Watershed Segmentation using MATLAB

Akshat Bajpai[#1]

[#]*School of Engineering and Computer Science , University of the Pacific*

3601 Pacific Avenue, Stockton, CA

[1]a_bajpai@u.pacific.edu

***Abstract*— This document contains MATLAB implementation of an image processing algorithm known as Watershed Segmentation. The algorithm is used to, as its name suggests, segment different objects in an image. The whole algorithm comprises 2 sections. The first section deals with normal watershed segmentation and the second section deals with watershed segmentation using Markers. There's also an additional section comprising of program that uses label image, identifies the ridge lines separating catchment basins and highlights the ridge lines in the main input image in green.**

***Keywords*— Computer Vision, Image Processing, Watershed Segmentation, MATLAB, Image Segmentation**

I. Implementation

This program is an MATLAB implementation of the Watershed Segmentation algorithm. The algorithm first adds the neighbors to a priority queue(sorted in value), chooses the local minima and performs a two step action from all the pixels from the queue. The first step is to check if all labeled neighbors have the same label. If they do, it's assigned to the pixel. The second step is to add all the non-marked neighbors. The marker based watershed algorithm isn't that different from the normal one. The notable differences are that the marker based watershed algorithm allows new catchment basins only where markers are placed. These markers are application specific. While growing new catchment basins, the algorithm checks if the image(q) is <= g. Finally, the catchment basins can be computed even before entering the outer for loop.

II. Methodology

We utilize most of the functions from the earlier labs such as double-thresholding, dilation, floodfill_seperate, convolution, canny edge detection, etc.

The program begins by taking in an input image("cell.pgm", "holes.pgm"). As an example, we'll select the "holes.pgm"(fig. 1) image. The program then calls the watershed function, which returns a labeled image denoting segmented portions of the image (fig.2). Fig.3. shows the quantized Magnitude image. The program computes the edge image of the normal watershed image from c to obtain lines between objects(Fig.4). The image then is processed using double thresholding(Fig. 5) and furthered to Fig.6, which depicts a chamfer image in a separate window titled 'Marker Watershed: Chamfer'. Then, the program performs Marker-Based watershed segmentation using Marker (Fig.7). Finally, the algorithm computes Ridge Lines and displays it on the final image with Green Color.

Additionally, the program also highlights the exact ridgelines between catchment basins, shown in Fig. 9.

The same process goes for “cell.pgm” image as well.

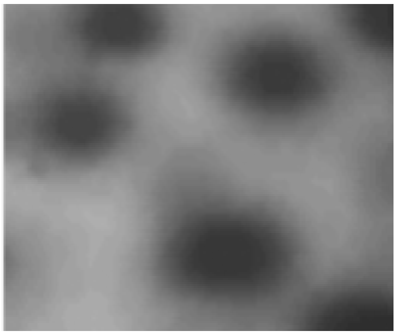
Fig.1 Original Image

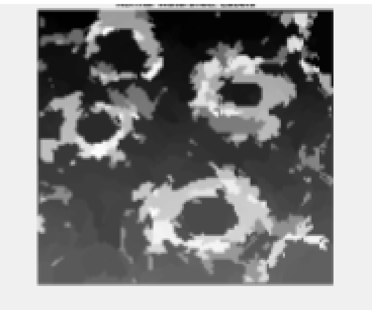
Fig. 2. Watershed Labels

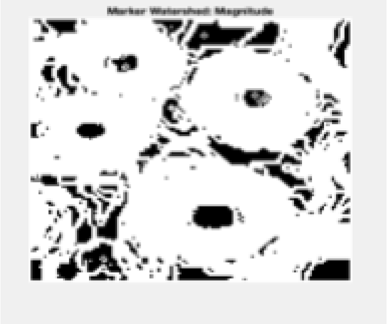

Fig. 3 Watershed: Magnitude

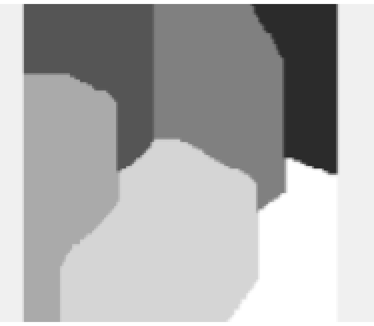
Fig. 4 Watershed: Edges seperating objects

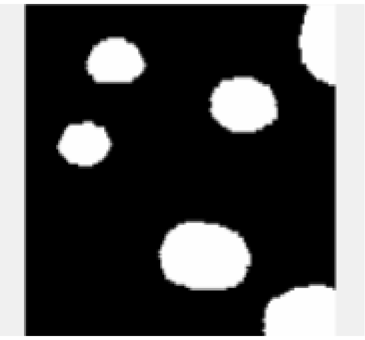
Fig. 5 Marker Watershed: Threshold

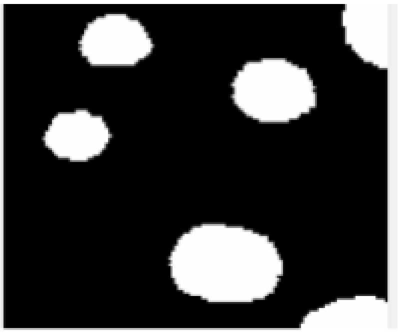
Fig. 6 Watershed: Chamfer

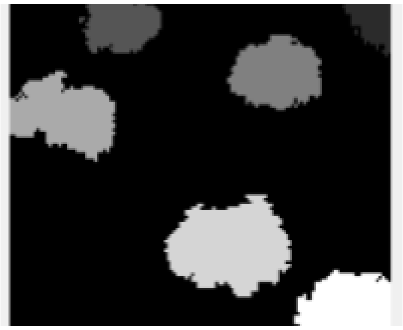
Fig. 7 Watershed: Label Image

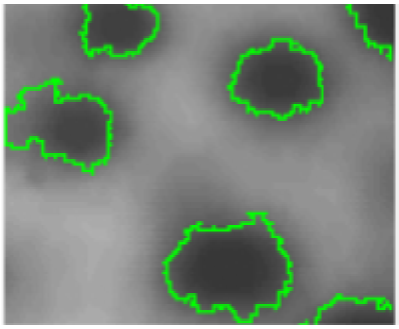
Fig. 8 Watershed: Ridge Lines

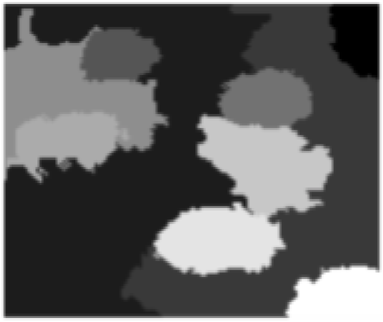
Fig. 9 Highlighting the exact ridgelines between catchment basins

IV. Conclusions

The program successfully implements the Watershed algorithm using MATLAB. It also provides intermediate outputs walking us through the entire process visually. The input image is a blurred image of holes in “.pgm” format. The output is the green ridged-lines augmented over the original image. Additionally, we are also highlights able to highlight the exact ridgelines between the catchment basins.