Journal of Engineering Science and Technology
Vol. 12, No. 9 (2017) 2536 - 2550

HARDWARE REALIZATION OF CANNY EDGE DETECTION ALGORITHM FOR UNDERWATER IMAGE SEGMENTATION USING FIELD PROGRAMMABLE GATE ARRAYS

ALEX RAJ S. M.[1,*], CLARIS JOSE[2], SUPRIYA M. H.[1]

[1]Department of Electronics, Cochin University of Science and Technology, Kochi, India
[2]Department of Electronics and Communication Engg., Government Engineering College
Barton Hill, Thiruvananthapuram, India
*Corresponding Author: alex.raj@gecbh.ac.in

Abstract

Underwater images raise new challenges in the field of digital image processing technology in recent years because of its widespread applications. There are many tangled matters to be considered in processing of images collected from water medium due to the adverse effects imposed by the environment itself. Image segmentation is preferred as basal stage of many digital image processing techniques which distinguish multiple segments in an image and reveal the hidden crucial information required for a peculiar application. There are so many general-purpose algorithms and techniques that have been developed for image segmentation. Discontinuity based segmentation are most promising approach for image segmentation, in which Canny Edge detection based segmentation is more preferred for its high level of noise immunity and ability to tackle underwater environment. Since dealing with real time underwater image segmentation algorithm, which is computationally complex enough, an efficient hardware implementation is to be considered. The FPGA based realization of the referred segmentation algorithm is presented in this paper.

Keywords: Underwater, Segmentation, Edge detection, Noise immunity, FPGA.

1. Introduction

Underwater image processing is accepted as one of the noticeable area of research and secure enormous assiduity within few years due to significant applications in defence, medical field, security, mining etc. Underwater images are somewhat complex compared to the visual information collected from other mediums since the physical properties of the water medium which may incorporate degradation

Nomenclatures

F	Image function
G	Resultant intensity gradient
G_x	Intensity gradient along x direction
G_y	Intensity gradient along y direction
H_{ij}	Gaussian kernel expression
I	Image Representation or Illumination function
K	Gaussian kernel size parameter
P	Logical predicative
R	Reflection function

Greek Symbols

θ	Direction of intensity transition, deg.
σ	Standard deviation

Abbreviations

ASP	Application Specific Processor
CPU	Central Processing Unit
FPGA	Field Programmable Gate Array
JTAG	Joint Test Action Group
LCD	Liquid Crystal Display
TFT	Thin Film Transistor
USB	Universal Serial Bus

effects to the data [1]. The major concern when dealing with underwater images is poor visibility caused by the light attenuation and its high susceptibility to noise.

The underwater image segmentation is recognized as a powerful practice to analyse the underwater images and to detect the objects contained in the visual data. Due to the aforementioned complexities, it is tough task to process the underwater images without cost crucial information involved. Most of the image processing techniques focused on extracting hidden crucial information with extreme accuracy level from the images. Hence image segmentation is emerged as the prominent image processing technique, widely used as decisive step in many complicated algorithms according to the applications.

2. Image Segmentation

An image is a visual representation of information which consist of ordered arrangement of pixels in rows and columns in the form of a matrix. The spatially limited location of intersection of a column and a row specifies the corresponding pixel in an image. An image function $f(x, y)$ is a mathematical representation of an image as a function of spatial variables x and y, composed of two spatially varying functions $I(x, y)$ and $r(x, y)$. The product of illumination function $i(x, y)$ and reflectance function $r(x, y)$ give the intensity of the image at a particular pixel location (x, y) [2].

$$f(x, y) = i(x, y).r(x, y) \quad (1)$$

Segmentation is the means to partition an image into purposeful parts to enhance the vital information hidden, which is relevant for the specified application. The goal of image segmentation is to cluster the image pixels into salient image regions which enfold the area of interest as per the prerequisite. This technique extricates the recommended information based on pixel characteristics such as intensity, grey level, texture, colour and other features. The segmentation process of an image stringently follow the conditions [2, 3] mentioned below.

