Assignment 3

**Prepared by**Brendan Burke

**Computer Vision  
4410.01**

**4/16/2018**

**Table of Contents**

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

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

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

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

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

[Figures](#_2et92p0)…………………………………………………………………………………………………………………………………………………... 5

[**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

# **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 the effect of changing certain variable values in an image segmentation algorithm. Specifically, I compared the results the Watershed algorithm with the Mean Shift algorithm. The Mean Shift algorithm provided to me allowed for three different segmentation methods to be used: RGB, LAB, and HLS. For the Watershed algorithm, I compared changes to the Binary Threshold, the Otsu’s Threshold, and the Normalize Threshold.

## 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 the same input image used by the Mean Shift algorithm. Four tests were performed. Two of these tests had four sub-tests each, making a total of ten tests performed. In the first test, *ColoredImage1*, I used the Otsu’s Threshold method to perform image segmentation. In the second test, *ColoredImage2*, I changed Otsu’s Threshold to Binary Threshold. Test three and four were each made up of four subtests, *ColoredImage3.1, ColoredImage3.2, ColoredImage3.3…*This is because in each subtest two threshold values are modified: the threshold type (Otsu or Binary), and the Normalize Threshold. For example, in *ColoredImage3.1* a low threshold value of 97, and high threshold value of 155 was used. The second subtest used a low Otsu/ Binary threshold, and a high Normalize threshold value. The third subtest used a high Otsu/ Binary threshold, and a low Normalize threshold value. The fourth subtest used a high Otsu/ Binary threshold, and a high Normalize threshold value. I chose these values because they represent the values below and above the median threshold value of 127.5.

## 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/1OjWm42gyRZsuEFwrUS11BOn0zqgBT6crL351895SiGs/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

*ColoredImage1* returned a very segmented image. It was able to adequately distinguish the background from the foreground. The segmentation of the foreground is very detailed. Much of the gravel pathway is segmented, parts of the trees on the right hand side are segmented, and parts of the shrubbery on the left hand side are also heavily segmented. Yet, although parts of the shrubbery are accurately segmented, this test segmented the background/ sky with a large area of the shrubbery. Here, the program believes the sky and most of the shrubbery are indistinguishable.

*ColoredImage2* returned an image that is less segmented than *ColoredImage1*. The first obvious difference is that the sky and the gravel pathway are segmented together. It seems that even parts of the grass in the image are segmented with the sky and gravel pathway as well. The second obvious difference in test two is that the majority of the trees are segmented together. There is an area at the tops of the trees that is segmented separately, but most of the trees are segmented together. The last obvious difference is the segmentation of the shrubbery. Due to the way the sky, the gravel pathway, and the grass are segmented, the shrubbery appears to be segmented separately, but in a clump. What I mean is that there are parts of the shrubbery that is segmented, but as a whole these separate segments are in close proximity to one another that they form the outline of the shrubbery.

*ColoredImage1* and *ColoredImage2* returned different results. *ColoredImage1* segmented the image more heavily than *ColoredImage2*. Some obvious examples are the heavily segmented gravel pathway and trees in *ColoredImage1*. *ColoredImage2* on the other hand mostly segmented the trees together, and had larger segments of the shrubbery than *ColoredImage1*. Both tests, however, had difficulty segmenting the sky from the foreground. *ColoredImage1* segmented the sky and parts of the shrubbery together, while *ColoredImage2* segmented the sky and the gravel pathway together. Deciding which test was more accurate at image segmentation, I argue that *ColoredImage1* was more accurate because it did not segment the gravel pathway with the sky, and more segments were returned.

