Assignment 3

**Prepared by**Brendan Burke

**Computer Vision  
4410.01**

**3/26/2018**

**Table of Contents**

[**List of Figures**](#_sb130bvo8meo)**………………………………………………………………………………………………….. 2**

[**List of Acronyms**](#_tnlodiaf95ve)**………………………………………………………………………………………………. 3**

[**Assignment 3**](#_u64gzylym7q0)**………………………………………………………………………………………………….. 5**

[Assignment Details](#_1fob9te)………………………………………………………………………………………………………………………………... 5

[Implementation](#_x1pj9godap69)…………………………………………………………………………………………………………………………………….. 5

[Resources](#_agcw5tdwdido)………………………………………………………………………………………………………………………………………………. 5

[Figures](#_2et92p0)…………………………………………………………………………………………………………………………………………………... 6

[**Results**](#_drcw7i5u8z40)**…………………………………………………………………………………………………………. 6**

[Mean Shift](#_lvr99fl7uwke)…………………………………………………………………………………………………………………………………………….. 6

[Watershed](#_remqjcupk9mz)…………………………………………………………………………………………………………………………………………….. 6

**References**

# **List of Figures**

**Figure 1** Mean Shift output images

**Figure 2** Watershed algorithm with modified variable values

**Figure 3** Implemented Watershed Algorithm Tutorial Input Image

# **List of Acronyms**

RGB………………………………………………………………………………....Red, Green, Blue

LAB…………….Luminosity, Chromatic layer (red-green axis), Chromatic layer (blue-yellow axis)

HLS……………………………………………………………………………..High Level Synthesis

# 

# Assignment 3

## Assignment Details

In this assignment, I analyzed how changing the variable values in an image segmentation algorithm affected the output image. I compared the results from using the Watershed algorithm with the default variable values of the Mean Shift algorithm. I changed the binary threshold types and the distance transform type used in the Watershed algorithm.

## Implementation

The Mean Shift algorithm uses an xml file that allows users to quickly change the segmentation method used by the algorithm. The three segmentation methods used are RGB, LAB, and HLS. I ran each of the segmentation methods and saved the output images to use for comparison with the Watershed algorithm. Figure 1 shows the three output images. Of the three methods, the RGB method performed the best for image segmentation. Both LAB and HLS returned splotches of white throughout their outputed images.

For the Watershed method, I used two RGB colored images and two grayscale images for testing. I changed the threshold type used to convert each image into binary images between CV\_THRESH\_BINARY, CV\_THRESH\_OTSU, and using both types. For Colored Image 1 and Black and White 1, I also extended their analysis by changing the distance transform type between CV\_DIST\_L2 and DIST\_LABEL\_PIXEL for each change of binary image threshold type. For Colored Image 2 and Black and White 2 I left the distance transform type as CV\_DIST\_L2. Figure 2 shows the output images.

## Resources

Aside from the OpenCV documentation, the only resource I used was the tutorial found from [1]. This tutorial explained how to apply the Watershed image segmentation algorithm to an image of a pile of playing cards. In the tutorial, the second step involved changing the white background to black in order to better distinguish between the foreground and the background. Since the background in the tutorial’s example image is completely white (255), this method did not work for the images I used in my analysis. Therefore I removed this step so it would not negatively influence my results. The next step in the tutorial is applying a Laplacian sharpening filter. I decided not to perform this step to keep the implementation simplistic. I performed the remaining steps as listed in the tutorial. These steps include converting the image to grayscale, converting the image into a binary image, using a distance transform, performing a morphology operation, applying seeds (markers), and finally applying the Watershed algorithm.

## Figures

In order to preserve the conciseness of the report, I have included a link to the figures of the output images. Please follow this link: <https://docs.google.com/document/d/1ZicZjRqvtds2QdkAMjlHCd8WJoKHhqAkO2brCoG1zr4/edit?usp=sharing>

# Results

## Mean Shift

While analyzing the Mean Shift algorithm, I found that the RGB segmentation method returned the best results. It is evident that the quality of segmentation from the output images decreases when the LAB and HLS methods are used. The RGB method does not return the white markings that can be found in the output images from the LAB and HLS methods.

