Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 7, No. 2, May 2016), Pages: 139-148
www.jacr.iausari.ac.ir

A Parallel Implementation of Modified Fuzzy Logic for Breast Cancer Detection

Farnaz Hoseini✉1, Asadollah Shahbahrami2, Anaram Yaghoobi Notash1, Peyman Bayat3

1) Department of Computer Engineering, Rasht Branch, Islamic Azad University, Rasht, Iran
2) Department of Computer Engineering, Faculty of Engineering, University of Guilan, Rasht, Iran
3) Department of Computer Engineering, Collage of Engineering, Rasht Branch, Islamic Azad University, Rasht, Iran

farnazhoseini@iaurasht.ac.ir; shahbahrami@guilan.ac.ir; anaramyaghoobi@iaurasht.ac.ir; bayat@iaurasht.ac.ir

Received: 2015/10/09; Accepted: 2016/05/16

Abstract

According to world health organization, breast cancer is one of the most deadly cancers occurred in women. Therefore accurate diagnosis and prediction is important to decrease the high death rate. The aim of this paper is twofold. First, improving breast cancer detection accuracy using Modified Fuzzy Logic (MFL) then improving the performance of MFL algorithm using GPU platform. The experimental results show that the accuracy of the breast cancer detection using MFL is higher than other techniques. In addition, by exploiting loop-level parallelism and pipeline parallel communication pattern in MFL algorithm, its performance is improved up to 19.17x for different image sizes.

Keywords: *Edge Detection Algorithms, Fuzzy Logic, Breast Cancer, GPU, MATLAB, Parallel Computing.*

1. Introduction

Cancer is responsible for one in eight deaths all over the worldwide [1], and most common of them in women is breast cancer [2]. Studies showing that although breast cancer incidence is high but the cause of breast cancer are still unknown [3]. As no effective way to prevent exists until now therefore, early detection is important for treating breast cancer. Most common diagnostic methods for detecting breast cancer are mammography and sonography [4, 5], but they have limitations in breast tissue density differentiation. MRI can provide architectural features of a breast lesion in several orientations, and improve analyzing the characteristics of the lesion [6]; therefore most of physicians prefer use MRI.

Edge detection is one of the crucial tasks in image processing, because it is usually the first step, performed before other image processing tasks such as image segmentation, boundary detection, object recognition, and classification [1]. While edge detection can be achieved by various traditional approaches, such as Sobel, Prewitt, and Roberts, they do not carry out edge correction correctly and are very sensitive to noise [2]. Objective of medical edge detection is to identify different regions, organs and anatomical. Today different edge detection algorithms have been vastly used for segmentation of most medical images [7]. Fuzzy logic techniques have found widespread application in image-understanding such as detection of edges, feature extraction, classification, and clustering [3]. For this reason we used fuzzy logic in the field of computer science for diagnosis of breast lesions on MRI images.

Most of application with sequential algorithm can no longer rely on technology scaling to improve performance, while image processing applications with high degree of parallelism are ideally excellent source for multi-cores platform. Major aim of parallel processing is not only improving high performance of images but also giving a solution to reduce processing time and better utilization of resources [4]. Studies have been shown that the bottleneck of limited processor speed affects the image processing algorithms in software implementation [5]. The past few years, microprocessor design has followed two separate paths: on the one hand, there were the multi-core multiprocessors (e.g. Intel Core i7 is a microprocessor with four processing cores, each of them implementing a complete x86 instructions set), On the other hand, the many-core microprocessors (e.g. graphics processors) developed mainly by the video-card manufacturers [6]. The recent opening of Graphics Processing Units (GPU) use NVIDIA CUDA application programming interface and offer a powerful platform with parallel calculation capabilities [7]. In February 2007, NVIDIA released the CUDA programming model to be used with their GPUs to make them available for general purpose application programming which is based on an extended ANSI C language and a runtime environment and allows the programmer to specify explicitly data parallel computation [8]. Since the GPUs are mainly based on multicores general-purpose processors and can provided a vast number of simple, loop-parallel, deeply multithreaded cores and high memory bandwidths [14], we choose it as a suitable tool for parallelism.

As fuzzy edge detection compared with the other traditional edge detection methods, can remove impulsive noise and smooth Gaussian noise [15] and with regard to positive effects of it [16] we used Modified Fuzzy Logic (MFL) algorithm to improve edge detection and noise reduction. In addition, the MFL is computationally intensive. Based on we have parallelized the MFL algorithm using GPU platform by taking profit from CUDA technology in Matlab environment. The speedup of GPU implementation over scalar implementation of up to 19.17x is achieved for different MR image sizes.

