Parallel algorithms for Edge detection in an Image

C.Mala[1]
Department of Computer Science and Engineering
National Institute of Technology
Tiruchirappalli , India – 620 015
mala@nitt.edu

M.Sridevi[2]
Department of Computer Science and Engineering
National Institute of Technology
Tiruchirappalli , India – 620 015
msridevi@nitt.edu

[1]Coresponding Author

Abstract— **Edge detection is an important process in image segmentation, object recognition, template matching, etc. It computes gradients in both horizontal and vertical directions of the image at each pixel position to find the image boundaries. The conventional edge detectors take significant time to detect the edges in the image. To reduce the computational time, this paper proposes parallel algorithms for edge detection with Sobel, Prewitt and Robert first order derivatives using a Shared Memory - Single Instruction Multiple Data (SM - SIMD) parallel architecture. From the experimental results, it is inferred that the proposed parallel algorithms for edge detection are faster than the conventional methods.**

Keywords - ***Segmentation; Edge detector; Derivative; Parallelism; Convolution.***

I. Introduction

The segmentation can be performed based on existence of either similarity or dissimilarity in the image. It forms different clusters based on the properties such as intensity, color, texture, etc. The result of segmentation process is used in the applications such as biomedical systems for detecting tumors, transportation systems to locate objects such as road, forest, land, etc in the satellite images, object and pattern recognition, biometric for identifying the persons through Iris ,finger print, image retrieval, robotic system [16], etc.

Two approaches are widely used for detecting the boundaries of the image namely, Edge detection and region based approach [17]. The basic idea of these approaches is to partition the foreground region form the background of the image using homogeneous or heterogeneous criteria.

The edge detection is simple and most efficient technique used in image segmentation problems. The edges are formed, when there is an abrupt change in the intensity values. They use derivative operators such as first or second order for solving the problem. The results of segmentation using edge detection mainly depend on factors such as type of mask, size of the mask and threshold value. The classic traditional edge detection algorithms such as Sobel edge detector, Perwitt Edge detector, Robert first order derivative, Laplacian of Gaussian edge detector, Canny's Edge Detector are applied to color and binary image [10]. The result of such application yields reasonable edge detection. These algorithms are in turn modified and improved to increase their performance and computational complexity [4, 5, 11].

In case of region based approaches the segmentation process is carried out by techniques such as K-means, Fuzzy C-Means, Neural Network, Fuzzy logic and Fuzzy Neuro Logic [11]. Most of the medical imaging uses region based approaches to identify specific patterns in the image.

In any real time application the key requirement is to process large images and its processing time should be minimized as much as possible in order to show results simultaneously. To effectively achieve such requirements, the traditional sequential processing of edge detection does not adhere.

In order to improve the computational efficiency of image segmentation process, this paper proposes a parallel algorithm that focuses on different levels of parallelism on the conventional edge detection methods.

II. Related Works

For detecting the edges in images with white Gaussian noise, a method is proposed by improving the Sobel Edge detection method [5]. This method combines Sobel edge detection operator and soft-threshold wavelet de-noising for edge detection. This method outstandingly filters maximum Gaussian white noise in the high-frequency signal and effectively removes the noise and detects edges.

Traditional Prewitt Edge Detection algorithm is very sensitive to noise. To make the algorithm robust, an improved Prewitt algorithm is proposed in [4] with eight different masks to perform the operation in eight directions. The improvement carried out in the proposed algorithm is as follows: 1.Mean value of the gradient magnitude is used as final gradient magnitude, 2. Automatic threshold is used and 3. Eight neighborhood templates remove the isolated single pixel noise. More delicate, smooth and continuous edges are formed and also it reduces random noise significantly.

The various classical and non classical approaches available for preprocessing and segmentation of color image are given in [3]. The multi-resolution segmentation techniques based on graph theoretic approach, segmentation as a linear problem instead of the eigenvector approach, Mean Shift (MS) analysis algorithm, MS Segmentation, MS Clustering, Markov Random Field (MRF) models in segmentation, approximate to the MRF model-based image segmentation technique and mixture models are discussed in

978-1-4799-4224-4/14 $31.00 © 2014 IEEE
DOI 10.1109/NBiS.2014.15

[3]. All the above techniques work on priori knowledge, hence non classical approaches [11, 14, 21] such as Neural Network, Fuzzy, Genetic Algorithm (GA) and Wavelet are used for image segmentation process. The performances of these methods depend on various factors on data distributions, operating parameters and the operating environment.

