Improving Watershed Algorithm with a Histogram Driven Methodology and Implementation of the System on a Virtex 5 Platform

Nadia Smaoui Zghal
Computers imaging and Electronic Systems Group ICOS research unit,
ENIS, Sfax, Tunisia

Dorra Sellami Masmoudi
Computers imaging and Electronic Systems Group ICOS research unit,
ENIS, Sfax, Tunisia

ABSTRACT

Watershed algorithm as was introduced by Vincent and Soille is a segmentation algorithm based on the inundation process of the image gradient which is observed as a relief. It aims at finding the peaks in the image gradient called watersheds and identifying them as the image contours. Due to its flexibility and rapidity, this algorithm is used in several applications. However, its main drawback is the over segmentation .In this paper, we improve this technique by introducing a histogram driven methodology.

The developed architecture is applied on an empirical basis for research on image segmentation and boundary detection in order to be compared with other segmentation algorithms. The simulation results show that the performance of our algorithm is superior to the other segmentation techniques. Finally, the whole design is implemented on a Virtex 5 platform based on a co-design methodology leading to 147 MHz frequency and 76% of hardware resource occupation for an image of the size of 128*128.

General Terms

Segmentation, Co-design

Keywords

Watershed, Canny, K-means, OTSU, Virtex5

1. INTRODUCTION

Image segmentation is an essential process for most image analysis tasks aiming at decomposing an image into regions having visual similarity. Segmentation algorithms are grouped as histogram-based algorithms (exp: OTSU), edge based algorithms (exp: Canny operator), clustering (exp: K-means) and region based algorithms.

The watershed transform is a region-based segmentation approach coming from the field of mathematical morphology. It is used in several applications like object-based motion estimation [17], medical imaging [15], multi spectral satellite imagery [18], computer vision, etc. A flexible and fast algorithm for computing watersheds in digital grey scale images was described by Luc Vincent and Pierre Soille [11].It is applied on an image gradient which is obtained by applying an appropriate morphological gradient operator on a grey scale image. The image gradient is considered as a topographic surface. During the inundation process, the algorithm detects the regional minima of the gradients in the grey level image and increases these minima according to the gradient values. The watershed of a relief corresponds therefore to the limits of the adjacent catchment basins. [10].

The main advantage of this algorithm is the fact that with a moderate computational complexity, it is able to identify the important contours in a given image [2]. However, it possesses an important drawback which is the over-segmentation due to the fact that watershed transformation finds out even gradient peaks of low values and identifies them as contour.

There are two kinds of approaches to reduce this problem. The first one is based on extending the watershed algorithm to process with the markers in order to localize its operation on candidate contours regions as proposed in [10]. The second approach is based on a pre-processing task applied to the original image aiming at annealing small peaks in the gradient that can be suspected as contours as presented in [9]. In this paper, we propose a novel approach for solving over segmentation which is a histogram driven. The implementation step of watershed algorithm is considered as an important task, since this method is used in many applications. For optimization purposes, we will implement the developed architecture on a Virtex 5 platform based on a co-design methodology.

Using this methodology in such embedded system development makes it possible to meet strict design constraints such as power, size and timing [6].

This paper is organized as follows: section 2 presents our histogram based approach in watershed implementation for reducing over segmentation. The evaluation of our method compared to other algorithms and the simulation results are presented in section 3. Section 4 deals with the implementation of the proposed architecture. Finally, we end with a brief conclusion.

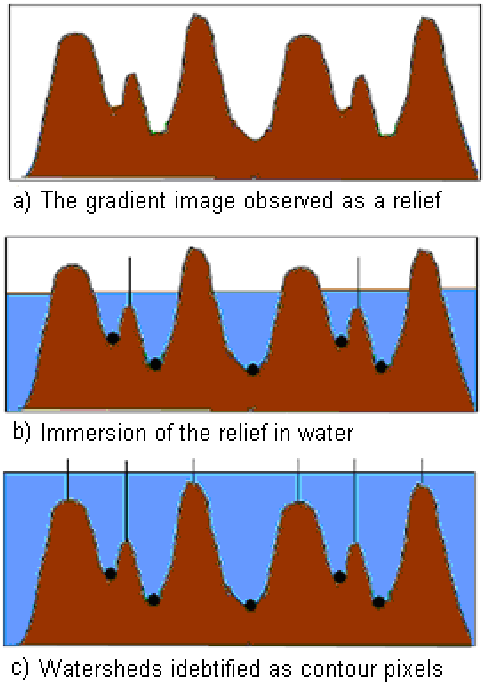
2. IMPROVING WATERSHED BASED SEGMENTATION