For an image I, having sub regions I_1, I_2, $I_3 \ldots I_n$

1. $I = \bigcup_{i=1}^{n} I_i$
2. $I_i \cap I_j = \phi, if\ i \neq j$
3. $P(I_i) = true, i = 1, 2, ...n$
4. $P(I_i \cup I_j) = false, i \neq j$

where P is logical predicate derived to apply on segments in the image which pointed out the terms to be met for effective segmentation process**.**

3. Segmentation Techniques

The image segmentation prevails as a noticeable research area for many years and search for an optimum method for underwater image processing applications definitely boost the challenges in this field. The absence of a universal segmentation approach, which is applicable to all types of images in different application retain this field as a most demanded research area. The persisting experiments in the field of image processing had led to evolvement of many segmentation approaches marginally differs from others, having variations in performance based on the application and procedure involved [4]. Practical applications of underwater image segmentation algorithms ranges from object recognition, object identification, medical purposes, constructive handling of noisy images, different authentication systems, satellite image processing etc. So, the segmentation approach most appropriate for underwater imagery is relevant in the current scenario.

Threshold or intensity based segmentation is probably the most frequently used technique to segment an image, in which the image pixels are classified into different pixel groups based on the intensity values of the pixel and the prefixed threshold value [5]. This technique mostly preferred to segment a light object from dark background and vice versa. Thresholding can be done either globally or locally according to the requirement. Intensity based segmentation approach is one of the powerful, simple, computationally inexpensive technique which can be easily realized. The main concern regarding this method is that, it is highly sensitive to noise and completely depends on threshold value [6].

Region based segmentation is often preferred algorithm which grant high immunity to noise and easiness. This approach leads to dissolution of the image into arenas similar according to a set of predefined conditions. Region based segmentation mainly incorporate region growing and region split and merge methods. Region growing technique distinguish pixels which belongs to a group based on some peculiar features specific to that group [7]. This method aligns the image pixels into different categories hinged on some predefined constraints. Region splitting and merging method rely on quad tree data for realization. It divides the entire image into four quadrants if some conditions involved is fulfilled and the process extended till the condition fails. Then merge the regions in an attempt to satisfy the conditions of reasonable image segmentation [8].

Clustering in one of the theory based segmentation approach which is an unsupervised learning exercise. Segmentation process starts with a finite set of categories known as clusters having some similarity benchmark to classify image pixels. The escalation of inter class similarity and minimization of intra class similarity are ensured to improve quality of segmentation process [9]. In Agglomerative hierarchical clustering approach, each pixel information is considered as a cluster and finally blend the pertinent clusters into single cluster. The portioning clustering algorithm starts with finite numbers of clusters and gradually identify the pixels fit to that cluster [10]. K-means hard clustering algorithm distinguish the pixels belongs to a particular group by comparing distance measured between the pixel and the centroid of the cluster over a prefixed value [11]. The predefined restricted number of clusters is the main drawback of this algorithm. Optimal segmentation method adopted in situations when there is no defined borderline between different objects in an image is fuzzy clustering. This approach classifies the pixels into different clusters on the basis of similarity index including distance, intensity, connectivity etc. Fuzzy Clustering Mean algorithm is most popular since it can sustain the imperative information than other approaches already reviewed [12].

Image segmentation based on Partial Differential Equation (PDE) is mainly carried out to find significant objects in presence of noise and other ambiguities. This method is accomplished by active contour model or snakes. The computer developed curves having ability to trace the image by the controlled motion within the image under the influence of internal and external forces is called active contours or snakes .The evolvement of an image is considered as a partial differential equation and is solved to achieve required level of segmentation [13]. Another substantial method to handle the segmentation using PDE is level set algorithm in which the curves or surfaces are staged as the zero-level set of a higher dimensional hyper surface. It has disparate advantages like stability and irrelevancy with topology, easiness in solving the problems of corner point producing, curve breaking and combining etc. The main concern about the method is that the curve may eventually pass through object boundaries [14].

Artificial Neural Network based segmentation is also known as energy minimization approach of segmentation. This approach is highly insensitive to noise due to its robustness and high computational speed that can be acquired make this method apt for real time applications. Neural network segmentation starts with identification of features to be inputted then the new images were segmented with trained neural network [15].

Another powerful classification of the image segmentation technique is discontinuity based segmentation approaches. In this category, subdivision of images is carried out on the basis of abrupt changes in the intensity of grey levels of an image contributed by the boundaries of the objects which is to be distinguished. The image pixels exhibit steep intensity variation results in edges of the objects. So, edge detection is considered as the basis for discontinuity based segmentation procedure. Most popular methods to identify true edges present in the image are first derivative, second derivative, Laplacian of Gaussian approach and canny edge detector [16, 17]. Canny edge detector is recognized as a better edge detection approach because of its potential to disclose strong and weak edges in the image which is the major requirement in underwater image processing. Hence canny edge detection based segmentation technique is widely accepted for various applications especially in underwater image processing [18, 19].