An edge detector based on color map is proposed in [6] using new morphological operation. The proposed method computes color gradient whereas the classical morphological operates on single channel. The method is computationally efficient and insensitive to noise. The method based on morphological edge detector and background differencing [7] is proposed and it is used for real time traffic analysis application.

The methods mentioned in the literature [13, 19, 20] are computationally expensive and slow. To overcome this drawback, an attempt is made to make the conventional edge detectors to operate in a parallel environment [1, 2, 8, 12, 15]. In any multi-core CPU and on many-core Graphics Processors Units (GPU), parallelization can be employed using various strategies. Parallelization is an efficient solution for applications that process enormous amount of data which requires huge computational time. In [1], two strategies are proposed namely, loop-level parallelism and domain decomposition, for Canny Edge Detection. The impact of data size on scalability is evaluated using different sizes (512x512 and 2048x2048) of images. In a multi-core CPU, the speedup of domain decomposition and loop-level parallelism are compared and results show that domain decomposition strategy offers a considerable lower execution time for large image sizes due to increased data locality. For small images (512x512), no considerable difference in the execution time could be found between the two strategies. Similarly, many-core graphics processors exhibit a lower execution time for the combined strategy due to the fact that the GPU architecture is more suitable for fine-grained data parallelism.

In [2], parallelism is achieved for Sobel edge detector using FPGA. The input image is converted into a binary image and they are stored in a text format. The approximations of the gradient are calculated with the magnitude in Verilog HDL (Hardware Description Language) synthesis and then it is simulated and checked with respect to the design summary and timing analysis. The pixel values of center, north, south, east, west, south east, south west, north east and northwest pixel for entire binary image both in row and column wise are identified. This made scanning of gradients in horizontal and vertical directions. The complexity of the design and the processing time is reduced in FPGA Virtex 4, device XC4VLX200 and package FF1513.

The remainder of the paper is organized as follows: Section III explains the proposed Parallel algorithms for edge detection. The experimental results and their performance analysis are discussed in Section IV. Finally, Section V concludes the paper.

III. PROPOSED WORK

The traditional edge detector is computationally expensive and time consuming due to large number of multiplication and addition operations that are to be performed for finding out the gradients of the image. Hence, they are unsuited for real time applications. To overcome this, a parallel approach is proposed in this paper using EREW - SM SIMD (Exclusive Read Exclusive Write - Shared Memory Single Instruction Multiple Data) parallel architecture [18]. The mask, image and their sub windows are kept in the shared memory. The algorithm requires k * k processors for its execution in parallel. Each processor (P_i) reads the intensity value (I_i) of a sub window simultaneously from the SM. Hence the read operation is exclusive. The convolution operations are performed in the local memory of each processor. Once the processors (P_i) complete the operation, the result will be sent back to the SM simultaneously. This uses exclusive write operation. The operation performed in N processors are same but uses different data sets ie., different subwindows. Hence, EREW SM SIMD model is well suited for the proposed parallel algorithm.

The proposed Parallel Edge Detector (PED) algorithm is given in Algorithm 1. The proposed method uses two levels of parallelism for performing the convolution and thresholding operations. If the input image is a color image, then it is converted to gray as the proposed algorithm handles only gray images. Let size of the image and mask be (m x n) and (k x k) respectively. The image is divided into (m – k +1) * (n – k +1) sub windows or overlapping blocks. The convolution operation [17] is performed between the entire sub window of the image and the mask as shown in Fig.1 which gives impulse response at the center pixel position as given eq. (1).

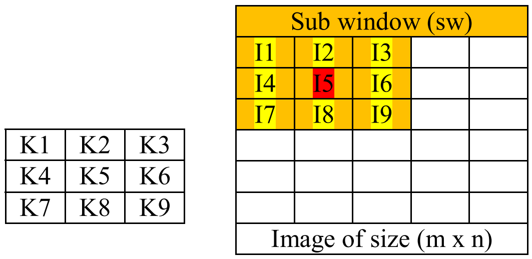
$$cp = \sum_{i=1}^{k*k} I_i * K_i \tag{1}$$

where cp is the impulse response at center pixel; K_i is the mask value in i^{th} location; I_i is Intensity value of subwindow at i^{th} location.