2.1 Watershed basic algorithm

In the field of mathematical morphology, the watershed transform as was introduced by Vincent and Soille is a segmentation algorithm based on the immersion principle [11].

This algorithm is applied to a gradient image considered as a relief. During the flooding process, watersheds with adjacent catchment basins are constructed in the following way. Imagine, the relief washed in water. As the water level rises, the basins, which are the lower intensity level in an image, are filled and water spills over the boundaries, growing those basins into larger ones. Once flooding process finished, watersheds are then defined as the peak forming the limit between two consecutive basins [7].

Thus, we can view a segmentation of an image (figure 1).



a) The gradient image observed as a relief

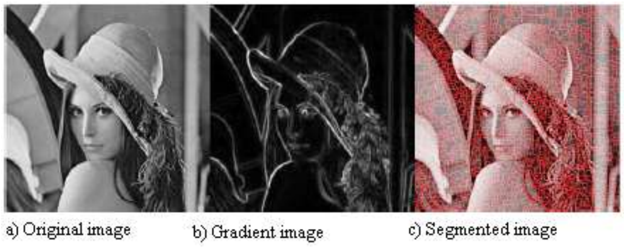
b) Immersion of the relief in water

c) Watersheds idebtified as contour pixels

Fig1. Immersion principle

Actually, the algorithm is operating in two steps. In the first one which is the sorting process, we sort the pixels of the gradient image in an ascending order of grey scale values and we extract the coordinates of all the pixels at each level. During the second process which is the flooding step and for each grey level, we make the following test: if a pixel has a labelled neighbour then it takes this label otherwise it takes a new label. Finally, if the considered pixel has two neighbours labelled differently, then the pixel is considered as a watershed [14].

We have applied this basic watershed algorithm on Lena image, the resulted image is presented in figure 2:



a) Original image

b) Gradient image

c) Segmented image

Fig2. Application of a watershed algorithm on Lena image

As can be shown, although this method is able to identify the closed contours of a given image, it suffers from over segmentation since it considers in a similar way lower and higher valued gradients. In the following section, we explain how to solve this limitation.

2.2 A histogram driven methodology

The over-segmentation problem is created due to the fact that watershed basic algorithm considers the low values of the gradient as watersheds. For that, we had the idea of ignoring areas of low gradient and giving them the same label. The algorithm will start therefore from a certain threshold. To define this threshold we will be based on the histogram of the image gradient. Since object contours correspond to high levels of gradient in the image, they will define high levels classes in the histogram with medium probability values. However, suspect gradient peaks leading to over segmentation problem belongs generally to object surfaces or backgrounds and corresponds accordingly to classes of larger probability in the histogram leading to high peaks for low intensity levels. The idea is to use a histogram driven methodology aiming at extracting thresholds for an image and then choose among them the optimal one. This method is the OTSU thresholding method. It is based on maximizing or minimizing the between-class variance of pixel intensity to perform picture thresholding. The pixels of an observed image are represented in L grey levels from 1 to L. Multilevel image thresholding consists in separating the pixels of the image in M classes C1,..,CM by setting the thresholds t1,.,tM[16][12].

For multilevel thresholding methods, the number of the classes M is obtained by the following equation [13]:

$$M = 1 + \frac{10\log(N)}{3} \tag{1}$$

In mathematical terms, the OTSU method can be described as follows: N is the number of the image pixels. For the placement of the thresholds, we use the histogram which is a statistic of the grey levels for the gradient image where:

$$\sum_{i=1}^{L} h(i) = N \tag{2}$$

The normalized histogram p (i) is calculated as follows:

$$p(i) = \frac{h(i)}{N} \tag{3}$$

$$\sum_{i=1}^{L} p(i) = 1 \tag{4}$$

As can be seen, p (i) can be interpreted as the probability to find the pixel level I in the whole image.

