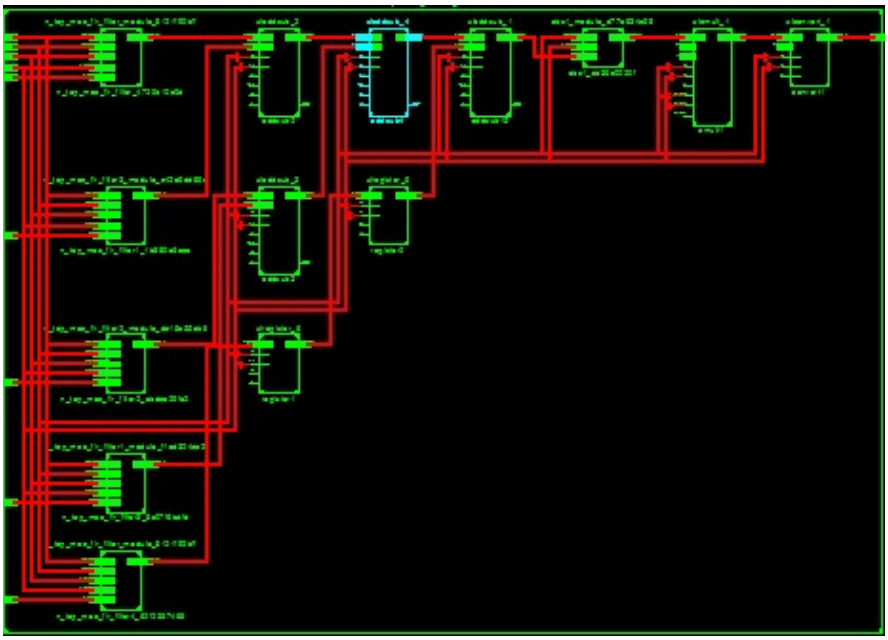


Figure 6.11: RTL Schematic of Object Prediction module of System

6.5 Conclusion

For regular shaped object prediction in underwater image, there are three basic challenge blurriness, false feature and distortion of shape. In this paper contrast-stretching is consider for preprocessing and blur removing, Sobel operator is used for edge detection and four different line detection kernel are used for line detection purpose. Concentration of numbers of lines and its size is bench mark for the presence of regular shaped object. There are many complex and advance object technique for example SIFT [14] are available, but for implement in FPGA very simple but effective techniques are integrated. Integration and implementation of these simple technique is fist step towards the development of advance and intelligent underwater object finder ROVs.

Chapter 7

Conclusion and Future Works

7.0.1 Conclusion

This research suggest the for removing noise form underwater blur images, statistical based simple method is more affective. Histogram Equalization perform better than many complicated recursive de-blurring technique. It is also tested that prediction of object using line detection in underwater object provides impressive information for object finding in underwater environment. Prediction is made using Gaussian Mixture Modal and also with convolution with line detection kernels. For the implementation of algorithm in system, FPGA proves that its performance is better than any other embedded system for image processing and signal processing as it is compare with PSoC in this research.

7.0.2 Future Works

With the implementation of this system in underwater vehicle PAP or ROV, some more techniques are required to implement in FPGA. It includes underwater ve-

hicle control system, underwater camera control system and implementation of statistical prediction algorithm that is GMM or some Fuzzy logic based technique in embedded system to make system more practical.

Bibliography

[1] Hordur Johannsson, Michael Kaess, Brendan Englot, Franz Hover, and John Leonard. Imaging sonar-aided navigation for autonomous underwater harbor surveillance. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4396–4403. IEEE, 2010.

[2] Shahriar Negahdaripour and Pezhman Firoozfam. An rov stereovision system for ship-hull inspection. *Oceanic Engineering, IEEE Journal of*, 31(3):551–564, 2006.

[3] Shigetaka Matsumoto and Yoshihiko Ito. Real-time vision-based tracking of submarine-cables for auv/rov. In *OCEANS'95. MTS/IEEE. Challenges of Our Changing Global Environment. Conference Proceedings.*, volume 3, pages 1997–2002. IEEE, 1995.