4. Canny Edge Detection Based Segmentation

Canny Edge detection has been widely applied techniques in various image processing systems, which extract useful high level information from different vision objects and thereby dramatically diminish the amount of data to be analysed for the expected results [20]. The capability of this algorithm to tackle the embedded noise in the visual information make it good enough for underwater applications. An optimal edge detector should detect all true edge with low error rate and distinguished edge points should be placed accordingly. There are numerous edge detection based segmentation algorithms, slightly differs from each other and apt for different environments. Among the discontinuity detection algorithms introduced so far, canny edge detection based segmentation is most preferable in underwater ambiance because of its high reliability and good detection characteristics [21]. Canny edge detection is a five-stage segmentation approach which is privileged to find both strong and weak boundaries efficiently.

The multistage segmentation algorithm incorporate the following stages [22]

- Smoothening of image using Gaussian filter to eradicate the noise exist.
- Determine intensity gradients to get steep variations in intensity.
- Non-maximum suppression to get rid of undesirable ridges around local maxima.
- Potential edge identification by double threshold process.
- Hysteresis for elimination of weak boundaries but not connected to strong edge detected.

The first step in canny edge detection is image smoothening by using the Gaussian filter which blur the image to remove noise in the image. The noise sometimes had a direct impact on the edge detection process and will adversely affect the detector outputs .So the noise should filtered out as far possible to deal with the presage impact on segmentation process [22, 23]. The deep effects of noise on detection process can be lightened by a smoothening filter and the required level of smoothening can be achieved by convolving the image with Gaussian filter. The Gaussian kernel with size (2k+1) (2k+1) is given by Eq. (2) [20].

$$H_{ij} = \frac{1}{2\pi\sigma^2} * \exp\left(- \frac{(i-k-1)^2 + (j-k-1)^2}{2\sigma^2} \right) \tag{2}$$

The size of the Gaussian kernel varies with the application and the performance of the canny edge detector directly influenced by the size of the kernel. The size of the kernel and the noise sensitivity of the detector are inversely proportional, but the localization error gets raised with increase in size of the Gaussian kernel. The determination of the intensity gradients is the most significant second stage in edge detection procedure. The pixels with large gradient magnitudes are marked as edge pixels which may vary in any direction. Hence the algorithm evaluates intensity transition along vertical, horizontal and diagonal directions using any first derivative operators. The edge gradient and direction of intensity variation are given by Eqs. (3) and (4) [24]

$$G = \sqrt{{G_x}^2 + {G_y}^2} \tag{3}$$

$$\theta = a\tan 2(G_y, G_x) \tag{4}$$

The next step to be followed is non-maximum suppression which is an edge thinning process. This stage allows the algorithm to contemplate on sharp edges present there by accomplish extreme performance efficiency. The edges derived from gradient values determined in the preceding stage may still haze enough to lead to false detection. Thus non-maximum suppression can make sure an exclusive unique response to an edge by suppress all the gradient values to zero leaving the local maximal pixels, which indicates location with the sharpest change of intensity value [24, 25]. Non-maximum suppression can be attained by checking the edge strength of the current pixel with the intensity gradient of the pixel in the positive and negative locations along the gradient direction detected in the previous step and pixels having high gradient magnitudes are preserved as the local maxima and remaining pixels are avoided to reduce computational complexity associated. The effectual performance can be ensured by implementing the algorithm with discrete gradient directions set along with mask of predefined size and this approach suppresses the intensity gradient of the pixel under consideration into zero if its gradient level is not strong enough over the adjacent pixels in the gradient direction under consideration.

The fidelity of the edge detection algorithm can be ensured by the next step of canny edge detection called double threshold. In order to expel the bogus responses from the bothering factors like noise and colour variation features, it is essential to filter out the edge pixel with the weak gradient value and preserve the edge with the high gradient value. The hysteresis threshold stage of canny algorithm which is adapted to the local content of the image, uses multiple thresholds instead of using a single benchmark for the entire image. The two empirically determined threshold values are set to illustrate the substantial edge pixels, one is called high threshold value and the other is called the low threshold value. The pixels with gradient weight less than lower threshold is suppressed altogether and pixel having gradient greater than high threshold is pronounced as strong edges and are conserved with maximal precision. If the edge pixel's gradient value is greater than the low threshold value and less than the high threshold value, they are considered as weak edge pixels.