For all classes, statistical properties such as the probability of the class (referred to as the class weight) and the mean of the class can be calculated respectively as follows:

$$\omega(k) = \sum_{i \in C_k} p(i) \tag{5}$$

$$\mu(k) = \sum_{i \in C_k} ip(i) \tag{6}$$

Then, we define the mean intensity of the whole image as:

$$\mu_T = \mu(L) = \sum_{i=1}^{L} ip(i) \tag{7}$$

After that, the inter-class variance of a class Ck can be estimated as:

$$\sigma_B^2(K) = \frac{[\mu_T \varpi(K) - \mu(K)]^2}{\varpi(K)[1 - \varpi(K)]} \tag{8}$$

The thresholds are therefore obtained by the following equation:

$$\sigma_B^2(k) = \max_{1\le k\le L}(\sigma_B^2(k)) \tag{9}$$

Once the thresholds extraction finished, we analyze the histogram of the gradient image in order to extract the optimal threshold. To do that, we define a table having M boxes. In each box, we put the number of pixels which varies between two consecutive thresholds and we compute the average of the boxes. Then, we make a comparison between the different boxes. Finally, we choose the value situated just after the box having the number of pixels which is close to the average. This value will be considered as the optimal threshold.

Watershed algorithm is therefore applied starting from this point. Before this optimal threshold, we will just extend the basins. Figure 3 summaries the proposed procedure and figure 4 presents the contour obtained by our improved algorithm.

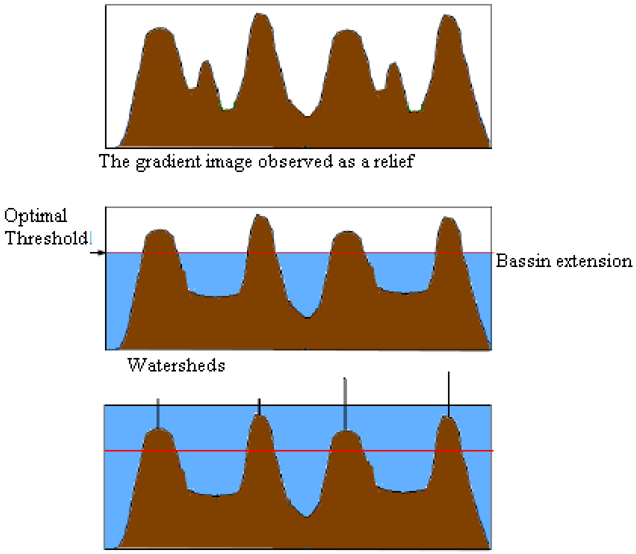


Fig3. Proposed Procedure

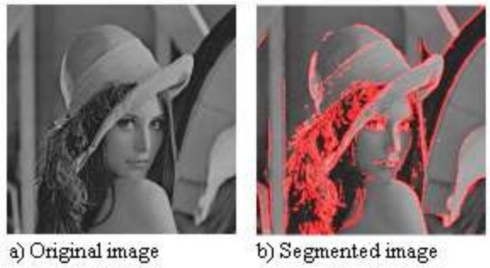


Fig4. Image segmentation

Since the threshold separates two sets of classes. Generally, we can be very strict and interested only by confirmed and clear contours, in such a case a high threshold is applied. But, we can also be very permissive by underlying almost all contours in the image. In such a case a low threshold is applied. In this way, we can play on the value of the threshold to extract all the details or just the strongest ones according to the requested application. Figure 5 shows three contours of Lena image obtained from different thresholds. We notice that the smaller the value of the threshold is, the clearer the contours are.