This paper is organized as follows. Related work is given in section II. The methodology of this paper is presented in section III which is involves explaining about the proposed edge detection enhancement using MFL algorithm and accelerating this using GPUs in Matlab. Then experimental evaluations are discussed in section IV. Finally conclusions are presented in section V.

2. Related Works

GPUs are emerging as platform of choice for parallel high performance computing and are also effective in data intensive parallel processing with availability of software development platforms such as Compute Unified Device Architecture (CUDA). Basic goal of CUDA is to help programmers focus on the task of parallelization of the algorithms rather than spending time on their implementation. In [18] the implementation of Sobel edge detection filter on GeForce GT 130 on MAC OS using CUDA and OpenGL was reduced. They got more speedup and less coalesced memory (zero or less) compared the original Sobel filter for various input images. In [19] canny edge detection algorithm implemented on CUDA demonstrated. The implementation of canny edge detection algorithm on CUDA achieved a speedup factor of 61 over a conventional software implementation. Another study showed that using CUDA framework, implement the entire canny algorithm compared with CPU implementations. They also integrated their detector into Matlab[20]. Another study aimed to improve processing time on larger edge detection filters [21]. Some researchers have used two parallelization strategies, loop-level parallelism and domain decomposition.

3. The Proposed Algorithm

Edge changes in image luminance or color values represent important regions in image. According to the characteristics, edges are often detected by derivative-based operators, such as first-order derivative methods, based on gradient or directional derivatives and second-order derivative methods, for example Laplacian operator. Edge detectors can often be implemented by means of convolution, either by a basic convolution or by a separable convolution, as many edge-detecting filters have a separable kernel [22]. At the beginning, vertical, horizontal, and directional derivatives are determined for image and obtained images are regarded as algorithm entry. Then, necessary adjustments for reinforcement of the intended edges and elimination of additional lines and points are carried out at later stages. Conditions of these adjustments are the main part of this paper that produce suitable results compared to other methods. This adjustment includes reinforcement of weak edges and elimination of additional points and lines (that are defined by if-then relationships.) Finally, the results of algorithms are compared with other methods. The proposed MFL algorithm is defined in some steps which are shown in Figure 1. These steps are explained in the following.

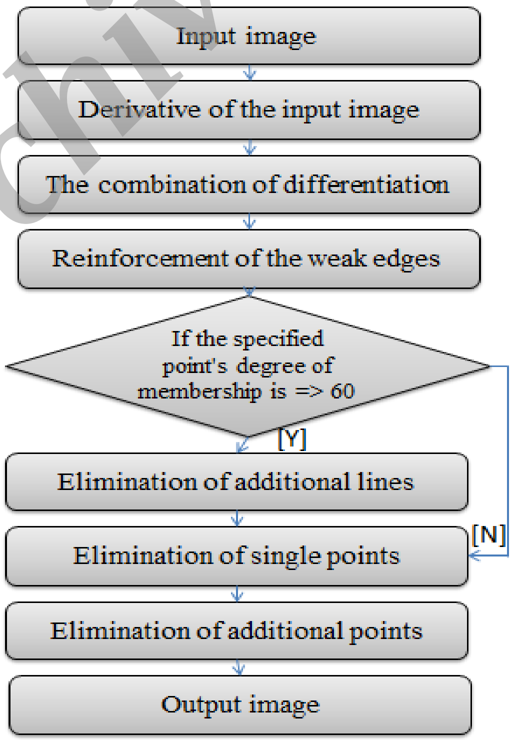
Input image
Derivative of the input image
The combination of differentiation
Reinforcement of the weak edges
If the specified point's degree of membership is => 60
[Y]
Elimination of additional lines
[N]
Elimination of single points
Elimination of additional points
Output image

Fig 1. Block diagram of the proposed MFL algorithm.

Step1. Input image: if a colorful image use as an input image, it should first convert into gray image. Gray images have only one layer with different degrees of gray color. One sample image is depicted in Figure 2.