The desired edge detection based segmentation can be achieved by tracking the edges by hysteresis, in which edges not connected to a definite edge are censored. Normally the weak edge pixel derived from true edges will be connected to the strong edge pixels and the edges developed from latter sources will be separated from strong edges. The edge connections can be established by Binary Large Object-analysis which examine 8-connected neighbourhood to check the existence of a strong edge pixel. As long as there is one strong edge pixel exist, that weak edge point can be identified as one that should be conserved for edge localization [26].

5. Hardware versus Software Implementation

Most of the sensing applications of today require some sort of sophisticated signal processing employing complex algorithms and these are implemented primarily on serial processors. The performance features of conventional CPUs has steadily increased for many decades, but we have now reached the saturation point .The exploitation of high level parallelism is the most promising emerged solution to tackle immensely complicated processing techniques [27]. The required output can be assured by make use of advantages of the parallelism and low power consumption offered by low cost hardware implementation [28].

Real time image processing on a serial processor is a challenging task because of the large data set required to represent the input image and the convoluted algorithms to be followed. Most of the processing techniques in image processing are pixel based and several operations be performed on each pixel in the image resulting in an even large number of operations per second. This may increase processing time and adversely affect the performance efficiency. Thus one perfect alternative is hardware based technologies like Field Programmable Gate Array (FPGA) [29]. The steady and continual advancement in the size and functionality of FPGAs over recent years has made it a promising technology to compensate the drawbacks of serial processors in image processing applications.

FPGAs are reprogrammable silicon chips, having prebuilt logic arrays as the basic building block with provision to configure these chips to implement custom hardware functionality. Radical performance acceleration can be realized for some algorithms exploiting the massive parallelism by an FPGA compared to an inherently sequential processor methodology. The performance enhancement ensured by hardware based FPGA implementation is due to the architectural flexibility of an FPGA which is able to exploit spatial and temporal parallelism efficiently [30, 31]. FPGAs outpace the computing power of digital signal processors (DSPs) and carried out more per clock cycle by hardware parallelism. The hardware level management of input and outputs contribute brisk response and provide close match to the requirements to be fulfilled.

6. Experimental Procedures

The intricateness associated with image processing procedures can be addressed productively using the reprogrammable logic array based FPGA which increases manufacturing efficiency extensively and cutback size and cost demands. The discussed segmentation algorithm had implemented using NanoBoard3000 development system along with intuitive software Altium Designer 10 introducing reprogrammable hardware platform that incorporate the benefits of a high-capacity, low-cost programmable device to realize the interactive implementation and debugging digital signal processing applications especially digital image processing by exploiting parallelism. Altium NanoBoard3000 shown in Fig. 1 have the key features including dedicated system JTAG programming port, and Programmable clock 6 to 200 MHz to make use in FPGA design with Fixed 20MHz reference clock, USB 2.0 High-Speed interface, RS-232 Serial Port , RS-485 Serial Port and 240 x 320 TFT LCD with touch screen [32].

Figure 1 shows the block diagram of Canny edge detection algorithm as explained in section 4. The output of any edge detection algorithm is the edge map of the input image. To get the edge map colour information is not required. Also, to reduce the computation density colour image is converted into greyscale image which will not affect the edge detection process significantly. After obtaining the greyscale image a smoothing operation with a Gaussian operator is performed to reduce the noise in the image. Then magnitude and direction of the edge pixels are computed. Non-maximal suppression step gives the edge map of the image with lot of spurious responses. In order to remove the false edges hysteresis thresholding is done which gives the final output image.

Figure 2 shows the architecture for hardware implementation of Canny edge detection algorithm on Xilinx Spartan3AN FPGA (XC3S1400AN). The processor has three main blocks; Decoder Module and Canny Edge Detection module. The interface to the host PC and to the display is handled by the processor. Original images from the host PC are stored in the SRAM external to the FPGA. From there it is accessed by the JPEG decoder which decodes the image. The decoded image or the raster data of the image is then accessed by the Canny Edge Detection module. The Canny Edge Detection module is implemented according to the block diagram shown in Fig. 1. The canny edge detection module executes the algorithm in a parallel or pipelined routine. The intermediate results are stored in the block RAM in the FPGA. The image smoothening is done using a 3x3 Gaussian mask after which the result is stored in the block RAM. In the next stage Sobel operator is used to calculate the horizontal and vertical gradients. The operation is performed in a pipelined manner. As soon as the horizontal and vertical gradients are available the corresponding gradient magnitude is calculated. The results of which are again stored in the block RAM. The non-maximal suppression stage is implemented as a simple comparator. To remove the false edges after this stage, hysteresis thresholding is done where there are two thresholds. This stage is also implemented in a pipelined manner using comparators and the final output obtained.