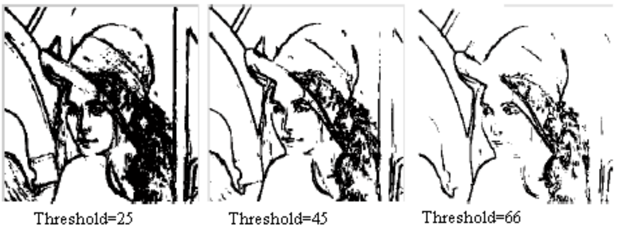


Fig5. Contours obtained from different thresholds

3. EVALUATION AND COMPARAISON:

For the evaluation of image segmentation many criteria are proposed. These criteria can be divided into two main categories: supervised evaluation and unsupervised evaluation.

Unlike the unsupervised evaluation , the supervised evaluation is based on evaluating the segmented image in comparison with a known reference called ground truth which is created manually by experts[5].Figure 6 illustrates the principle of the supervised evaluation.

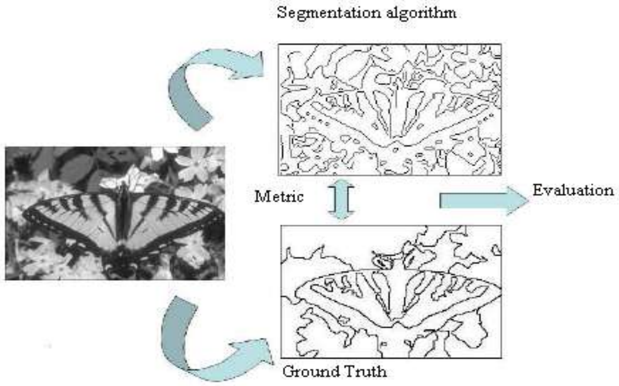


Fig6. Supervised evaluation procedure of a segmentation result

For the evaluation of the improved algorithm, we are based on the Berkley database which is an empirical basis for research on image segmentation and boundary detection [21]. Figure 7 shows some images selected from each category in this database and their corresponding ground truth.

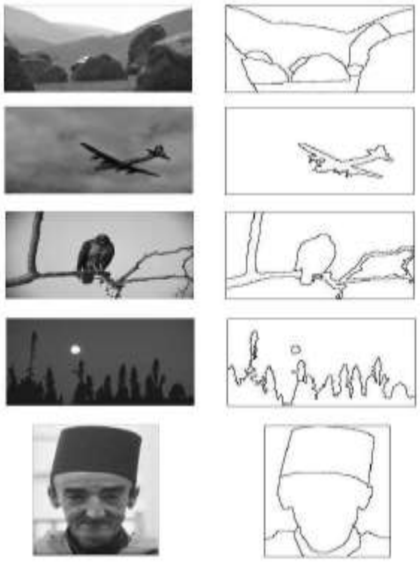


Fig7. Some images selected from the Berkley database and the proposed contours

To evaluate segmentation algorithms, we use three error measures. Let Iref be the reference contour corresponding to the ground truth and IF the detected contour obtained through the segmentation result of image I. The ODE (over detection error) corresponds to detected contours which do not coincide with Iref. The UDE (under detection error) corresponds to pixels of Iref which have not been detected. The LE (localization error) computes the distance between the misclassified pixels and the nearest pixels of Iref. Segmentation is considered good if the three errors are all small [19].

- Under detection error

$$UDE = \frac{card(I_{\mathrm{Re}\,f/F})}{card(I_{\mathrm{Re}\,f})} \tag{10}$$

- Over detection error

$$ODE = \frac{card(I_F) - card(I_F \cap I_{\mathrm{Re}\,f})}{card(I) - card(I_{\mathrm{Re}\,f})} \tag{11}$$

- Localization error

$$LE = \frac{card(I_{\mathrm{Re}\,f/F} \cup I_{F/\mathrm{Re}\,f})}{card(I)} \tag{12}$$

3.1 Comparison with the basic Watershed algorithm

In the first step, we evaluate our improved algorithm by comparing it to the basic one implemented in Matlab. The performance measures obtained for 100 images in the Berkley database are presented in table1 and the application of these two methods on 5 images selected from each category in the Berkley database is presented in figure 8. Starting from table 1, we notice that our algorithm was well improved compared to the basic one since for the three error measures, values are lowered considerably. (In table 1, values in bold indicate best result).