The gradient values (g_x and g_y) of the image are computed by performing convolution operation in both directions in a parallel manner by using N number of processors. The image gradient magnitude is computed using eq.(2) .

$$grad(x,y) = (g_x^2 + g_y^2)^{1/2} \tag{2}$$

where g_x is the convoluted value in x direction and g_y is the convoluted value in y direction.



(a) *(b)*

Fig.1. Representation of Convolution process
a) Mask (3 x 3); b) Image (m x n) and sub window [sw] (3 x 3)

Bilevel thresholding is applied [17] to the computed gradient magnitude which detects edges in the image based on selected threshold value. The result of segmentation mainly depends on the threshold value. Hence, different thresholds (T_i) are used for grouping the image pixels and they yield the resultant image $g_i(x,y)$ by using eq. (3) .

$$g_i(x,y) = \begin{cases} 0, & grad(x,y) < T_i \\ 1, & grad(x,y) \geq T_i \end{cases} \tag{3}$$

where i takes bilevel threshold values from 0 to 255. Among g_i (x,y) results, best one is considered for further processing and can be obtained by execution of the eq. (3) .

The thresholding operation can be done in a parallel manner with different sets of threshold values, further reducing the computational time.

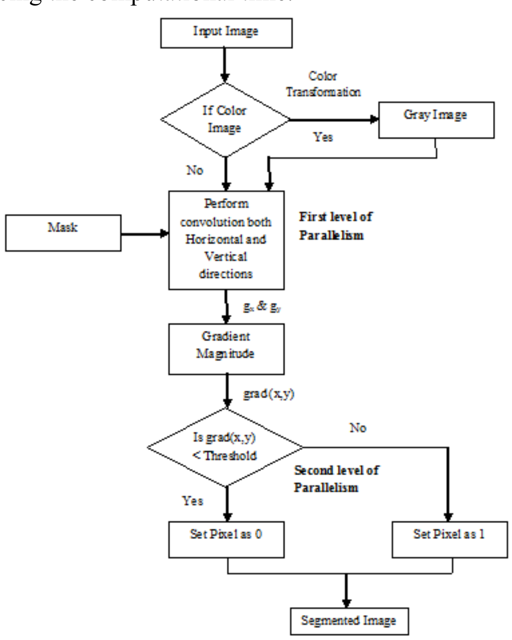


Fig. 2. Flow diagram of the proposed PED algorithm

Algorithm 1: Proposed Parallel algorithm for Edge Detection (PED)

1. Convert the input color image f(x,y) with size (m x n) into gray scale.
2. Perform convolution operation between the image and the selected mask, both in X and Y directions in a parallel manner.
3. Compute gradient magnitude of the image.
4. Apply different bilevel threshold values to the gradient magnitude and get appropriate regions. This process is done in a parallel manner.
5. Display the best segmented result $g_i(x,y)$.

The PED algorithm uses two levels of parallelism. The first level of parallelism is used for convolution operation and the second level is for thresholding process as shown in Fig. 2. The following Sections III A and III B explain the implementation of parallel algorithms.

A. First Level of Parallelism

The data parallelism is applied to step 2 of the Algorithm 1 using EREW SM SIMD parallel architecture model with N processors. The steps involved in parallel convolution operation are given in Algorithm 2.