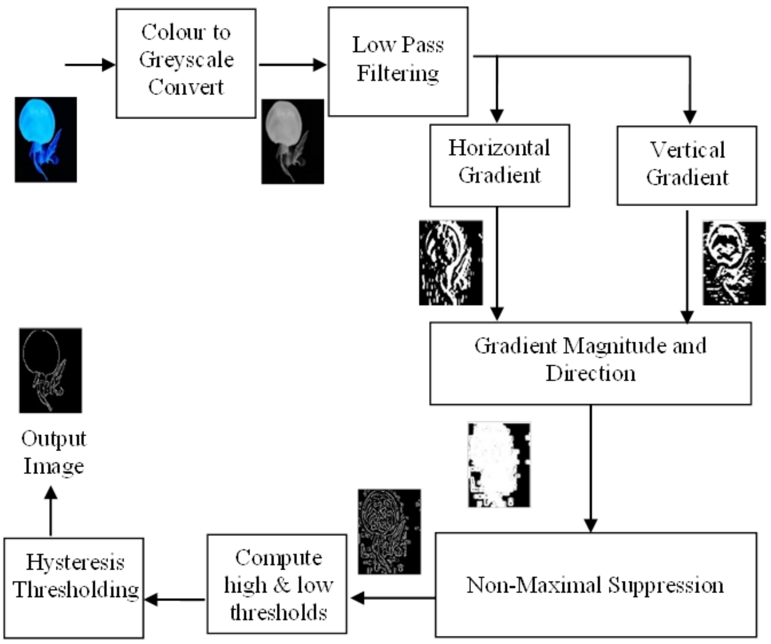

Fig. 1. Block diagram of the Canny edge detection algorithm.

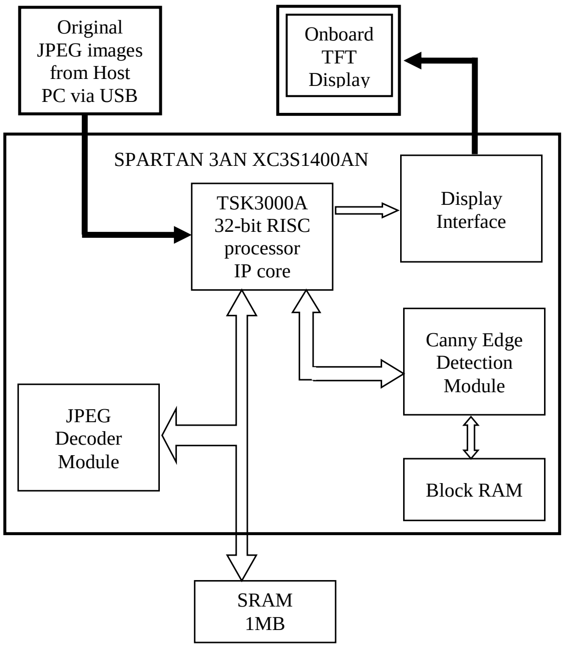

Fig. 2. Architecture for the Canny edge detection algorithm.

7. Results and Discussions

Underwater image segmentation using canny edge detection algorithm were successfully implemented on FPGA using Altium nanoboard NB3000. The segmented image seen on the nanoboard had revealed the competence of Canny edge detection algorithm to tackle tangled underwater images and made known the hidden significant information required for different applications. Various input underwater images used for the implementation are shown in Figs. 3(a), 4(a), and 5(a). The underwater images were obtained the segmented output data had displayed on available 240×320 TFT LCD in development system as shown in Fig. 2. Figures 4 and 6 are the respective output for the input images shown in Figs. 3 and 5. The synthesize report describing hardware features and other characteristics of the field programmable array used for the purpose of canny edge detection segmentation algorithms had been outlined in Appendix A.

Source: The telegraph earth picture galleries, image by joshua lambus/ solent [33].

Fig. 3(a). Input image for canny edge detection.

Fig. 3(b). Segmented image on tft display of nanoboard.