Table 1. Performances measure obtained from the watershed basic algorithm and our improved algorithm

Category	Error Measure	Basic Watershed	Our algorithm
Landscape	Over	0.9486	**0.8071**
Landscape	Under	0.0516	**0.0115**
Landscape	Loc	0.4768	**0.1146**
Object	Over	0.9599	**0.8275**
Object	Under	0.0540	**0.0067**
Object	Loc	0.2122	**0.1208**
Animal	Over	0.9256	**0.7574**
Animal	Under	0.0945	**0.0289**
Animal	Loc	0.2907	**0.1752**
Nature	Over	0.9530	**0.7064**
Nature	Under	0.0461	**0.0064**
Nature	Loc	0.1321	**0.0763**
Portrait	Over	0.9511	**0.8236**
Portrait	Under	0.0634	**0.0193**
Portrait	Loc	0.2396	**0.1231**
Average	Over	0.9476	**0.7844**
Average	Under	0.0619	**0.0145**
Average	Loc	0.2703	**0.1220**

3.2. Comparaison with other segmentation algorithms

In the second step, we apply canny operator [23] and K-means algorithm [1] considered as efficient segmentation algorithm on the images selected from the Berkley database in order to make a comparison between those methods and our improved algorithm.

3.2.1. Canny operator

The traditional canny operator is based on the following processing steps:

1. Noise reduction:

2D Gaussian filtering and the original image are convolved to reduce noise. The Gaussian filtering function is represented in the following equation:

$$G(x,y,\sigma) = \frac{1}{2\Pi\sigma^2} \tag{13}$$

The result is a slightly blurred version of the original image.

2. Finding the intensity gradient of the image:

The edge detection operator (such as Prewitt operator, Sobel operator, and etc) is used to calculate the horizontal direction Gx and the vertical direction Gy of the grey level image. From this, the edge gradient and direction can be determined:

$$G = \sqrt{Gx^2 + Gy^2} \tag{14}$$

$$\theta = \arctan(\frac{Gy}{Gx});\theta = (0,45,90,135) \tag{15}$$

3. Non-maximum suppression:

Once the edge gradient and direction are obtained, a search is carried out to determine whether the gradient magnitude assumes a local maximum in the gradient direction. When compared to the two pixels along the gradient direction and if the grey-level of a certain pixel is not the greatest, then the value of the pixel is set to zero, indicating that this pixel is not the edge and the whole process is called non maxima suppression. From this stage, a set of candidate edge points, in the form of a binary image, is obtained.

4. Thresholding:

Dual-threshold algorithm is used to detect and connect edges. It has two thresholds: $\zeta 1$ and $\zeta 2$, where $2\zeta 1 \approx \zeta\ 2$ .Therefore, two output images $N1$ $(i;\ j)$ and $N2$ $(i;\ j)$ could be obtained respectively by the low and high threshold ones. $N2$ $(i;\ j)$ rarely contains false edges since it is obtained by the upper threshold. The dual-threshold algorithm is based on connecting the edges of $N2$ $(i;\ j)$ in order to form the contour. Once reaching the points of the contour, this algorithm will try to search for the edges that could be connected onto the contour at the 8 neighborhood locations of $N1$ $(i;\ j)$. The algorithm will continuously gather the edges of $N1(i;\ j)$ until$N2(i;\ j)$ is connected.

3.2.2. K-means algorithm

K-means clustering is a method of cluster analysis which aims at partitioning n observations into K clusters in which each observation belongs to the cluster with the nearest mean. The k-means clustering algorithm is commonly used in computer vision as a form of image segmentation.

The results of the segmentation are used to help border detection and object recognition[1].

The realization procedure of k-means algorithm is presented as follows:

1. We Initialize the (K) class centers. To simplify, an equal-distance method is used to define the initial class centers:

$$Center_i^0 = L\min + (i - \frac{1}{2})\frac{L\max - L\min}{k} \qquad (16)$$

Where i varies between 1 and *k; Center$_i^0$* is the initial class center for the ith class, *Lmax* and *Lmin* are respectively the maximum and minimum of the grey value.

