FPGA Implementation of Parallel and Pipelined Approach of Watershed Image Segmentation Algorithm for Automated Video Surveillance System

Megha Sharma and Seema Verma
Electronics Department Banasthali University, Rajasthan.

ABSTRACT
The watershed algorithm is a commonly used method of solving the image segmentation problem. A new watershed based Connected Component Image Segmentation using Finite State Machine (CCISSM) algorithm is proposed. The results prove that it exhibits least computational complexity, good segmentation quality and can be implemented in the FPGA. This paper proposes a new parallel watershed based image segmentation technique and its architecture is implemented on Virtex-4 FPGA board. The results show that the proposed architecture requires minimum hardware resources, low execution time and is also suitable for use in real time applications.

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.

Image segmentation means assigning a label to each pixel in the image such that pixels with same labels share common visual characteristics. It makes an image easier to analyze in the image processing tasks. There are many different techniques available to perform image segmentation. The algorithm used in this paper 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. The watershed transformation [1] is popular image segmentation technique for gray scale images. A new watershed based Connected Component Image Segmentation using Finite State Machine (CCISSSM) algorithm is proposed.

In connected components based watershed image segmentation algorithm, image is scanned from top left to bottom right and from bottom right to top left. During each scan, unique labels are given to each detected regional minima. If the labels have already been given to their neighbor pixels during previous scan then those labels are copied to the pixels. Finally each component (pixel) is connected to its local minima and all components connected to same local minima make a segment.

Watershed based image segmentation produces over-segmentation based on different properties of the image. In this paper, pre-processing of the image before image segmentation is considered to reduce the over-segmentation problem of watershed based image segmentation.

2. Image Segmentation
Image segmentation means division of an image into meaningful structures. It is process of extracting and representing information from the image to group pixels together with region of similarity [8]. Sonka et al. define the goal of segmentation as “to divide an image into parts that have a strong correlation with objects or areas of the real world contained in the image” [6]. Figure 1 shows a basic example of the image segmentation where Figure 1a is an original gray scale image and Figure 1b is a segmented image [9].
3. Connected Component Image Segmentation using Finite State Machine

Watershed transformation also called, as watershed method is a powerful mathematical morphological tool for the image segmentation. It is more popular in the fields like biomedical and medical image processing, and computer vision [3]. In geography, watershed means the ridge that divides areas drained by different river systems. If image is viewed as geological landscape, the watershed lines determines boundaries which separates image regions. The watershed transform computes catchment basins and ridgelines (also known as watershed lines), where catchment basins corresponding to image regions and ridgelines relating to region boundaries [4]. Segmentation by watershed embodies many of the concepts of the three techniques such as threshold based, edge based and region based segmentation.

Watershed algorithms based on watershed transformation have mainly two classes. The first class contains the flooding based watershed algorithms and it is a traditional approach where as the second class contains rainfalling based watershed algorithms.

Many algorithms have been proposed in both classes but the proposed connected components based watershed algorithm shows very good performance compared to all others. It comes under the rainfalling based watershed algorithm approach. It gives very good segmentation results, and meets the criteria of less computational complexity for hardware implementation.

There are mainly three stages as indicated by Figure 2 for watershed based image segmentation approach. First stage is defined as pre-processing, second stage as watershed based image segmentation and last stage as post-processing. Input image is first processed by the pre-processing stage, and then given to watershed based segmentation stage. The resulting image is post processed by the final stage to get a segmented image. Pre-processing and post-processing are necessary to overcome the problem of over-segmentation in watershed based image segmentation.

4. Parallel Architecture Of Watershed Image Segmentation

Parallel architecture proposal is given in this section. It uses wider data bus interface with the external memory and higher hardware resources compared to the pipeline implementation. But, it reduces total time required to perform segmentation. Block diagram is shown in Figure 3.

**Algorithm**

**Step-A:** For each pixel in a given image, its lowest neighborhood is detected, and the pixel points the lowest (using its address in \( BM \) and chain code in \( SYR \)). If the pixel is on a plateau, no lower neighbors are found.

**Step-B:** For each pixel on plateaus, if one of its neighbors points somewhere (this means that this neighbor is on the edge between lower plateau, and points the lower plateau), the pixel points the neighbor. By repeating this procedure using a FIFO, pixels on plateaus point one of their lower plateaus. The FIFO is used to divide plateaus evenly according to the distances from their lower plateaus.
Step-C: For each pixel which points nowhere (this means that the pixel is on local minima plateaus which are not labeled yet), a unique label is given. Then, its neighbor pixels on the same plateau point the pixel. By repeating this procedure using recursive calls (BM) or a stack (SYR), all pixels on the plateau points the pixel (namely the label) directly or indirectly.

Step-D: Pointers for all pixels are dereferenced, and labels are given to all pixels.

Table 1. Total segmentation time for 512 x 512 image with 10 scans and device utilization for Virtex-4 FPGA device.

<table>
<thead>
<tr>
<th>Segmentation Units</th>
<th>Total Segmentation Time (ms)</th>
<th>Device Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.10</td>
<td>7%</td>
</tr>
<tr>
<td>5</td>
<td>80.2</td>
<td>33%</td>
</tr>
<tr>
<td>10</td>
<td>401.1</td>
<td>75%</td>
</tr>
</tbody>
</table>

Table shows total segmentation time for 512 x 512 image with single, five or ten segmentation units. It is considered that each image needs 10 scans to compute the segmentation. Device utilization is given for Virtex-4 FPGA device. As shown by total segmentation time in Table 1, it is possible to perform real time image segmentation using the proposed parallel architecture.

Conclusion

By using the FPGA device which has larger internal memory, the use of external memory can be avoided. It is possible to have multiple segmentation units to achieve high performance because single pipelined segmentation unit requires very less hardware resources. The starting point of the thesis is to build image segmentation architecture for an FPGA implementation that provides good segmentation quality, requires few hardware resources and short execution time, and is suitable for use in real time applications.

References