Program 4 Report

Ellis Chen

ECPE 124

Abstract— **This document provides information about the procedures and results of various digital image processing techniques used to detect and segment objects and regions within a given grayscale image.**

I. Normal Watershed

Section 1 of this report covers the image processing procedures for the normal watershed implementation. The watershed algorithm separates regions within an image based on grayscale pixel intensities.

1) The approach to normal watershed was to first compute the magnitude of the input image by convolving the image with a gaussian kernel and gaussian derivative with sigma = 0.6
2) The magnitude of the image was quantized into integer values. This makes 256 possible regions for the watershed, but realistically, the watershed implementation will less than 256 regions.
3) All the pixel coordinates of the input image were preallocated in a list where each index of the list corresponds to the quantized pixel intensity.
4) All labels are set to -1, which means pixels are unassigned to a region. Global label is set to 0, which keeps track of the number of segmented regions for normal watershed.
5) For each level of pixel intensity, the algorithm will do the following until each pixel intensity is accounted for:
    a) The label of the pixel will be updated if any of the 8 neighbouring pixels have a label that is not -1. If a neighbouring pixel has a unique label, the current pixel will have the same label as the neighbouring pixel. This essentially expands the catchment basin one pixel at a time. The current pixel is pushed to the frontier.
    b) The frontier pops a pixel coordinate, in which the program will search for neighbouring pixels that are unlabelled and have the same pixel intensity. If a neighbouring pixel meets these criteria, the current pixel will assign the neighbouring pixel its label and push the neighbouring pixel onto the frontier. This essentially expands the catchment basin even further. This process will repeat until there are no more neighbouring pixels with the same pixel intensity and that are unlabelled, which means the frontier will be empty.
    c) This step will create a new catchment basin and give it a new label by flood fill.

II. Normal Watershed Results

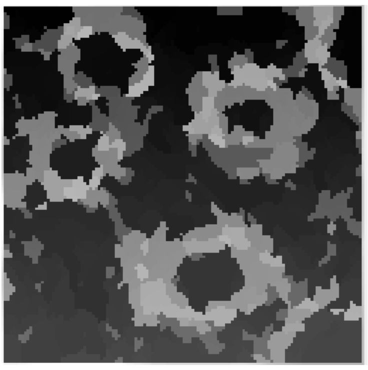
Figure 1: Final Output for Normal Watershed using holes.pgm

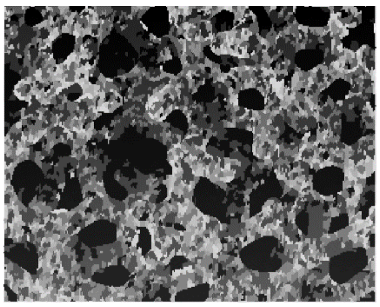
Figure 2: Final Output for Normal Watershed using cells_small.pgm

The final results for both images seem to have numerous segmented regions. For a normal watershed implementation, these results are to be expected since there were no techniques that were used to limit the number of segments that the program would output. Otherwise, both images managed to retain minor details reminiscent of the original images.

III. Watershed Using Markers

Section 2 of this report covers the image processing procedures for the marker watershed implementation. The watershed algorithm separates regions, and the number and location of the regions is dependent on the markers. This algorithm extends on the knowledge of various image processing techniques not used in section 1.

1) Similar to normal watershed, the image needed to be convolved with gaussian kernels and derivatives.
2) The input images were quantized as well.

3) The steps in implementing marker watershed started to deviate from this point on. A thresholded image was produced from the input image. The high contrast between dark and bright pixel intensities make it possible for the implementation of the next step.
4) The chamfer distances between all the thresholded objects were computed. The pixels that were closer to the thresholded objects were darker in intensity while the pixels furthest away were the brightest.
5) The simplified image from the chamfer image was segmented into regions using normal watershed.
6) The canny edge image was computed from the watershed image to create clean edges.
7) The thresholded image and the clean edge image are ORed together, in which any bright pixels from either image will override the darker pixels. This results in an image suitable for marker-based watershed.
8) Finally, in the marker-based watershed implementation, the program first flood filled all the regions that were marked by the marker image, this means that the number of regions is limited by the number of markers, which makes the number of regions independent of the pixel intensities.

IV. Watershed Using Markers Results

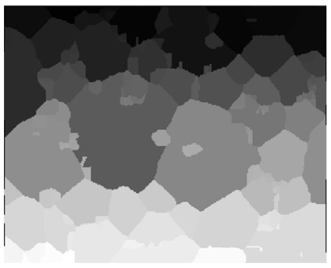
Figure 3: Normal Watershed on Chamfer using cells_small.pgm

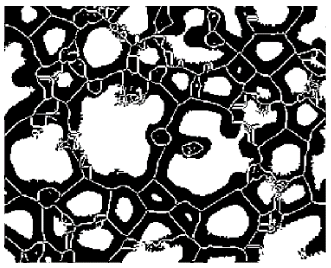
Figure 4: Threshold image ORed with Edge image using cell_small.pgm

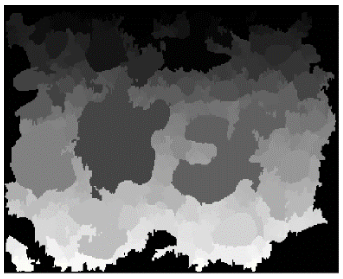
Figure 5: Watershed using Markers using cells_small.pgm

The final image that resulted from watershed using markers for cells_small.pgm could have been implemented more efficiently. There still exists too many segments in this output image. The cause may have either been poor implementation in the code such as faulty label overriding, or this could have been a result of a poor choice in the values for the high and low threshold values used in producing the double thresholded image.

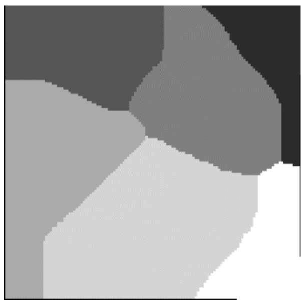
Figure 6: Normal Watershed on Chamfer using holes.pgm

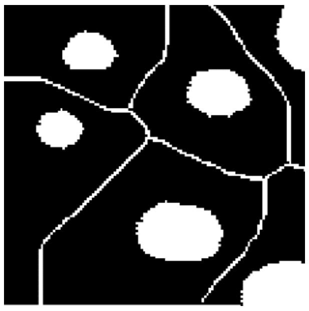
Figure 7: Threshold image ORed with Edge image using holes.pgm

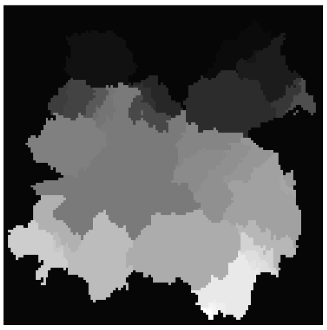
Figure 8: Watershed using Markers using holes.pgm

The final image of holes.pgm is slightly better than the final image of cells_small.pgm. Still, the extra number of segmented regions still persist. Looking at figures 6 and 7, there seems to be no apparent mistakes in those implementations, which suggests that the implementation for watershed using markers is not as efficient as initially suspected. Perhaps the problem is that the flood fill algorithm that creates these new segmented regions happened to occur more than the number of times expected, or there exists a problem in assigning the correct labels to certain pixels.