Algorithm 2: Parallel Algorithm for Convolution Operation

for j = 1 to nb
  for i = 1 to N do in parallel
    1. Multiply the intensity value (I_i) of the subwindow with the value (K_i) in the corresponding mask position as shown in Fig. 1 in the local memory of the processor (P_i).
    2. Add the multiplicative results as given eq.(4) in the Shared Memory (SM)

$$\sum_{i=1}^{k*k} I_i * K_i \tag{4}$$

  end for
end for

The steps 1 and 2 of the algorithm 2 are performed with nb where nb is the number of subwindows as given in eq. (5).

$$nb = (m - k + 1) * (n - k + 1) \tag{5}$$

An uniprocessor takes more time to perform these operations for the entire image. Hence, data parallelism is used to reduce the computational time. Let N number of processors in EREW – SM SIMD be equal to size of the mask (N = k * k). The response at center pixel position is computed for each sub windows. To get better segmentation results, the convolution operation is to be done on both horizontal and vertical directions. Different masks are used for two directions. The conventional first order derivative edge detectors [17] such as Robert, Prewitt and Sobel for defining the mask are explained in the Sections III A a) to III A d).

a) Robert Mask

Robert mask of size 2 x 2 is shown in Fig.3 (a) and (b) for X and Y directions respectively.

-1	0
0	1

(a)

0	-1
1	0

(b)

Fig.3. Robert mask (a) X direction; (b) Y direction

b) Prewitt Mask

The 3 x 3 Prewitt mask is shown in Fig.4 (a) and (b) for X and Y directions respectively.

-1	-1	-1
0	0	0
1	1	1

(a)

-1	0	1
-1	0	1
-1	0	1

(b)

Fig. 4. Prewitt Mask (a) X direction; (b) Y direction

c) Sobel Mask

Sobel mask of size 3 x 3 is shown in Fig.5 (a) and (b) for X and Y directions respectively.

-1	-2	-1
0	0	0
1	2	1

(a)

-1	0	1
-2	0	2
-1	0	1

(b)

Fig.5. Sobel Mask a) X direction; b) Y direction

All the masks can be extended to any size. The result of segmentation depends on two factors i) Proper selection of mask and ii) Size of the mask

The convolution operation provides two results namely, g_x for X direction and g_y for Y direction, based on the corresponding mask values. The magnitude of the gradients [grad(x, y)] are computed using eq.(2).

B. *Second Level of Parallelism*

The edges are determined from the gradient magnitude using a threshold value. The magnitude of the gradient is compared with the selected threshold value (T) to get a binary image (0 and 1) of size (m x n) by using eq. (3). The pixel values (grad(x, y)) greater than T are represented as 1 and others by 0.

The edges should be continuous and thin. The selection of threshold value plays a vital role in image segmentation. The result of segmentation will be influenced by the successive level of processing such as recognition, classification, matching, retrieval, etc. Hence, different thresholds are selected for detecting proper edges. Let the set of threshold values T be $\{t_1, t_2, \ldots t_i\}$, where i varies from 1 to number of threshold used.

Algorithm 3 : Parallel Algorithm for Threshold Operation

for i = 1 to N do in parallel

1. Assign T_i to P_i
2. Each processor (P_i) read grad(x,y) from Shared Memory (SM) to its Local Memory (LM)
3. Compare T_i with gradient magnitude grad(x,y).

$$g_i(x,y) = \begin{cases} 0, & grad(x,y) < T_i \\ 1, & grad(x,y) \geq T_i \end{cases}.$$

end for

Each processor gives the result g_i (x,y). Among them, the best one is considered such that the edges of the image will be continuous, thin, good precision. The edges formed by applying the above three masks are thick. To reduce the edge thickness, thinning process has to be applied for the detected edges.

C. *Algorithm Analysis*

The time complexity for the edge detection when done sequentially and parallely is tabulated in Table 1.

TABLE 1. TIMING ANALYSIS OF EDGE DETECTION IN SEQUENTIAL AND PARALLEL ENVIRONMENT

Operation	Single subwindow [size of subwindow is equal to the size of the mask (k x k)]		Entire image of size (m x n) with nb subwindows	
	Unipr-ocessor (N =1)	EREW SM SIMD ($N = k^2$)	Uniproce-ssor (N =1)	EREW SM SIMD ($N = k^2$)
1 (Multipli-cation)	$O(k^2)$	O(1)	$O(nb * k^2)$	O(nb)
2 (Addition)	$O(k^2)$	$O(\log_2 k)$	$O(nb * k^2)$	$O(nb * \log_2 k)$
3 (Thresho-lding)	$O(k^2)$	O(1)	$O(nb * k^2)$	O(nb)

where N is Number of processors.

