

An Segmentation Under Connected Components Based on Watershed Algorithm Using FPGA Processor

R. Kiruthikaa¹, S. Salaiselvapathy²

¹M.E Applied Electronics, Electronics and Communication Engineering, Coimbatore

²B.E CSE, Computer Science and Engineering Accenture, Chennai, India

Abstract: *Watershed segmentation is a region based segmentation algorithm. It forms the peak joining two catchment basins containing regional minima. Watershed segmentation is more sensitive to noise and can leads to serious over-segmentation. In order to overcome the shortcomings of traditional watershed segmentation, this paper deals with marker controlled watershed algorithm based on connected components approach locating a regional minima. Mostly the watershed segmentation result in over-segmentation. The markers based watershed segmentation transformation reduces noise and over-segmentation. Watershed has been written in matlab code and also in VHDL hardware descriptive language. The main aim of this project is to implement in a FPGA which requires minimum hardware resources, low execution time and is suitable for use in real time applications.*

Keywords: Watershed, segmentation, threshold, region growing, edge detection

1. Introduction

Image segmentation is process of partitioning the image into multiple segments. It is first important step in many image processing applications like image analysis, image description and recognition, image visualization and object based image compression. The algorithm used in this project is watershed based image segmentation. It is a hybrid technique because it is the result of threshold based, edge and region based techniques using morphological watershed transform. Watershed algorithm is a region based segmentation algorithm. The basic concept of watershed algorithm is used to find the watershed lines containing regional minima. It defines the peak forming the limit between two catchment basins and the intercepting points constitute Watershed. The main drawback of watershed segmentation is over-segmentation, presence of noise, time consuming. This can be removed easily using marker controlled watershed segmentation. A marker is a connected component belonging to an image. The markers include the internal markers, associated with objects of interest, and the external markers, associated with the background.

Several approaches have been proposed for the hardware implementation of the watershed algorithm [1-...3]. In [1], a watershed hardware implementation derived from Meyer's simulated flooding-based algorithm is described. It is based on ordered queues which add high complexity in the design. The proposed architecture has been modelled in VHDL and synthesised for FPGA. The frequency performance of the implementation is limited to 55 MHz and the circuit occupies 64% of the hardware resources. In [2, 3], a pipelined and parallel watershed hardware implementation on FPGA has been described for an image of size 512*512. The circuit runs at 66 MHz and occupies 92% of the hardware resources.

Chuiig J. Kuo, SoulieilF. Odeh', & Mirig, C. Hi—aiig, proposed [9], the watershed transformation and the region

merging method based on neighbor regions where the edge value is employed. The simulation results show that the proposed algorithm in this paper can greatly alleviate the over-segmentation problem, but the edge locating is not very accurate.

L. Rittner, F. Flores, R. Lotufo [10] proposes New representation of Color Images where the Morphological Gradient Applied to Color Image Segmentation and the dissimilarity information is richer in color images than in grayscale ones, the design of methods to edge enhancement in color images is complex. Note that the metric that measures the natural dissimilarity information of color images is not known, it will be not natural for the human eye.

The future works on the segmentation method of the watershed algorithm is improved based on the marker controlled watershed image segmentation at the same time, locating a local minimum in the object for watershed transform.

2. Digital Image Processing

Classical image processing techniques [1] when applied in segmentation tasks present the difficulty that borders are left normally open (what we need are closed contours). Another problem is that distinct objects in the scene are assumed as a single object in the image (i.e., classical techniques have troubles identifying touching objects as separate entities). Those problems are compounded by noisy images, by uneven illumination, and by overlapping. Figure 1 illustrates some of the problems with classical image processing techniques. Fig. 1(a) shows a scene where the two rocks at the bottom left are touching. Conventional thresholding produces the image in Fig. 1(b), whereas a well-known edge operator produces the image in Fig. 1(c). The Watershed Transformation, also known as the lines dividing waters is a powerful tool for image segmentation.

The basic idea consists of segmenting images by simulating flooding on a topographic surface. Drilling a hole at each local minimum performs this process and then slowly submerging the entire surface into a lake. The water level will rise continuously. Then, in order to avoid that water coming from different holes merge, a dam is built at each point of contact. At the end of the process, the union of all those dams constitutes the watersheds.

