Assignment #3

Prepared by  
Michael Pérez

Computer Vision  
CAP 4410

29 October 2019

Table of Contents

[List of Figures 2](#_Toc23282759)

[1. Introduction 3](#_Toc23282760)

[1.1 Introduction 3](#_Toc23282761)

[1.2 Running the Project 3](#_Toc23282762)

[1.3 Product Scope 3](#_Toc23282763)

[2. Otsu Method with Two Classes 4](#_Toc23282764)

[2.1 Implementation 4](#_Toc23282765)

[2.2 Testing 5](#_Toc23282766)

[2.3 Discussion 5](#_Toc23282767)

[3. Otsu Method with Four Classes 6](#_Toc23282768)

[3.1 Implementation 6](#_Toc23282769)

[3.2 Testing 7](#_Toc23282770)

[3.3 Discussion 7](#_Toc23282771)

[4. Mean Shift Method 7](#_Toc23282772)

[4.1 Implementation 7](#_Toc23282773)

[4.2 Testing 9](#_Toc23282774)

[4.3 Discussion 10](#_Toc23282775)

# List of Figures

**Figure 1** Otsu Method with Two Classes Implementation……..…………………..……..................... 5

**Figure 2** Otsu Method Background…………………………………..……..............................................6

**Figure 3** Otsu Method with Two Classes Test Images…........................................................................6

**Figure 4** Otsu Method with Two Classes Output……..………………….............................................. 6

**Figure 5** Otsu Method with Four Classes Implementation ……….………..….…………..….............. 7

**Figure 6** Otsu Method with Four Classes Test Images.…..…………………………….…................... 8

**Figure 7** Otsu Method with Four Classes Output ………….…..............................................................8

**Figure 8** Mean Shift Method Implementation………….…....................................................................9

**Figure 9** Mean Shift Segmentation Method………………………..………….......................................9

**Figure 10** Mean Shift Visualization……………………………..……..…….......................................10

**Figure 11** Mean Shift Method Test Images………………………..…………......................................10

**Figure 12** Mean Shift Method Output……………………………………….........................................11

# Introduction

## Introduction

My name is Michael Pérez, and I am a senior studying Computer Science at Florida Polytechnic University. This objective of this project was to implement different methods of image segmentation. I implemented these methods using OpenCV in Visual Studio, in C++. I tested each method on its respective test images in the “Input Folder” on Canvas.

## Running the Project

There are two windows that appears immediately when the program is run:

1. Program Console
2. Image (image file name specified in main.cpp)

When the Otsu segmentation method is applied to an image, the edited image is displayed in a new window:

1. Edited Image

When the mean shift segmentation method is applied to an image, the filtered image then the edited image is displayed in a new window:

1. Edited Image
2. Filtered Image

While the program is running, pressing the following buttons corresponds to these commands:

‘1’: Apply the Otsu method for two classes and display the edited image.

‘2’: Apply the Otsu method for multiple classes and display the edited image.

‘3’: Apply the mean shift segmentation method and display the filtered image, and edited image.