*ColoredImage3.1* used a low Binary and Normalize threshold value. Similar to *ColoredImage1*, *ColoredImage3.1* also segmented the sky with part of the shrubbery. However, unlike the *ColoredImage1*, it segmented all of the trees on the right hand side together. Furthermore, its segmentation of the shrubbery is interesting. There are several small segments of the upper area of the shrubbery, that is separated from the main body. It is like these segments are floating above the shrubbery. At the same time, the lower area of the shrubber and the gravel pathway are heavily segmented. There is, however, a large segment in the middle of the pathway, with numerous, smaller segments above and below it. *ColoredImage3.2* also segmented the trees on the right hand side together as well. One noticeable difference from *ColoredImage3.1* is that *ColoredImage3.2* did not segment large areas of the shrubbery. Actually, there are only small segments of the shrubbery, while the majority of the shrubbery is segmented with the sky. Interestingly, there is a larger segment of the gravel pathway. It is interesting because *ColoredImage3.2* has segmented more of the gravel pathway and has segmented more of the sky and shrubbery together, while *ColoredImage3.1* segmented less of both areas. *ColoredImage3.3* returned with the largest segments of all the subtests. It was able to completely segment the background from the foreground. However, because of this, all of the area that is not a part of the sky has been segmented together, aside from several very small segments throughout the image. Similarly, *ColoredImage3.4* returned large segments as well. Interestingly, these segments are practically split down the middle of the image. It appears the trees and gravel pathway have been segmented together, while the sky and the entire shrubbery together. There is, however, some segmentation of the gravel pathway at the bottom middle portion of the image.

The results from *ColoredImage4* were very interesting. *ColoredImage4.1* and *ColoredImage4.3* returned the exact same results. The image is segmented into three large sections: the sky, the trees and gravel pathway, and the shrubbery and gravel pathway. Even more interesting is that *ColoredImage4.2* and *ColoredImage4.4* also returned the exact same results. Their image segmentation covers two large areas: the sky, shrubbery, and the trees and parts of the gravel pathway. After confirming these subtests were performed correctly, I investigated what could have possibly caused this outcome. I deduced that it is because of the Normalize Threshold. Please note, that when the Normalize Threshold is set to a lower value that *ColoredImage4.1* and *ColoredImage4.3* results are found. When a higher Normalize Threshold is set, *ColoredImage4.2* and *ColoredImage4.4* results are returned. Furthermore, it does not appear that changing the Otsu threshold value has an effect on the results. Note that *ColoredImage4.1* uses a low Otsu threshold value, and *ColoredImage4.3* uses a high threshold value. Yet, there is no change in either image. Similarly, *ColoredImage4.2* uses a low Otsu threshold value, and *ColoredImage4.4* uses a high value, and yet, the images are the same.

*ColoredImage4* and *ColoredImage4* had some similarities, but are also very different. Both tests have large areas of segmentation, but there are more defined segmented areas in *ColoredImage3*. For example, *ColoredImage3.3* and *ColoredImage3.4* have two large segments. This is similar to *ColoredImage4.2* and *ColoredImage4.4*. Yet, *ColoredImage3.1* and *ColoredImage3.2* by far are much more segmented than any of the subtests in *ColoredImage4*. Because of this difference, *ColoredImage3* was more accurate in segmenting the image than *ColoredImage4*. Now determining which of the subtests in *ColoredImage3* was the most accurate is difficult. Because the first two subtests are more segmented than the later two, it can be argued that they are more accurate. However, because *ColoredImage3.3* clearly distinguished the sky from the foreground, it can be argued it performed perfect segmentation, at a certain level. One can argue that *ColoredImage3.4* performed poor segmentation because it segmented the shrubbery with the sky, and thus did not segment as well as *ColoredImage3.3*. I argue that a combination of *ColoredImage3.3* and *ColoredImage3.1* could be used to obtain near perfect segmentation. Clearly, *ColoredImage3.3* was able to distinguish the background from the foreground better than *ColoredImage3.1*. Yet, *ColoredImage3.1* was able to segment more of the image than *ColoredImage3.3*. So by combining the strengths of each image segmentation subtest, one could obtain a near perfect segmented image. **References**

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