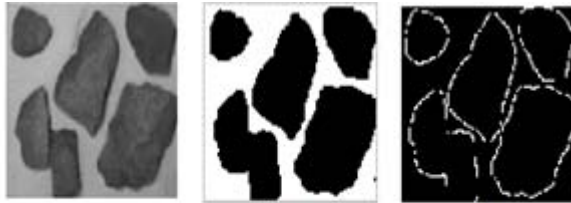


Figure 1: (a) Gray level image, (b) Thresholding, (c) Edges

Unfortunately, this transformation usually produces an over segmentation of the image. To deal with this problem, proposed a strategy based on controlled marks. Therefore, before segmentation we must indicate which objects are to be segmented and which one is the background.

3. Fundamentals

Consider a gray level image $f(x,y)$ as a topographic surface S , where each gray level can be seen as terrain elevation, and where hills correspond to clear regions and where valleys or basins possess a single minimum. Suppose that each minimum $m_i(f)$ is perforated and that we introduce S vertically into a lake, assuming constant speed of descent. Water will pass through the holes flooding the surface. During this flooding, water coming from two or more minima (hence belonging to different basins) may collide. This event is to be avoided so, dams are built at the points where these waters can join. At the end of this process, only those dams shall be visible from above. These dams separate different basins $C_i(f)$, each containing a single minimum $m_i(f)$.

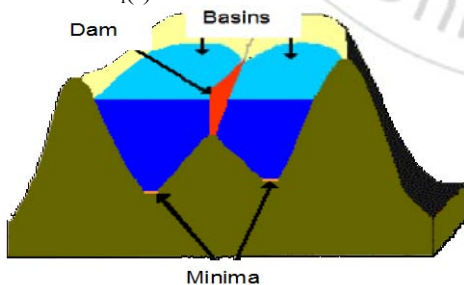


Figure 2: Flooding of the topographic surface and construction of the dam.

Figure 2, shows an example with two minima associated to two basins. The red zone represents the dam built to avoid that the water coming from adjacent basins join.

3.1 Algorithm

Let us imagine a vertical cut (cross section) through a topographical surface along the x -axis. The result will be a

1-dimensional function similar to that shown in Figure 3. In this figure, the vertical axis corresponds to gray level, the horizontal axis corresponds to position along the x -axis, the areas of lower intensity correspond to valleys. Let us also define $\max = n+1$, where n is the maximum gray level. In figure 3, $n = 3$, and $\max = 3$.

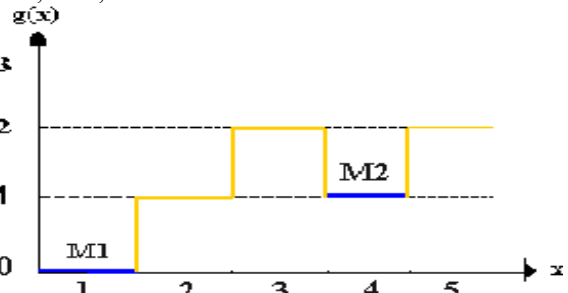


Figure 3: One-dimensional representation of a function with three gray levels (0, 1, 2).

Figure 3 is composed of three well-defined regions:

- Set of pixels belonging to regional minima (in blue)
- Set of pixels in which a drop of water shall slide towards a single minimum (in brown). For a given minimum, this set of pixels is called basin.
- Set of pixels in which a drop of water could fall into more than one basin. The points satisfying this condition form the crest line of the surface and are called watershed lines (in red in Fig. 4).

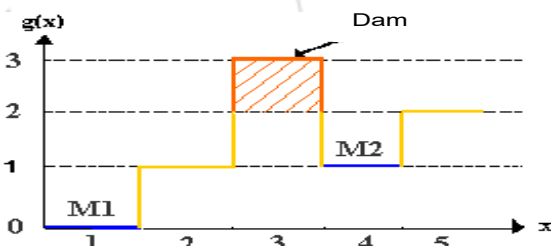


Figure 4: Building the dams

Considering again Fig. 3, we shall describe the process of dam construction. Flooding will be simulated by incrementing gray level values from 0 to \max in unit steps. This way, water will penetrate from the minima $M1$ and $M2$ and will begin flooding the image. Nothing special can be said for the first two steps (gray levels 1 and 2, respectively). However, as soon as the flooding level is reached level 3 and for the particular point $x=3$, then those waters associated with $M1$ will become in contact with waters associated with $M2$. To avoid that contact from taking place, we build a dam at this location (in red in Figure 4). The algorithm is programmed to stop when flooding reaches level $\max=3$.

4. Experimental Results of Matlab

Marker based watershed segmentation is involved. By increasing the number of marks will reduce the risk of missing boundaries, meanwhile, on the other hand, selecting less marks will reduce the risk of false edge detection. To achieve improved results, it is more convenient to deal with the morphological gradient as input image rather than the original image.