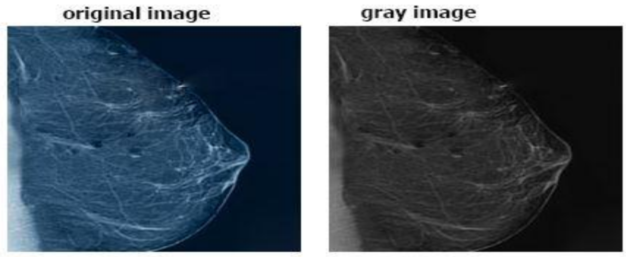

Fig 2. A ***sample image, original and gray images.***

Step2. Derivative of the input image: Several edge detection operators' work on basis of the first derivative of light intensity; that is, gradient of light intensity of main data is deal with using this information. It can be seek an image for peaks of light gradient. If I(x) represents the light intensity of pixel x and I′(x) of the first derivative (gradient of light intensity) of pixel x, that is shown in (1).

$$I'(x) = -1.I(x-1) + 0.I(x) + 1.I(x+1) \tag{1}$$

Derivatives in 4 directions on the main input image are shown in Figure 3.

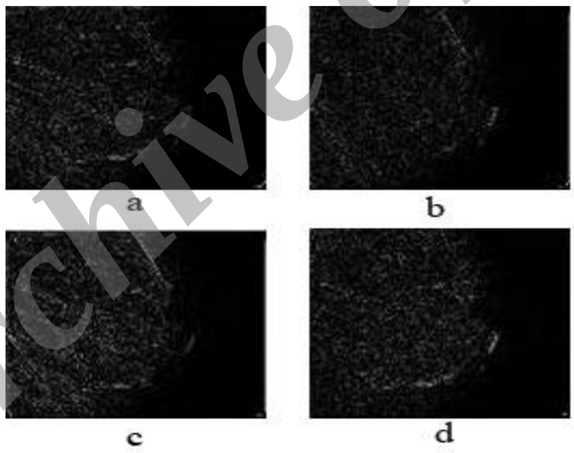

Fig 3. Derivative in a) horizontal, b) vertical, and c),d) oblique directions on the main input image.

Step3. The combination of differentiation: Four images obtained from derivation are combined by maximum operator in order to combine and get an image of fuzzy input. It means that for each pixel, there is a degree of edge membership in four images. Now, the maximum values of fuzzy membership occurred for each pixel is considered as a degree of its edge. Raw edges have been prepared for the next stage. The necessary adjustments is applied for reinforcement of intended edges and elimination of additional lines and points. We have defined 60 membership functions. These membership functions are calculated with binary edge range between 0 and 255. For functions, "and" and "or" has been used respectively for minimum and maximum. They selected for each modification using thresholding. It means that using each deduction rule, obtained fuzzy sets have joined input data through an additive function. thresholding helped to choose the type of modification. Edge modification will have done with two kernels that they moved on image. Output of the system has been calculated as a membership function and a binary image has been produced by the output condition, in which white pixels indicate edge and black pixels are areas without edge.

Step4. The first modification, reinforcement of the weak edges and elimination of additional lines: For the first edge modification, a 5*5 kernel and a 3*3 kernel is moved on image, and in the case of meeting the condition, two pixels greater than middle pixel and two pixels at the left of middle pixel of the kernel have the degree of fuzzy membership of the edge. One of the conditions is that if the fuzzy membership's degree of upper left point, the middle point and right lower point is greater than experimental threshold 60, this edge must be reinforced. For this reason, point of two pixels upper than middle point and two pixels at the left middle point take the label "edge" and the intended edge is developed and reinforced. In the case of not meeting the condition, the additional line is eliminated. Figure 4 shows a black edge, conditions, and results.

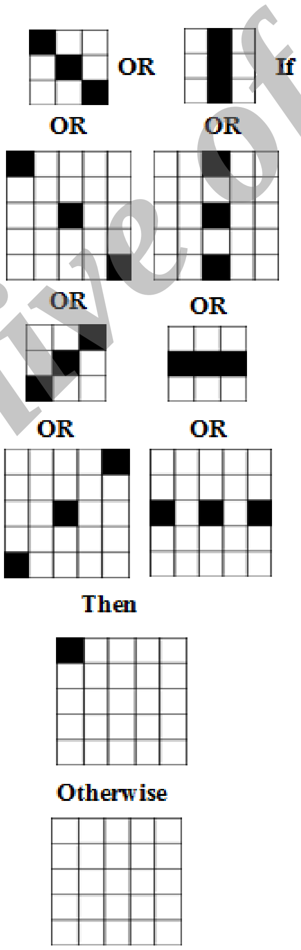

Fig 4. Rules of the first modification database.