IV. Results And Performance Evaluation

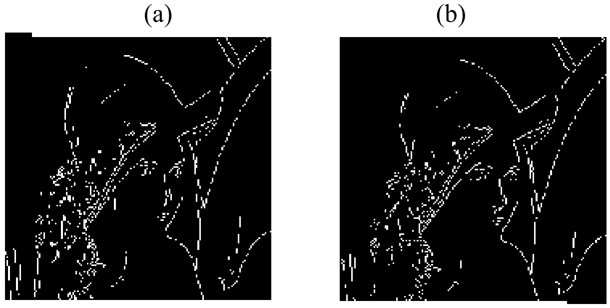
The Proposed Parallel Edge Detector Algorithm (PPEDA) is implemented using EREW SM SIMD architecture and Matlab 12 on Intel core i5 processor with benchmark the images in Berkeley segmentation image. The proposed algorithm is tested for the following scenarios.

Scenario 1: Convolution operation done in parallel
Scenario 2: Thresholding operation done in parallel
Scenario 3: Both Convolution and Thresholding operations done in parallel

A. Experimental Results

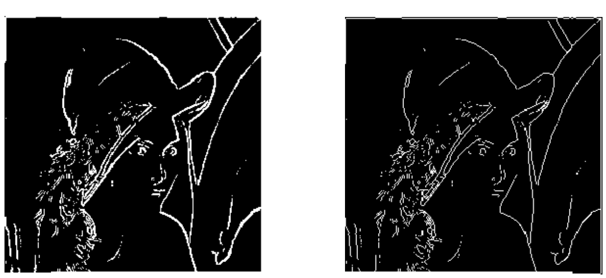
The proposed algorithm is tested for the input image as shown in Fig. 6 a). The different edge detectors (Robert, Prewitt, Sobel) are used for finding out the edges in the image and the segmented images are shown in the figures from Fig. 6 b) to d) respectively with a single bilevel threshold (T =50). The segmented results are obtained (based on Scenario 1) by applying first level parallelism to the respective masks ie., convolution operation is done in parallel manner whereas in cases of thresholding operation, single threshold value say T = 50 is used as a constant for all operators.

(a) (b)



(c) (d)

Fig. 6. Result of segmentation with different mask
a) Input image b) Robert c) Prewitt and d) Sobel

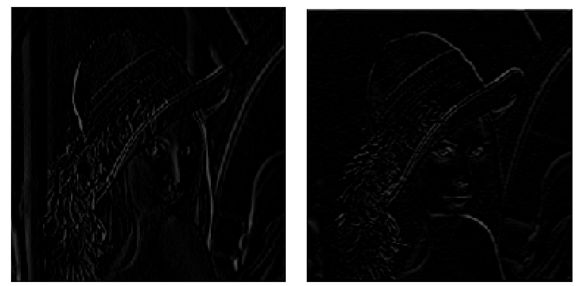


(a) (b)

Fig. 7. Output of Sobel mask (size 3 x 3)
a) With threshold value 50; b) Thinned image

Thick edges are formed as a result of edge detection. To reduce thickness of the edge, thinning process [17] has to be applied to the segmented image and the resultant images are shown in Fig. 7 for Sobel mask.

The convolution operation is performed in both horizontal and vertical direction to compute gradient magnitude. Fig. 8 - 10 show the intermediate result of scenario one.



(a) (b)

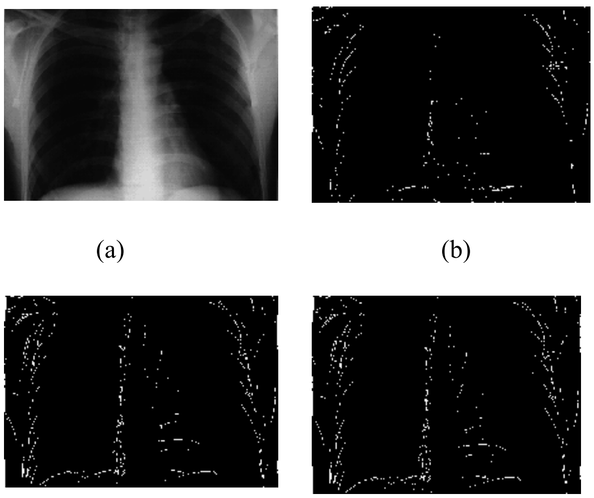
Fig .8. Result of convolution operation in
a) X direction b) Y direction