This gradient is a tool from morphological mathematics, defined as: the difference between the dilation of f minus the erosion of f , under the same structuring element or set B .

$$g = (f \oplus B) - (f \ominus B)$$



Figure 5 (a): Original image



Figure 5 (b): Morphological gradient of (a)



Figure 5 (c): Applying watershed



Figure 5(d): Topographical Image



Figure 5(e): Watershed boundaries containing minima points

The watershed segmentation process has too many basins, leading to an over segmented result (Fig. 5(c)). Each one of these basins corresponds to a minimum of the morphological gradient. These minima are produced by small variations, mainly due to noise, over the gray level values. This oversegmentation could be partially overcome employing proper filtering. However, a much-improved result can be obtained if we can impose a mark for each object as well as the background. Waters shall flood the topographical surface from the marks and shall generate as many basins as marks may exist. This way, we can solve the over segmentation problem. Figure 5(g) shows the improved results when using marks after thresholding. Especially important is the fact that the two rocks that are touching in the scene appear now as two separate objects as shown in figure 5(h) containing local minima regions in the segmented image.



Figure 5(f): distance transformation

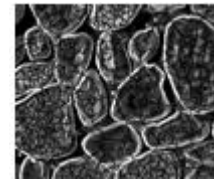


Figure 5(g): Superimposed image

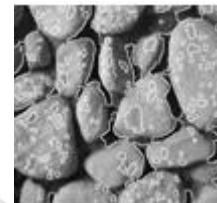


Figure 5(h): Watershed segmentation using markers containing local minima regions

Proposed Methodology for Implementation

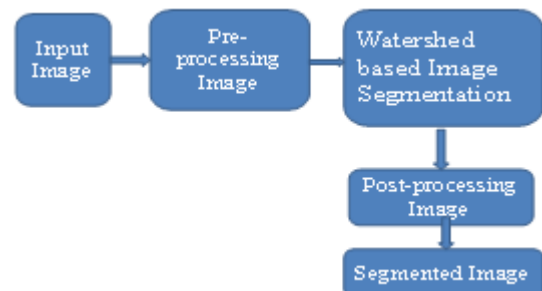


Figure 6: Proposed method

Pre-processing stage involves

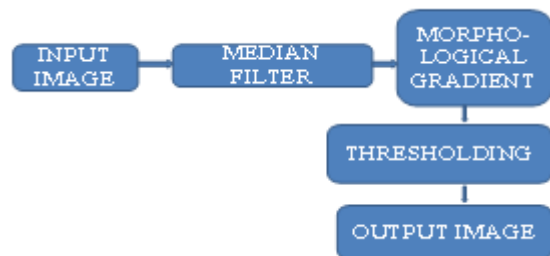


Figure 7: Pre-processing stage

Watershed Image Segmentation involves:

The approaches involved to overcome the problem of over-segmentation:

- Region growing using DIS maps
- Watershed Segmentation using Edge Markers

5.1 Region Growing Using DIS Map

This scheme consists of two major steps:

- Stage of Edge Detection
- Stage of Region Growing

5.1.1. Stage of Edge Detection

At this stage a 3X3 running window runs pixel by pixel on the input image and the DIS for the center pixel is defined. Thus at this stage, DIS maps are computed using a 3X3 window which is made to run pixel by pixel on an input image. The small DIS values indicates smooth regions on the other greater DIS values represent that pixel is on the area that changes severely in gray levels. The DIS for center pixel in is calculated as

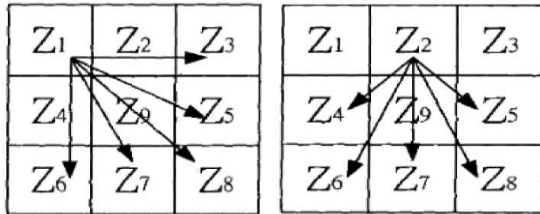


Figure 8: DIS windows

The following algorithm is used to perform edge detection:

```
BEGIN
Run 3X3 window (DIS detecting window) pixel by pixel over
an entire image
Define the DIS for the center pixel and compute the DIS map
END
```