[4] M Caccia. Vision-based rov horizontal motion control: Near-seafloor experimental results. *Control Engineering Practice*, 15(6):703–714, 2007.

[5] Soo-Chang Pei and Chien-Cheng Tseng. An efficient design of a variable fractional delay filter using a first-order differentiator. *Signal Processing Letters, IEEE*, 10(10):307–310, 2003.

[6] LB Lucy. An iterative technique for the rectification of observed distributions. *The astronomical journal*, 79:745, 1974.

[7] William Hadley Richardson. Bayesian-based iterative method of image restoration. *JOSA*, 62(1):55–59, 1972.

[8] M. Cannon. Blind deconvolution of spatially invariant image blurs with phase. *Acoust, Speech, Signal Processing*, 24(1):58–63, 1976.

[9] GR Ayers and J Christopher Dainty. Iterative blind deconvolution method and its applications. *Optics letters*, 13(7):547–549, 1988.

[10] BC McCallum. Blind deconvolution by simulated annealing. *Optics Communications*, 75(2):101–105, 1990.

[11] Deepa Kundur and Dimitrios Hatzinakos. A novel blind deconvolution scheme for image restoration using recursive filtering. *Signal Processing, IEEE Transactions on*, 46(2):375–390, 1998.

[12] HD Cheng and XJ Shi. A simple and effective histogram equalization approach to image enhancement. *Digital Signal Processing*, 14(2):158–170, 2004.

[13] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[14] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[15] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.

[16] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

[17] Hastie T. Friedman, J. H. and R. Tibshirani, 1998. Additive Logistic Regression: a Statistical View of Boosting. Technical Report, Dept. of Statistics*, Stanford University.

[18] Tibshirani R. Friedman J. H. Hastie, T., 2001. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics.

[19] Robert E. Freund, Yoav; Schapire, 1995. A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting.

[20] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[21] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–900. IEEE, 2002.

[22] Programmable system-on-chip. http://www.cypress.com/psoc/?source=CY-ENG-HOMEPAGE. Accessed: 2010-01-01.

[23] Edward H. Currie and David Van Ess. PSoC3/5 Reference Book. pages 131 – 216, March 29 - 2010.

[24] Cypress semiconductor corporation. http://www.cypress.com/. Accessed: 2014.

[25] Edward H. Currie and David Van Ess. PSoC3/5 Reference Book. page 67, March 29 - 2010.

[26] Yeong-Taeg Kim. Contrast enhancement using brightness preserving bi-histogram equalization. *Consumer Electronics, IEEE Transactions on*, 43(1):1–8, 1997.

[27] Kurt Rossmann. Point spread-function, line spread-function, and modulation transfer function: Tools for the study of imaging systems 1. *Radiology*, 93(2):257–272, 1969.

[28] RE Hufnagel and NR Stanley. Modulation transfer function associated with image transmission through turbulent media. *JOSA*, 54(1):52–60, 1964.

[29] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.

[30] Paul VC Hough. Method and means for recognizing complex patterns. *US patent*, 3(069):654, 1962.

[31] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[32] Geoffrey J McLachlan and Kaye E Basford. Mixture models. inference and applications to clustering. *Statistics: Textbooks and Monographs, New York: Dekker, 1988*, 1, 1988.

[33] AN77759 Getting Started with PSoC5LP. http://www.cypress.com/?rID=60890. Accessed: 2014-02-13.

[34] Just enough verilog for psoc. http://www.cypress.com/?docID=42936. Accessed: 2014-07-13.

[35] MATLAB. *version 7.12.0 (R2011a)*. The MathWorks Inc., Natick, Massachusetts, 2011.

[36] Rafael C. Gonzalez Richard E. Woods. *Digital Image Processing*. Pearson Prentice Hall, third edition, 2008.

[37] Prof.A.V.Gokhale Neha. P. Raut. Fpga implementation for image processing algorithms using xilinx system generator. *IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)*, (4):26–36, May Jun. 2013.

[38] Andrea Vedaldi and Andrew Zisserman. Sparse kernel approximations for efficient classification and detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2320–2327. IEEE, 2012.