Fig. 9. Result of magnitude of the Gradient

Fig. 10. Resultant image after thresholding (T =40)

(a) (b)

Fig. 11. Segmentation based on different threshold value a) T =40 and b) T = 80



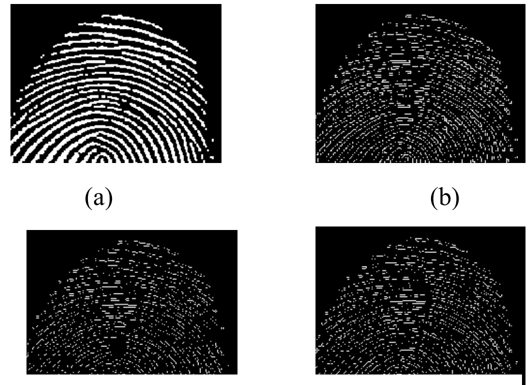
(a) (b)

(c) (d)

Fig. 12. First order edge detector for medical image a) Input image, b) Roberts, c) Prewitt, d) Sobel

In scenario two, convolution operation is performed in a sequential manner whereas thresholding is done in a parallel manner. Different threshold values are selected say T_1 to T_n. The output of the convolution operator is given to thresholding operator to determine the edges based on T_i value. Each Processor (P_i) uses a threshold value (T_i) for the output data of convolution operator. Fig. 11 a) - b) shows the segmented image based the threshold value 40 and 80 respectively. From the figures, it is inferred that threshold value 40 gives better segments than 80.

The edge detection is applied in many fields, to test the proposed PED algorithm. Two different image sets namely, medical and biometric images are processed and their results are shown in Fig. 12 and Fig. 13 respectively.



(a) (b)

(c) (d)

Fig. 13. First order edge detector for biometric image a) Input image, b) Roberts, c) Prewitt, d) Sobel

B. Performance Evaluation for three Scenarios

The performance of the proposed algorithm is analyzed in the following Section IV B a) to IV B c) for the three scenarios mentioned in Section IV.

a) Scenario 1: Convolution operation done in parallel

First level of parallelism is used as mentioned in III A. The sequential and parallel implementations are tested for the input image shown in Fig. 6 a) and the execution time of the algorithm is compared with different masks namely, Robert, Prewitt and Sobel. Fig. 14 shows the execution time of the three different masks in sequential (N =1) and parallel processor (N varies from 2 to 64). It is inferred from the graph that the execution time decreases when the number of processors is increased.

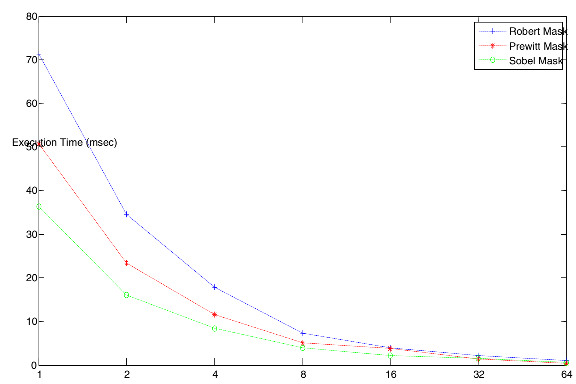


Fig. 14. Execution time for three different masks with N number of processors

b) Scenario 2: Thresholding operation done in parallel

As discussed in III B, thresholding operation is performed in parallel manner with different threshold values. The sequential and parallel implementations are done and Fig. 6 a) is tested using Sobel mask. Plot is drawn between execution time and number of processor as shown in Fig. 15.

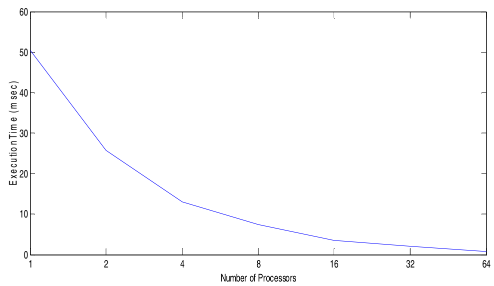


Fig . 15. Execution time of thresholding operation for Sobel mask N processors (N = 1 to 64)

Different threshold values from 0 to 255 are considered for testing the implemented model in sequential and parallel manner. The computation time of the parallel processor (with 16 processors) is less than the sequential one which can be inferred from Fig. 16. If the number of thresholds used are more, then the sequential model takes more time to obtain the result , whereas parallel algorithm is much faster depending on the number of processors.