5.1.2. Stage of Region Growing

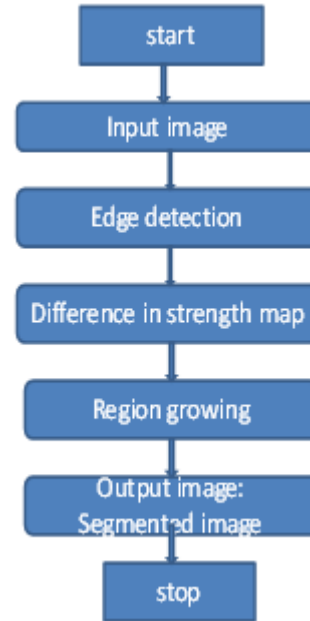
The stage of region growing involves grouping of pixels or sub regions into larger regions based on predefined criteria for growth. The syntax for the function to do basic region growing is

$$[g, NR, SI, TI] = \text{regiongrow}(f, S, T)$$

Where, f is an image to be segmented and parameter S, T is an array or a scalar. Parameter NR is the number of different regions. Parameter SI is an image containing the seed points and parameter TI is an image containing the pixels that passed the threshold test before they were processed for connectivity.

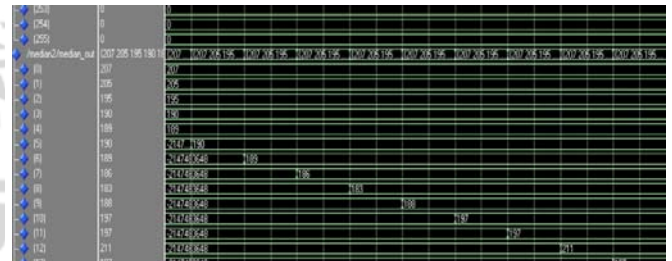
The following algorithm is used to perform region growing:-

- ```
BEGIN
```
- Choose an unprocessed pixel that has a smallest DIS value on map as R.
  - Check each of the 4 neighbouring pixel of R to see if it is unlabeled. If yes edit to the candidate list. If none has been added assign the label of a neighbouring pixel which has the gray level closest to R, then go to END.
  - Choose the pixel in the candidate list that has a smallest DIS value as a new reference pixel R.
  - Check all neighbouring pixel of R to see if they are edge pixel, label them as P BOUNDARY.
  - Check if R is an edge pixel. If so, label it as P BOUNDARY. If candidate list is not empty, go to Step 3.
  - Add 4 neighbours of R that has not been processed to the candidate list. Label R as P REGION. If the candidate list is not empty go to Step 3.

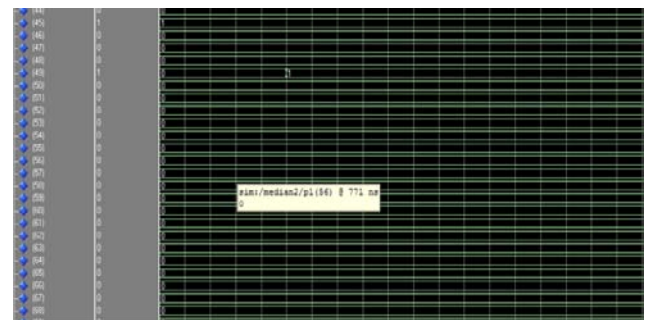


**Figure 9: Region Growing Using DIS Map**

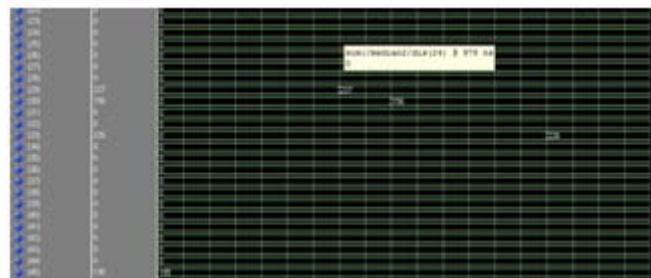
Experimental Results under Modelsim SE 6.2C in pre-processing stage involves calculating median value using median filter and finding gradient values, then calculating DIS map value of each 3x3 pixel and achieving a center pixel and following region growing techniques.