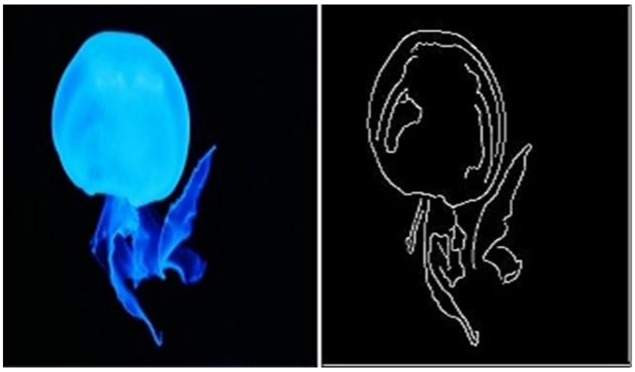

Image Source: Publicdomain Pictures.net [34]

Fig. 4(a). Input image. **Fig. 4(b). Output image.**

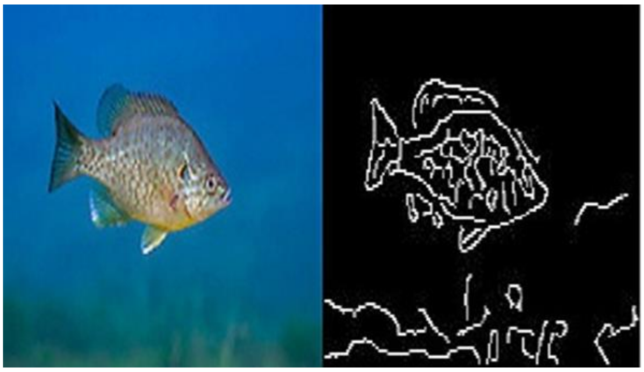

Image Source: Engbreston underwater photography image, by Issac Szabo [35]

Fig. 5(a). Input image **Fig. 5(b). Output image**

Edge detection performance analysis

The performance of the edge detection is analysed by comparing the perceptual significance of the resulting edge map with the one produced by the software based Canny edge detector. For that the time of execution taken by both the software based Canny edge detector and the hardware based Canny edge detector implemented in FPGA. Table1 shows the result for an image of size 320*240 pixels. The software based Canny edge detection takes around 3.4 ms while the hardware based Canny edge detector takes only 0.78 ms. To make the quantitative analysis the entropy of the resultant image along with PSNR, its correlation and distortion is calculated [35]. Hardware comparison was not performed so far.

Table 1. Runtime Comparison on various platforms.

Image	Size	Software@2.4GHz Greyscale image	FPGA @ 100MHz
Fig. 1	320×240	3.43 ms	0.78 ms
Fig. 2	320×240	3.42 ms	0.78 ms
Fig. 3	320×240	3.44 ms	0.78 ms

Table 2. Conformance evaluation.

Evaluation indicators	Canny software			Canny hardware		
	Fig. 1	Fig. 2	Fig. 3	Fig. 1	Fig. 2	Fig. 3
Entropy	0.6929	0.3222	0.2958	0.283	0.177	0.217
PSNR	6.7970	10.4425	8.7766	15.935	13.535	10.511
Correlation	0.0491	0.1143	0.0328	0.730	0.412	0.314
Distortion degree	9.3434	10.2797	22.2826	3.350	8.503	18.955

Table 3. Conformance evaluation for standard image.

Evaluation Indicators	Canny Standard		
	Fig. 1	Fig. 2	Fig. 3
Entropy	0.2869	0.2821	0.4512
PSNR	6.415	8.253	8.459
Correlation	1	1	1
Distortion degree	0	0	0

Performance evaluation of the edge detector is given in Table 2. A standard image to form the basis for evaluation is obtained using the inbuilt canny edge detection operation in MATLAB. From Table 1, Table 2 and Table 3, the statistical results of the edge detector implemented in FPGA are better than the traditional Canny edge detector implemented in software. The hardware based Canny edge detector is better, compared with the standard image in the entropy, the PSNR and the distortion degree. Consequently, the hardware based implementation of Canny edge detector is very effective.

8. Conclusion

Image segmentation is remaining as a demanded research area because of dearth of a universal approach applicable to handle all types of visual information. In this paper, we have deliberated the potentiality of segmentation methods to deal with the complex underwater images to expose crucial information without exhibit considerable loss. The segmentation techniques can help to analyse underwater images and divulge pivotal information efficiently. Even though there does not exist any universality, most popular segmentation technique apt for underwater image processing is Canny edge detection based segmentation which maintain significant information of the image as such with least distortion. The canny edge detection algorithm provides following advantages over other approaches.