Step5. The second modification, elimination of single points: The image obtained from the first adjustment is examined by the rules of fuzzy database and a kernel 5*5 in the third modification and single points of image are eliminated. Single points are those ones that do not have edge around them. Figure 5 shows the rules of database.

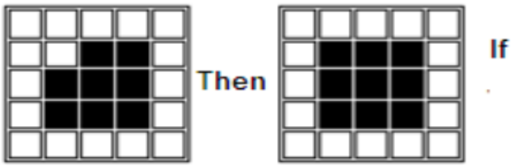

Fig 5. Rules of the second modification database.

Step6. The third modification, elimination of additional points: The image obtained from the second modification is examined by the rules of fuzzy database and a kernel 2*2 in the third adjustment and additional points are eliminated. Additional points are those ones smaller than 4 pixels. Figure 6 shows these conditions and related results.

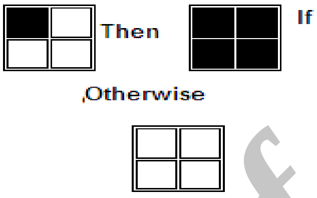

Fig 6. Rules of the third modification database.

In addition to achieve accurate edge of medical images using the proposed algorithm in Figure 1, other side of this paper is parallel implementation of this algorithm. This is because the MFL algorithm is computationally intensive especially for large image sizes. In order to improve the performance of the algorithm we have paralyzed it on GPU platform. There are different techniques for parallel implementation of edge detection algorithms such as data-level, loop-level, task-level, and pipeline parallelism. Loop-level parallelism has been used in this paper. In order to be able to process an image on the GPU; the corresponding data first had to be copied from main CPU memory to GPU memory. In CUDA, our programs are known as kernels. Kernels have a single program multiple data programming model. Our kernels were run on a grid, which is an array of blocks. Blocks are mapped into GPU architecture, and each thread is mapped into single processor. Pipeline is the most common parallel pattern that is used in Matlab environment. Pipelining involves each Program IDentity (PID) receiving data, processing it, and then sending it to another PID, as it is depicted in Figure 7. We used pipeline parallel pattern in this paper, because we had a set of data processing elements connected in series, where the output of one element is the input of the next one. The elements executed in time-sliced fashion; as a result the amount of buffer storage is often inserted between elements.

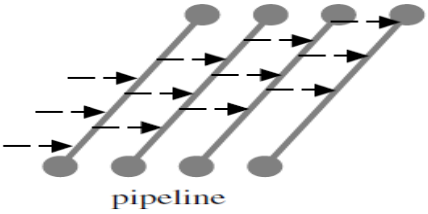

Fig 7. The structure of pipeline parallel pattern.

4. Experimental Results

The experimental platform based on NVIDIA GeForce series 9500GT was used in the experiment. MATLAB R2012a, CUDA 5.5, and Windows 7, 64 bit operating system have been used. The MFL algorithm has been compared to other algorithms for different images. Results of the traditional edge detection algorithms on the main input image are shown on Figure 8 that none of them has a good performance in this image and discontinuous lines are visible on all images. Canny method has regarded projections as an edge.

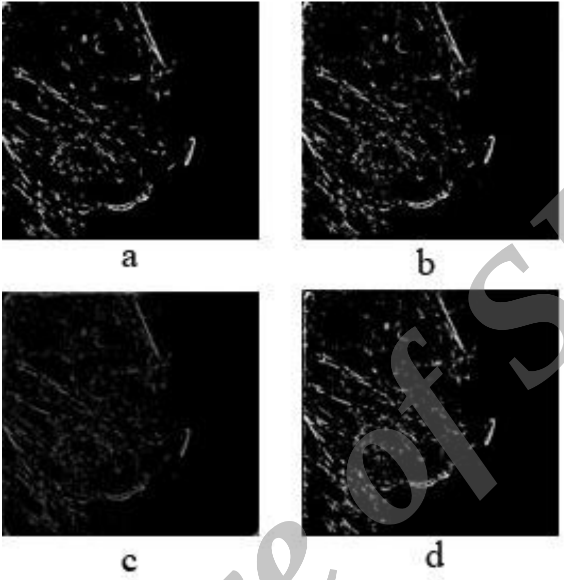

Fig 8. Result of the traditional edge detection algorithms; a) Canny, b) Sobel, c) Prewitt, and d) Roberts.