## Watershed

When the binary threshold variable was set to CV\_THRESH\_BINARY, the output images from the Watershed algorithm returned the most detail. Changing the threshold to CV\_THRESH\_OTSU returned completely purple images without any details. Furthermore, changing the distance transform type, or changing the input images from color to black and white did not significantly affect the output images either.

The results of changing the distance transform type in Colored Image 1 are all nearly identical aside from a difference in hue. The greatest change is the difference between the binary threshold outputs. Only the CV\_THRESH\_BINARY output image returned any significant results, since both CV\_THRESH\_OTSU and using both types of distance transform types returned completely purple images. The same results were returned from Colored Image 2, except even less detail was returned from using the CV\_THRESH\_BINARY threshold. A notable difference between Colored Image 1 and 2 is that Colored Image 2’s results, there is a small, dark circle in the top left corner. I believe this is a marker that was not transformed during one of the steps of the algorithm.

As expected, the output images from Black and White 1 are the same as Colored Image 1. These results were expected because the implemented version of the Watershed algorithm first converts the image into grayscale regardless if the image is originally RGB or grayscale. The reason an analysis was performed with an image that was already grayscale was to eliminate the chance of erroneous image input. Black and White 1 was already grayscale when used as the input image, but I wanted to eliminate the possibility of the image being altered before it was used as input for the algorithm. Also, I wanted the implemented algorithm to remain as constant as possible, except for the test variables.

Interestingly, the output images from Black and White 2 do not follow the pattern that the other output images show. Typically, the first output image, which uses the CV\_THRESH\_BINARY threshold, returns the most detail, while the second and third images return very few details. The first output image of Black and White 2 actually returns the least amount of visible detail, while the second and third output images return the most details. Black and White 2 using the CV\_THRESH\_BINARY threshold returns a mostly purple image, while using the CV\_THRESH\_OTSU threshold and using both thresholds returns an image with a segmentation between the foreground and background. The segmentation returned is of part of the sky which is very dark, segmented from the rest of the image. A possible explanation for why Black and White 2 does not follow the pattern found in the other images is because of the input image used here. The input image for Black and White 2 consists of a mostly light grey image, with dark areas in the sky. The implemented version of the Watershed algorithm might consider the majority of the image to be so similar as to classify it as such, with the dark ares in the sky as separate areas in the image.

The difference of segmentation quality between the output images from the Watershed and Mean Shift algorithms is obvious: the Mean Shift algorithm by far outperforms the implemented Watershed algorithm. Although the quality of the Mean Shift output deteriorates when using the LAB or HLS segmentation methods, it still performs better than the implemented Watershed algorithm. It was expected that the Mean Shift algorithm would perform better, since it has a much more in depth implementation than this Watershed algorithm. Furthermore, the implemented Watershed algorithm was designed for very simple image segmentation. Figure 3 shows the input image used from the Watershed tutorial. It is a simple image of a few playing cards laying on top of each other with a white background. Because this image is very simple, the program is able to easily distinguish between the objects and return very accurate image segmentation. However, when this implementation of the Watershed algorithm is applied to the complex, real-world images used for this report, it does not return quality segmentation.

For future work, I would alter the Watershed algorithm to be able to segment more complex images. As this implementation was designed for simple images, it did not perform well. Reviewing the steps and implementation of the Mean Shift algorithm could possibly help with improving the Watershed algorithm. Furthermore, it would be beneficial to use more input images to be able to see how the output images are affected when using the Watershed algorithm. I would also use a more diverse set of images, ranging from low to high complexity. This would also provide insight into how the images are affected. **References**

[1] (n.d.). Retrieved March 24, 2018, from https://docs.opencv.org/3.1.0/d2/dbd/tutorial\_distance\_transform.html