- Probability analysis for finding error rate and efficiency.
- Localization and response with extreme accuracy.
- Enhancing signal to noise ratio.
- Superior detection capability under noise conditions.

The high-performance enhancement obtained by the FPGA implementation of the referred segmentation algorithm implies the complex computations is likely to rely on the ability to efficiently exploit the advantages of parallelism by hardware based realizations in coming era of image processing technology. FPGA platform can incorporate following benefits to the system.

- Flexibility in architecture.
- Parallelism based implementation to improve speed factors.
- Reprogrammable and compact system.
- Cost effective system with ease of design.

Most of the discussed segmentation approaches only tries to allot the pixels to a particular class. Sometimes pixels in the image may belongs to two or more clusters resulting straddling pixels over the real boundary of the objects. Under such scenario probabilistic approaches of segmentation like Gaussian Mixture Model can be considered as a better solution which is highly accurate but complex in nature. The FPGA implementation of the Gaussian Mixture Model based segmentation is selected as the future work to achieve better performance efficiency in this area.

References

1. Schettini, R.; and Corchs, S. (2010). Underwater image processing: state of the art of restoration and image enhancement methods. *EURASIP Journal on Advances in Signal Processing*, 746052.
2. Gonzalez, R.C.; Woods, R.E.; and Eddins, S.L. (2004). *Digital image processing using MATLAB* (2nd ed.). India: Pearson Education..
3. Khan, A.M.; and Ravi, S. (2013). Image segmentation methods: a comparative study. *International Journal of Soft Computing and Engineering*, 3(4), 84-92.
4. Nagabhushana, S. (2005). *Computer vision and image processing* (1st ed.). India: New Age International.
5. Kang, W.X.; Yang, Q.-Q.; and Liang, R.-P. (2009). The comparative research on image segmentation algorithms. *Education Technology and Computer* Science. Hubei, China, 703-707.
6. Medina-Carnicer, R.; Carmona-Poyato, A.; Munoz-Salinas, R.; and Madrid-Cuevas, F.J. (2010). Determining hysteresis thresholds for edge detection by combining the advantages and disadvantages of thresholding methods. *IEEE Transaction on Image Processing*, 19(1), 165-173.
7. Kaganami, H.G.; and Beiji, Z. (2009). Region-based segmentation versus edge detection. *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. Kyoto, Japan.
8. Prasantha, H.S.; Shashidhara, H.L.; Murthy, K.N.B.; and Madhavi, L.G. (2010). Medical image segmentation. *International Journal of Computational Science and Engineering*, 2, 1209-1218.
9. Chang, Y.L.; and Li, X. (1994). Adaptive image region-growing. *IEEE Transactions on Image Processing*, 3, 868-872.
10. Duraiswamy, K.; and Mayil, V.V. (2008). Similarity matrix based session clustering by sequence alignment using dynamic programming. *Computer and Information Science*, 1(3), 66.

11. Jain, A.K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31, 651-666.
12. Choong, M.Y.; Kow, W.Y.; Chin, Y.K.; Angeline, L.; and Teo, K.T.K. (2012). Image segmentation via normalised cuts and clustering algorithm. *IEEE International Conference on Control System, Computing and Engineering*. Pemang, Malaysia.
13. Li, C.; Huang, R.; Ding, Z.; Gatenby, J.C.; Metaxas, D.N.; and Gore, J.C. (2011). A level set method for image segmentation in the presence of intensity inhomogeneities with application to MRI. *IEEE Transactions on Image Processing*. 20(7), 2007-2016.
14. Chan, T.F.; and Vese, L. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, 10(2), 266-277.
15. Pal, N.R.; and Pal, S.K. (1993). A review on image segmentation techniques. *Pattern Recognition*, 26(9), 1277-1294.
16. Ravi, S.; and Khan, A.M. (2012). Operators used in edge detection computation: a case study. *International Journal of Applied Engineering Research*, 7, 2022-2027.
17. Senthilkumaran, N.; and Rajesh, R. (2009). Edge detection techniques for image segmentation-a survey of soft computing approaches. *International Journal of Recent Trends in Engineering and Technology*, 1(2), 250-254.
18. Muthukrishnan, R.; and Radha, M. (2011). Edge detection techniques for image segmentation. *International Journal of Computer Science & Information Technology*, 3(6), 259-267.
19. Thakare, P. (2011). A study of image segmentation and edge detection techniques. *International Journal of Computer Science and Engineering*, 3(2), 899-904.
20. Canny edge detector. Wikipedia, The free encyclopedia, (2004). Retrieved from https://en.wikipedia.org/wiki/Canny_edge_detector.
21. Foresti, G.L. (2001). Visual inspection of sea bottom structures by an autonomous underwater vehicle. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(5), 691-705.
22. Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6), 679-698.
23. He, X.; Jia, W.; and Wu, Q. (2008). An approach of canny edge detection with virtual hexagonal image structure. *Internaltional Conference on Advanced Concepts for Intelligent Vision Systems*. France.
24. Wang, B.; and Fan, S. (2009). An improved Canny edge detection algorithm. *2009 Second International Workshop on Computer Science and Engineering*.
25. Maini, R.; and Aggarwal, H. (2009). Study and comparison of various image edge detection techniques. *Internatinal Journal of Image Processing*, 3(1), 1-11.
26. Padmavathi, G.; Muthukumar, M.; and Thakur, S.K. (2010). Implementation and comparison of different segmentation algorithms used for underwater images based on nonlinear objective assessments. *3rd International*