Weak and strong edges have been extracted in the first modification. Two of weak edges are marked with red circles on Figure 9. Furthermore, these edges may include undesirable points and additional edges. Single points have been eliminated in the second modification.

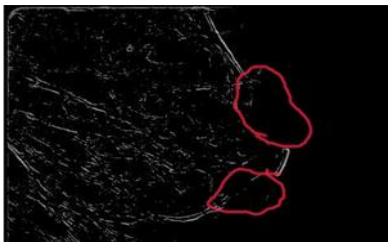

Fig 9. Result of weak edges extraction in the first modification.

As seen in Figure 10, the weak edges have been eliminated in the third modification and thickness of edges has been reduced. Thick lines become thin in the third modification and only main lines remain. Moreover, scattered lines in image, which are not connected to the main lines, are eliminated. This modification can be used to extract the main edges. For having all edges, either weak or strong, the third modification cannot be applied.

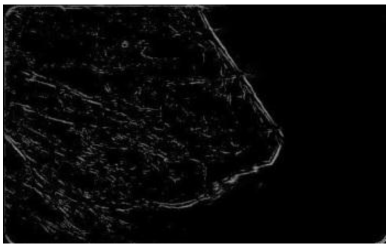

Fig 10. Result of the MFL algorithm method in the third modification.

Different image sizes have been used for our evaluation. Some of the images are taken from the database on the Internet from Digital Database for Screening Mammography (DDSM) [22] and some of them are taken manually. There are 132 different sizes of breast cancer images in our dataset. We have already been used the fuzzy inference system for image edge detection and the experimental results show that the performance is improved for different image sizes of up to 11.8x [23]. In this paper by using MFL algorithm the performance is improved much better than last work. We executed the implemented algorithms many times independently and measured the smallest execution time. In parallel implementations, the number of threads in each block is fixed to 512. The speedup of GPU implementation over the CPU implementation is depicted in Figure 11. As can be seen in this figure the speedup of up to 19.17x is achieved. In addition, when image sizes are increased, speedup is also increased.

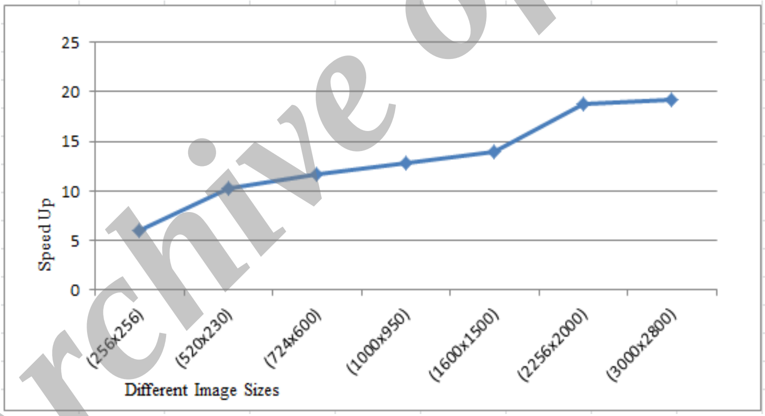

Fig 11. Speedup of GPU implementation over CPU implementation for MFL algorithm using different image sizes.

5. Conclusions

This work presented Modified Fuzzy Logic (MFL) algorithm that is more effective than the other traditional edge detection algorithms in terms of exploiting different edges. This technique eliminates noises, extra and weak edges. Its accuracy is higher than other traditionally edge detection algorithms, while it is computationally intensive. In order to improve the performance of MFL algorithm, it has been implemented on GPU platform. Loop-level parallelism of the algorithm has been exploited. Our experimental results using different image sizes show that the speedup of up to 19.17x is achieved.

References

[1] Thun Guzman, B. and Fortier, M., "Global cancer facts & figures" Atlanta, GA: American cancer society vol. 1,No 3, pp. 52 , 2007.

[2] Sivaramakrishna, R., Powell, K.A., Lieber, M.L., Chilcote, W.A. and Shekhar, R. , "Texture analysis of lesions in breast ultrasound images." Computerized medical imaging and graphics vol. 26, no. 5, pp. 303-307, 2003.