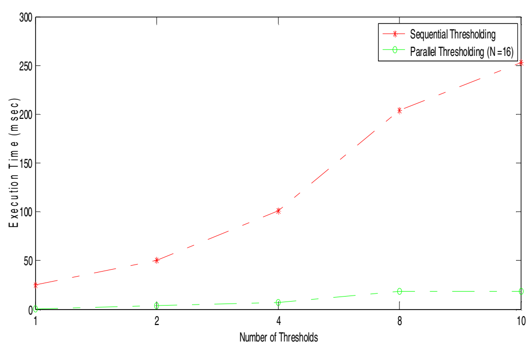


Fig . 16. Execution time of different threshold values to perform thresholding in parallel manner by keeping number of processors (N =16) as constant

c) Scenario 3 : Both Convolution and Thresholding operations done in parallel

Scenario 3 uses two levels of data parallelism, one for convolution operation and other for thresholding operation as explained in Section III. The execution time of the parallel model with varying number of processors (2, 4, 16, 32, 64) is less than the sequential model (with number of processor as 1) for the proposed PEDs which can be inferred from the graph Fig.17. From Fig. 18, it is inferred that the computational time of the proposed algorithm can be still reduced by increasing number of processors.

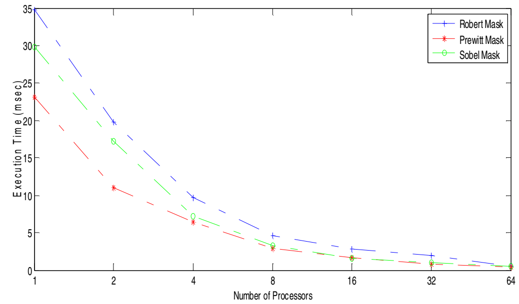


Fig . 17. Execution time of three different masks to perform convolution and thresholding in a parallel manner by varying number of processors from 1 to 64

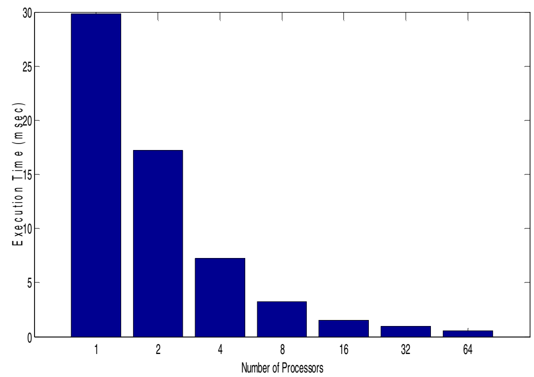


Fig. 18. Execution time of Sobel mask with number of processors varying from 1 to 64

V. CONCLUSION

Image segmentation using conventional edge detections require large computational time to handle enormous amount of data and instruction. To overcome this, an algorithm is proposed in parallel environment using Exclusive Read Exclusive Write Shared Memory Single Instruction Multiple Data (EREW SM SIMD) parallel architecture model. The proposed algorithm using different masks such as Sobel, Prewitt, Robert runs faster and gives better results than conventional methods. From the experimental results, it is inferred that the computational time of the algorithm decreases when number of processors increase which suits most of the real time applications. In future, the time can be still reduced by using multilevel data parallelism.

REFERENCES

[1] Lamine Ben Cheikh, T., Beltrame G., Nicolescu G., Cheriet, F. and Tahar, S. "Parallelization Strategies of the Canny Edge Detector for Multi-core CPUs and Many-core GPUs," in NEWCAS, Montreal, QC, 2012 , pp. 49 – 52.

[2] Varun Sanduja and Rajeev Patia, "Sobel Edge Detection using Parallel Architecture based on FPGA," Int. Journal of Applied Information Systems (IJAIS), vol. 3, no. 4, July 2012.