2. We assign each point to its closest class center. The criterion to assign a point to a class is based on the Euclidean distance presented by the following equation:

$$Dis\tan ce_{i,j} = abs(Lj - Center_i) \qquad (17)$$

Where *Distance$_{i,j}$* is the distance from the j th point to the i th class, and N is the total number of points in the sample space

3. We calculate the (K) new class centers out of the mean of the points that are assigned to it by the following equation:

$$Center_i^m = \frac{1}{Ni}\sum_{j=1}^{Ni} Lj \qquad (18)$$

Where i varies between i and k and *Ni* is the total number of points that are assigned to the ith class in step 2.

4. We repeat step 2 until the class centers converge and remains unchanged.

Since K-means algorithm permits to obtain regions, we apply a gradient operator to the segmented image in order to extract borders.

The parameter k is chosen in order to have best result for the segmented image.

The application of Canny operator and K-means algorithm on the same images selected from the Berkley data base is presented in figure 8 and table 2 presents the performance measures obtained respectively from Canny operator, K-means algorithm and our improved watershed algorithm. In this table, values in bold indicate the best segmentation result.

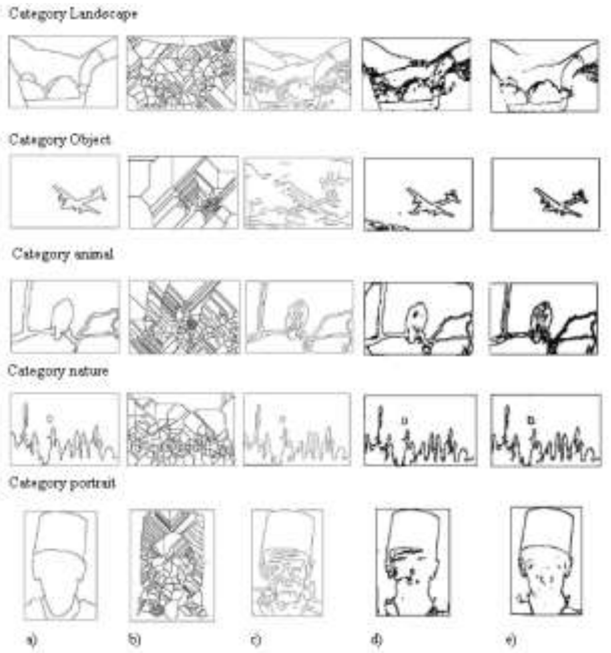


Fig 8. Image segmentation: a) Manual contours as proposed in the Berkley database b) Contours obtained by the basic watershed c)Contours obtained by Canny operator d)Contours obtained by K-means algorithm e) Contours obtained by our improved watershed algorithm

Table 2. Performance measures

Category	Error Measure	Canny	Kmeans	Water
Landscape	Over	0.8919	0.8592	**0.8071**
Landscape	Under	0.0279	0.0138	**0.0115**
Landscape	Loc	**0.1043**	0.1582	0.1146
Object	Over	0.8995	0.8301	**0.8275**
Object	Under	0.0207	0.0071	**0.0067**
Object	Loc	**0.0955**	0.1256	0.1208
Animal	Over	0.8163	0.7750	**0.7574**
Animal	Under	0.0573	0.0316	**0.0289**
Animal	Loc	**0.1220**	0.1761	0.1752
Nature	Over	0.8096	0.7862	**0.7064**
Nature	Under	0.0253	0.0090	**0.0064**
Nature	Loc	0.0784	0.1136	**0.0763**
Portrait	Over	0.8984	0.8796	**0.8236**
Portrait	Under	0.0316	0.0225	**0.0193**
Portrait	Loc	**0.1177**	0.1786	0.1231
Average	Over	0.8631	0.8260	**0.7844**
Average	Under	0.0325	0.0168	**0.0145**
Average	Loc	**0.1036**	0.1504	0.1220

Starting from table2, the values of the three error measures obtained by our algorithm are better than that obtained by K-means algorithm. Thus, our algorithm is better than K-means algorithm. In comparison with canny operator, we have better result for the under-detection and the over-detection errors. But, for the localisation-error, the better result is obtained by canny operator. We can, therefore, conclude that our algorithm is close to if not better than canny operator. So we are led to the conclusion that our method is efficient in image segmentation.