[3] S. Kathryn, L. Rehana A. Andrea T. Andrea Cheville, R. Smith, L. Lewis-Grant, C.J. Bryan, C.T. Smith, and Q. P. Greene, "Weight lifting for women at risk for breast cancer–related lymphedema: a randomized tria." Vol. 34, no. 24, pp. 2699-2705, 2010.

[4] S. Kathryn, L. Rehana A. Andrea T. Andrea Cheville, R. Smith, L. Lewis-Grant, C.J. Bryan, C.T. Smith, and Q. P. Greene, "Weight lifting in women with breast-cancer–related lymphedema." New England Journal of Medicine vol.36, no. 7,pp. 664-673,2009.

[5] N. J. Alison, and D. Boukerroui, "Ultrasound image segmentation: a survey.", IEEE Transactions on Medical Imaging, vol. 25, no. 8 pp. 987-1010,2006.

[6] N. Samina, H. Majeed and H. Irshad, "Image segmentation using fuzzy clustering: A survey." In 6th IEEE International Conference on Emerging Technologies, pp. 181-186., 2010.

[7] L. Zhang, M. Xiao, J. Ma and H. Song, "Edge Detection by Adaptive Neuro-Fuzzy Inference System," 2nd International Congress on IEEEImage and Signal Processing,pp. 1-4, Tianjin, 2009.

[8] Z. Dao-Qiang, and S. Chen, "A novel kernelized fuzzy c-means algorithm with application in medical image segmentation." artificial intelligence in medicine vol.32, no. 1,pp. 37-50,2004.

[9] E.B. Kumar and M. Sundaresan , "Edge detection using trapezoidal membership function based on fuzzy's mamdani inference system," IEEE International Conference on Computing for Sustainable Global ,New Delhi,pp. 515 – 518, 2014 .

[10] P. kaur , " Implemention of Image Processing Algorthms The Paralle Platform Using MATLAB," International Journal of Computer Science & Engineering Technology, vol. 4, No 6, June 2013 .

[11] K. Yadav, "Parallel Implementation of Similarity Measures on GPU Architecture using CUDA," Indian Journal of Computer Science and Engineering, vol. 3, No 1, February- March 2012 .

[12] R. Din, "Intellectual MATLAB Medical Images Classication on Graphic Processors," U.P.B. Sci. Bull., Series C, Vol. 75, No.2, 2013.

[13] V. Garcia, E. Debreuve and M. Barlaud, "Fast k nearest neighbor search using GPU," IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, June 2008.

[14] Krishnamurthy, S. Samsi and V. Gadepally, "Parallel MATLAB Techniques," Ohio Supercomputer Center and Ohio State University,Croatia, pp. 572, December 2009.

[15] J.C Bezdek, "Pattern Recognition with fuzzy Objective Function Algorithm," Plenum Press,1981.

[16] F. Farbiz, M.B. Menhaj, A. Motamedia and M.T. Hagan, "A New Fuzzy Logic Filter for Image Enhancement," IEEE Transactions on Systems, Man, and Cybernetics vol. 30, No 1, 2000 .

[17] H.J. Cai, X. Tian and T. Li, "A Multiscale and Anisotropic Edge Detection Algorithm," In International Conference on Logistics Engineering, Management and Computer Science, Atlantis Press, 2014.

[18] K. Ogawa and Y. Ito, K. Nakano, "GPU Efficient Canny Edge Detection Using a GPU,"1st International Conference onNetworking and Computing ,pp. 279 – 280,Higashi-Hiroshima,November2010 .

[19] Y. Luo and R. Duraiswami, “GPU Canny edge detection on NVIDIA CUDA,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 1-8,Anchorage, AK, June 2010 .

[20] Georgantzoglou, J.D.A Silva and R. Jena, “Image Processing with MATLAB and GPU,” Department of Oncology, University of Cambridge, UK , 2014.

[21] J. Kepner, “Parallel MATLAB for Multi-cores and Multinode Computers,” Society for Industrial and Applied Mathematics, 2009.

[22] M.Heath, K. Bowyer, D. Kopans, R. Moore and W.P. Kegelmeyer, “The digital database for screening mammography. InProceedings of the 5th international workshop on digital mammography,” pp. 212-218, 2000.

[23] F. Hoseini and A. Shahbahrami, “An efficient implementation of fuzzy edge detection using GPU in MATLAB,” IEEE International Conference of High Performance Computing & Simulation, pp. 605-610, 2015.