**Figure 10: Calculating mid values**



**Figure 11: Calculating gradient values**



**Figure 12: DIS value**

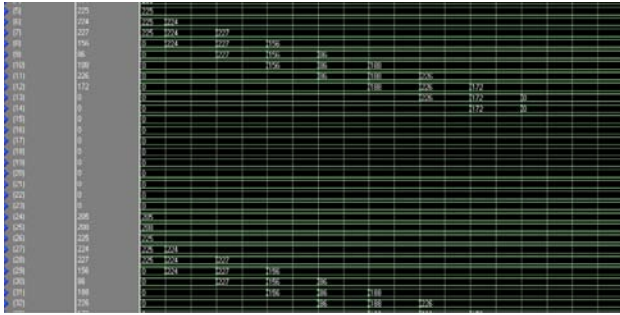


Figure 13: Region Growing

## 5.2 Watershed Segmentation Using Markers:

A marker is a connected component belonging to an image. The markers include the internal markers, associated with objects of interest, and the external markers, associated with the background. The definitions of an internal marker is

- A region that is surrounded by points of higher "altitude".
- The points in the region form a connected component.
- All the points in the connected component have the same intensity value.

The output image from the pre-processing stage is the segmented image of the watershed and markers is used to highlight the edges of the segmented image.

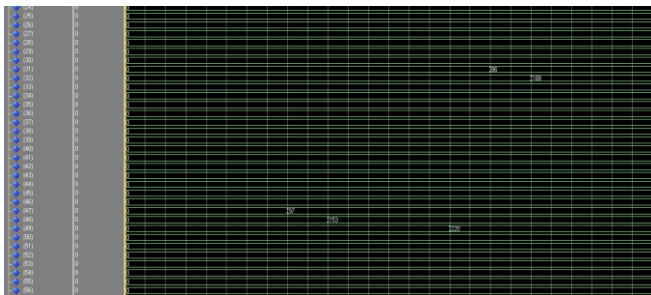


Figure 14: External markers

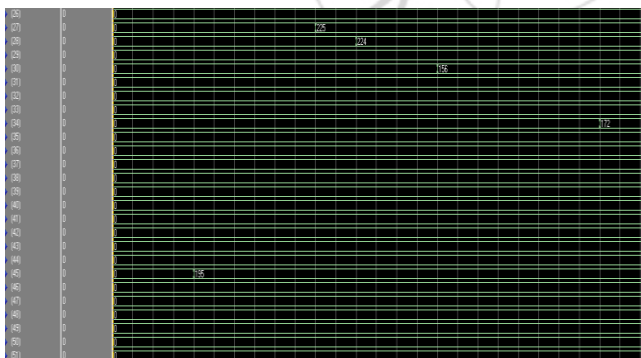


Figure 15: Internal markers

## 6. Conclusions and Future Work

It has been show that good segmentation results can be achieved and when combined with the process of marker selection, permit an effective approach for practical image segmentation.

The future works includes processing under the XILINX ISE8.2i system. Hardware implementation offers the advantage of speeding up processing. In this project, we are interested in a co-design of watershed based segmentation

system, making use of the potential software in FPGA Processor, enabled by the embedded processor power Pc.

## References

- [1] Dang Ba Khac Trieu and Tsutomu Maruyama "An implementation of a watershed algorithm based on connected components on FPGA" 978-1-4244-2796-08/25.002008IEEE Japan,2008
- [2] S.Chenl, J.Luol, Z.Shenl, X.Hul, L. Gaol "Segmentation of Multi-spectral Satellite Images Based on Watershed Algorithm" Institute of Remote Sensing Application, Chinese Academy of Sciences, Beijing, 100101, China 2008.
- [3] Dang Ba Khac, Trieu Maruyama, Tsutomu Maruyama "A pipeline Implementation of a watershed algorithm on FPGA" Systems and Information Engineering, University of Tsukuba 2007
- [4] 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
- [5] Dang Ba Khac, Trieu Maruyama, Tsutomu Maruyama " Implementation of parallel and pipelined watershed algorithm on FPGA" Systems and Information Engineering, University of Tsukuba 2006
- [6] 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
- [7] G. F. Zaki, R. A. Girgis, W. W. Moussa, and W. R. Gobran "Using HW/SW Co-design to implement an Embedded Face RecognitionIVerification System on an FPGA" 2006
- [8] C. Rambabu, 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
- [9] Chuiig J. Kuo, Soulieil F. Odeh', and Mirig C. Hi--aiig "Image segmentation with improved watershed algorithm and its FPGA implementation" 0-7803-6685-9/01/\$10.0002001 IEEE
- [10] L.Rittner, F.Flores, R. Lotufo "New Tensorial Representation of Color Images: Tensorial Morphological Gradient Applied to Color Image Segmentation" State University of Campinas - UNICAMP, Brazil, 2000
- [11] Vincent, L., and Soille, P.: "Watersheds in digital spaces: an efficient algorithm based on immersion simulations", IEEE Trans. Pattern Anal. Mach. Intell., 1991, 13, (6), pp. 583-598