4. IMPLEMENTATION

4.1. Previous works

In the literature, several approaches have been proposed for the hardware realization of watershed transformation. In [2],we describe a watershed algorithm based on the flooding process. The implementation phase of the proposed architecture is based on ordered queues which add high complexity to the design. In fact, the synthesis results of the proposed architecture for Virtex FPGA show that the circuit runs at 55 MHz for an image of the size of 30*30.

In [4], we describe an implementation method of a watershed algorithm based on connected components on FPGA XC2V6000. In the proposed design, large data structures used in the algorithm are placed redundantly in the external memory banks in order to allow parallel access to them and are accessed as read or write only. In this way, eight pixels can be processed in parallel. The circuit runs at 86 MHz and occupies 9,8%of the slices and 119 blocks RAMs for an image of the size of 512*512.

[3] shows a pipeline implementation of a watershed algorithm aiming at propagating the value of each pixel to its neighbours. The implementation is based on processing w-sets of k-lines of an image as follows: a new set is taken from the external memory banks and an oldest one is shifted to other external memory banks until processing the whole image. The proposed system runs at 66 MHz and occupies 93% of the hardware resources of an FPGA XC2V6000.

The proposed architectures designed for hardware implementation are interested only in watershed transformation, considering as input the gradient image, due to the limitation of hardware resources in FPGA.

In this paper, we try to exploit the co-design methodology in implementing the whole system described in section 2 in a virtex 5 platform. In this platform, the Virtex 5 FPGA contains a PowerPC processor integrated for running software application while the surrounding FPGA fabric can be configured as hardware accelerators. The reason for using a software/hardware co-processing architecture is that it takes advantage of both sides. In fact, the Software solution offers more flexibility and a lower cost of implementation, while the hardware implementation is faster in processing [8].

4.2. Implementation of the proposed architecture on a Virtex 5 platform

The aim of our work is to reduce design time and reach better performances when implementing the system with hardware and software resources. For that, we use Co-developper of Impulse Accelerated technologies for design optimization. This tool is a C-to-HDL compiler which realises all the steps from initial design to final implementation of an algorithm [20]. In the first step which is the simulation, the source code of the design is described in ANSI C and then simulated using Impulse C. In the second step which is the partitioning, the source code is divided into software and hardware components. The software component consisting of the input and output interfacing is running on the target PowerPC processor. The hardware component is composed of the gradient operator, the OTSU method and the watershed algorithm. Then, the compiler is used to generate automatically the appropriate interfaces for both the hardware and the software components.

Finally, constraints and logic are mapped onto the target resources and a configuration bit stream is generated.

In our case, the Studio Design Kit from Xilinx is used for the generation of the bit stream file to configure the FPGA on a Virtex 5 platform [22]. We notice that for an image of the size of 128*128 the system runs at 147 MHz and occupies 76% of the hardware resources.

5. CONCLUSION

In this paper, an improved watershed segmentation algorithm with a histogram driven methodology is proposed.

The application of the novel approach on an empirical basis for research on image segmentation and boundary detection and the comparison of the resulted images with those obtained respectively by the basic watershed algorithm, Canny operator and K-means algorithm show the effectiveness of our algorithm. Moreover, this paper applies the co-design methodology in the implementation of the improved architecture on a Xilinx Virtex 5 development platform. The software application which is the interfacing of the input and output image are running on the Power PC processor. The hardware component, which is the gradient operator, the OTSU method and the watershed algorithm co-process with the Power pc to achieve the goals of acceleration. This strategy leads to acceptable hardware resources occupation (about 76%) and to a frequency performance of approximately 147 MHz for an image of the size of 128*128.

6. REFERENCES

[1] Bo Zhao, Zhongxiang Zhu, Enrong Mao (Corresponding author) and Zhenghe Song ”Image Segmentation Based on Ant Colony Optimization and K-Means Clustering” College of Engineering China Agricultural University Beijing, China 1-4244-1531- 4/07/25.002007*IEEE*.