[3] Siddhartha Bhattacharyya. "A Brief Survey of Color Image Preprocessing and Segmentation Techniques," Journal of Pattern Recognition Research, 1[online], 2011, pp. 120-129. Available: http://www.jprr.org/index.php/jprr/article/viewFile/191/96

[4] Lei Yang, Dewei Zhao, Xiaoyu Wu, Hui Li and Jun Zhai, "An Improved Prewitt Algorithm for Edge Detection Based on Noised Image," in CISP, Shanghai, Oct. 2011, vol. 3, pp. 1197 – 1200.

[5] Wenshuo Gao, Xiaoguang Zhang, Lei Yang and Huizhong Liu, "An Improved Sobel Edge Detection," in ICCSIT, Chengdu, July 2010, pp. 67 – 71.

[6] Adrian N. Evans and Xin U. Liu, "A Morphological Gradient Approach to Color Edge Detection," IEEE Trans. Image Processing, vol. 15, no. 6, pp. 1454-1463, June 2006.

[7] M. Fathy, and M. Y. Siyal, "An image detection technique based on morphological edge detection and background differencing for real-time traffic analysis," Pattern Recognition Letters, 16, 1995, pp. 1321-1330.

[8] Prabir KR. Biswas, S.S. Biswas and B.N. Chatterji, "An SIMD algorithm for range image segmentation," Pattern Recognition, vol. 28, issue 2, Feb. 1995, pp. 255–267.

[9] Yong-Ren Huang and Chung-Ming Kuo, "Image Segmentation Using Edge Detection and Region Distribution," in CISP 2010, pp. 1410-1414.

[10] Ganesan, P., Rajini, V., Rajkumar, R.I., "Segmentation and Edge Detection of Color Images Using CIELAB Color Space and Edge Detectors," in INTERACT, Chennai, Dec. 2010, pp. 393 – 397.

[11] Sathish Kumar, S., Moorthi, M., Madhu, M. and Amutha, R., "An Improved Method of Segmentation Using Fuzzy-Neuro Logic," in Computer Research and Development, Kuala Lumpur, May 2010, pp. 671 – 675.

[12] Baydoun, M., Al-Alaoui, M.A. and Ferzli, R., "Parallel edge detection based on digital differentiator approximation," in ICCIT, Beirut, June 2013, pp. 371 – 375.

[13] Parminder Kaur and Ravi Kant, "A Review on: Comparison and Analysis of Edge Detection Techniques," Int. Journal of Engineering Research and General Science vol. 2, issue 3, Apr. 2014.

[14] Othman, A.A. and Tizhoosh, H.R. "Image classification using evolving fuzzy inference systems", IEEE Trans. Fuzzy Systems, vol. 22, No.1, Feb. 2013, pp. 1435 – 1438.

[15] Gupta, K.G., Agrawal, N. and Maity, S.K., "Performance analysis between aparapi (a parallel API) and JAVA by implementing sobel edge detection Algorithm," in PARCOMPTECH, Bangalore, Feb. 2013, pp. 1- 5.

[16] Bunil Kumar Balabantaray, Panchanand Jha and Bibhuti Bhusan Biswal. (2013, Dec.). "Application of edge detection algorithm for vision guided robotics assembly system," presented at Proc. SPIE 9067, ICMV 2013, 906713.[online] Available: http://dx.doi.org/10.1117/12.2051303

[17] R. C. Gonzalez and R.E. Woods, Digital Image Processing, Prentice Hall, New Jersey, 2008.

[18] S.G. Akl, The Design and Analysis of Parallel Algorithms, Prentice Hall of India, India, 1990.

[19] Beant Kaur, Anil Garg, "Comparative study of different edge detection techniques," International Journal of engineering science and technology. 3, p. 1927 ,2011.

[20] Salem Saleh Al – amri., Kalyankar, Khamitkar, N.,V. "Image segmentation by using edge detection," International Journal on computer science and engineering 2, p. 804 , 2012.

[21] Senthilkumaran, N., .Rajesh, R., "Edge detection techniques for image segmentation – A survey of soft computing approaches," International Journal of recent trends in engineering 1, p. 250 , 2009.