Conference on Advanced Computer Theory and Engineering(ICACTE). Chengdu, China.

27. Shrikanth, G.; and Subramanian, K. (2008). Implementation of fpga-based object tracking algorithm. Postgraduate Thesis. SVCE, India.
28. Xilinx (2014). What is a FPGA? Xilinx Getting Started, (2014). Retrieved from http://www.xilinx.com/fpga/.
29. Thomas, D.B.; Howes, L.; and Luk, W. (2009). A comparison of CPUs, GPUs, FPGAs, and massively parallel processor arrays for random number generation. *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays*. Californina, USA.
30. Guo, Z.; Najjar, W.; Vahid, F.; and Vissers, K. (2004). A quantitative analysis of the speedup factors of FPGAs over processors. *Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays*. New York, USA.
31. Kuon, I.; and Rose, J. (2007). Measuring the gap between FPGAs and ASICs. IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(2), 203-215.
32. Altium, NanoBoard 3000 Series Altium Documentation, (2015). Retrieved from http://techdocs.altium.com/display/HWARE/NanoBoard+3000+Series.
33. Lambus, J. (2011). Luminous creatures: Glowing deep sea creatures photographed by JoshuaLambus. Retrieved from http://www.telegraph.co.uk/news/earth/earthpicturegalleries/8445782/Luminous-creatures-Glowing-deep -sea-creatures-photographed-by-Joshua-Lambus.html?image=5.
34. Kratochvil, P. (2011) and BlueJellyfish, (2011). Retrieved from http://www.publicdomainpictures.net/view-image.php?image=85
35. Szabo, I. (2013). Engbreston Underwater Photography. Retrieved from http://fish.photoshelter.com/image/I0000CBjM8JT3xEc.
36. Deng, C.X.; Wang, G.B.; and Yang, X.R. (2013). Image edge detection algorithm based on improved canny operator. *International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*. Tianjin, China.

Appendix A

Resource Utilization and Synthesize Report

A.1. Area Estimates

Area Estimate (2 Input Gate Count)	: 9507
Area Estimate (LUT Count)	: 5597

A.2. Xilinx Mapping Report File for Design

Target Device	: XC3S1400AN
Target Package	: FGG676
Target Speed	: -4
Mapper Version	: SPARTAN3A

A.3. Design Summary

Table A1. Design Characteristics.

Logic Utilization	Number of Slice Flip Flops	30%
	Number of 4 input LUTs	58%
Logic Distribution	Number of occupied Slices	83%
	Number of Slices containing only related logic	100%
	Number of Slices containing unrelated logic	0%
Total Number of 4 input LUTs	Total	64%
	Number used as logic	13,009
	Number used as a route-through	1,221
	Number used as 16x1 RAMs	14
	Number used for Dual Port RAMs	256
Slice Logic Distribution	Number of bonded IOBs	36%
	Number of BUFGMUXs	4%
	Number of MULT18X18SIOs	40%
	Number of RAMB16BWEs	37%
Timing Constraints	Average Fan out of Non-Clock Nets	3.12
	Total REAL time to MAP completion	2 mins 32 secs
	Total CPU time to MAP completion	18 secs
Peak Memory Usage		463 MB