[2] C. Rambabu, I. Chakrabarti and A. Mahanta ”Flooding-based watershed algorithm and its prototype hardware architecture” IEE Proc.-Vis. Image Signal Process., Vol. 151, No. 3, June 2004

[3] Dang Ba Khac Trieu and Tsutomu Maruyama”A PIPELINE IMPLEMENTATION OF A WATERSHED ALGORITHM ON FPGA” 1-4244-1060-6/07/25.00*C*2007*IEEE*

[4] Dang Ba Khac Trieu and Tsutomu Maruyama ”An implementation of a watershed algorithm based on connected components on FPGA” 978-1-4244-2796- /08/25.002008*IEEE* Japan,2008

[5] DO Minh Chau ”Evaluation de la segmentation d’images”,Institut de la francophonie pour l’informatique,Vitnam,2007

[6] G. F. Zaki, R. A. Girgis, W. W. Moussa , and W. R. Gobran, Junior Member, IEEE ”Using HW/SW Co-design to Implement an Embedded Face Recognition/ Verification System on an FPGA”

[7] H.S.Sheshadri and A. Kandaswamy”Detection of Breast Cancer Tumor based on Morphological Watershed Algorithm” Department of ECE, PSG College of Technology, Coimbatore-641004

[8] J.Wang ”An FPGA Based Software/Hardware Codesign for Real Time Video Processing” Technical University Munich, Germany,2006

[9] K.Karantzalos, D. Argialas ”Improving edge detection and watershed segmentation with anisotropic diffusion and morphological levellings” Remote Sensing Laboratory, School of Rural and Surveying Engineering (SRSE), National Technical University of Athens (NTUA), Iroon Polytechniou 9, 15780, Zografou Campus, Athens, Greece June 2006

[10] Li Gao , Shuyuan Yang , Jie Xia , Junli Liang , Yuhua Qin ” A new marker-based watershed algorithm” 1-4244-0549 *¡* 1/06/20.002006 IEEE. China 2006

[11] Luc Vincent and Pierre Soille ” Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations ” 0162-8828/91/0600-058301:0001991*IEEE*

[12] M. Eichmann, M.Lussi ”Efficient Multi level Image Thresholding” Rapperswil, December 2005

[13] Maurice Lethielleux, Statistique descriptive, ditions Dunod, Paris, 1999

[14] N. Smaoui Zghal, A. Yangui Jammoussi, D. Sellami Masmoudi ”Implementation of a watershed based image segmentation system in Virtex II pro platform” DTIS 2009, Cairo, Egypt 978-1-4244-4321-5/09/25.00*IEEE*

[15] P. Dillinger, J. F. Vogelbruch, J. Leinen, S. Suslov, R. Patzak, H. Winkler, and K. Schwan ”FPGA-Based Real-Time Image segmentation for Medical Systems and Data Processing” IEEE Transactions on nuclear Science VOL. 53, NO. 4, August 2006

[16] P.Sung Liao, T.Sheng Chen, P.Choo Chung ”A Fast Algorithm for Multilevel Thresholding” Journal of information science and engineering, 2001

[17] R.i, S.Yu ”Efficient Spatio-temporal Segmentation for Extracting Moving Objects in Video Sequences” Institute of Image Communication and Information Processing of Shanghai Jiao Tong University, Shanghai, P.R. July 2007

[18] S.Chen1, J.Luo1, Z.Shen1, X.Hu1, L. Gao1 ”Segmentation of Multi-spectral Satellite Images Based on Watershed Algorithm” Institute of Remote Sensing Application, Chinese Academy of Scienses, Beijing, 100101, China 2008

[19] S.Chabrier,H.Laurent, C.Rosenberger ”Supervised evaluation of synthetic and real contour segmentation results”,14 th European Signal Processing Conference, Italy,2006

[20] www.ImpulseAccelerated.com

[21] www.eecs.berkeley.edu/Research/Projects /CS/ vision/ grouping/segbench/
[22] www.Xilinx.com
[23] Xun Wang and Jian-Qiu Jin ”An Edge Detection Algorithm Based on Improved CANNY Operator”