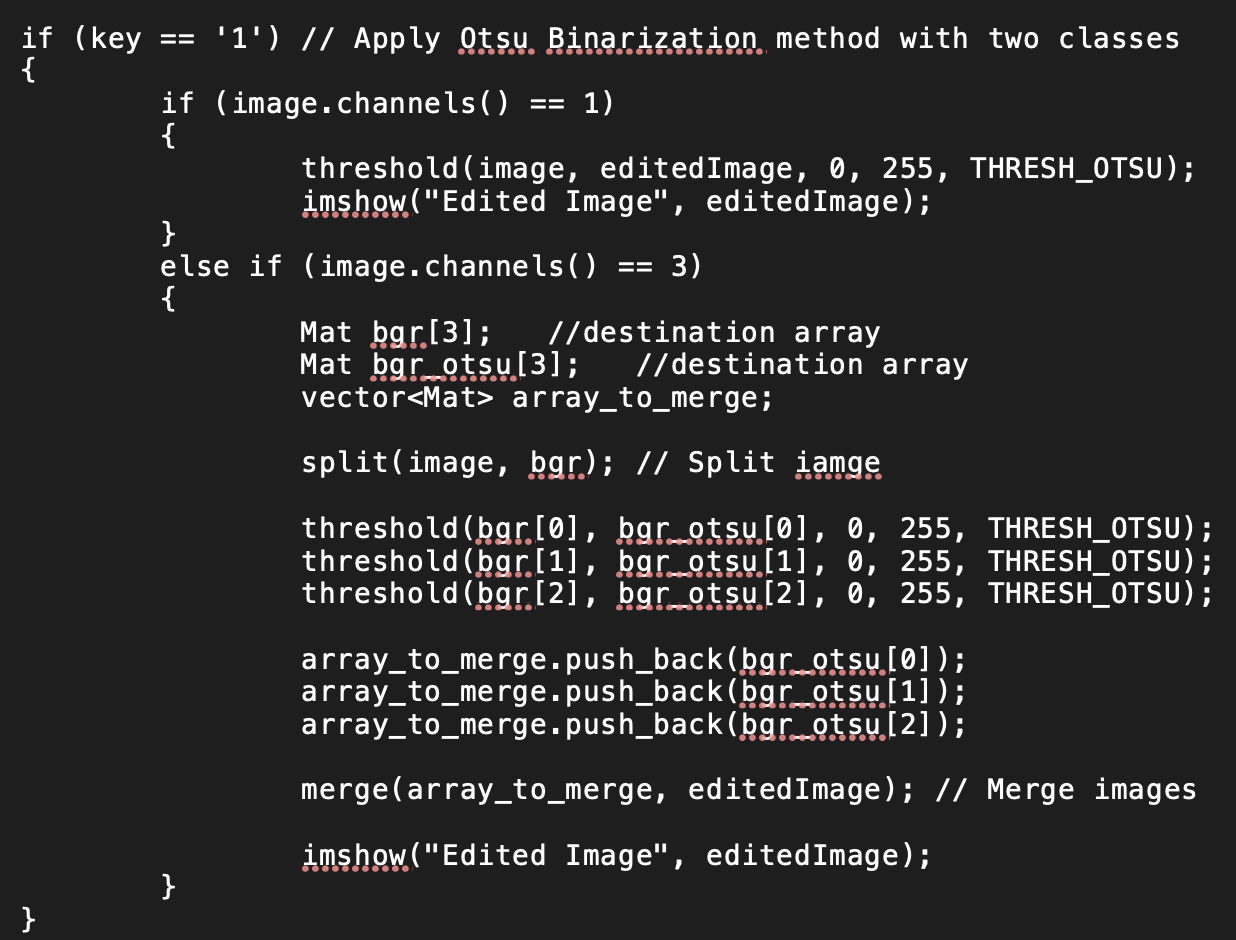
## Product Scope

This product has the ability of applying three types of segmentation methods to an image. Image segmentation has numerous applications, from medical imaging to object detection and other recognition tasks. These segmentation methods are broad in the sense that they can be suited to fit any needs. After more specialization, these methods can be applied to a wide variety of applications.

# Otsu Method with Two Classes

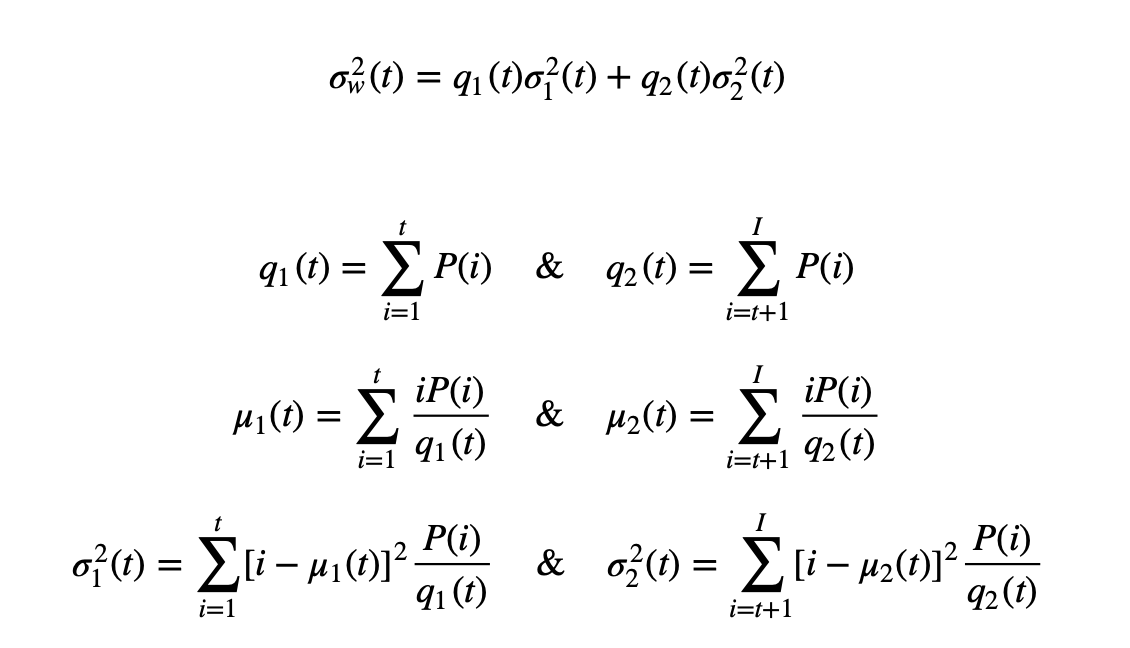
## Implementation

I read about the Otsu binarization method in the lecture slides and on the internet and created an algorithm that first checks if the image is color or grayscale. If the image is grayscale, I perform Otsu binarization on the one channel of pixel values using OpenCV’s built-in threshold() function. If the image is color, I perform the same operation on each of the red, green, and blue channels by using the same method [1].



**Figure 1: Otsu Method with Two Classes Implementation**

Thresholding an image is a method in which all pixel values less than a certain threshold are set to 0 and all pixel values greater than that threshold are set to the maximum pixel value. Otsu Binarization is a method of thresholding in which the threshold level is calculated automatically as opposed to it being set by the user. Otsu’s algorithm finds the best threshold value such that the weighted with-in class variance in minimized [2].

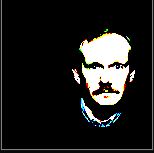


**Figure 2: Otsu Method Background** [2]

## Testing



**Figure 3: Otsu Method with Two Classes Test Images**



**Figure 4: Otsu Method with Two Classes Output**

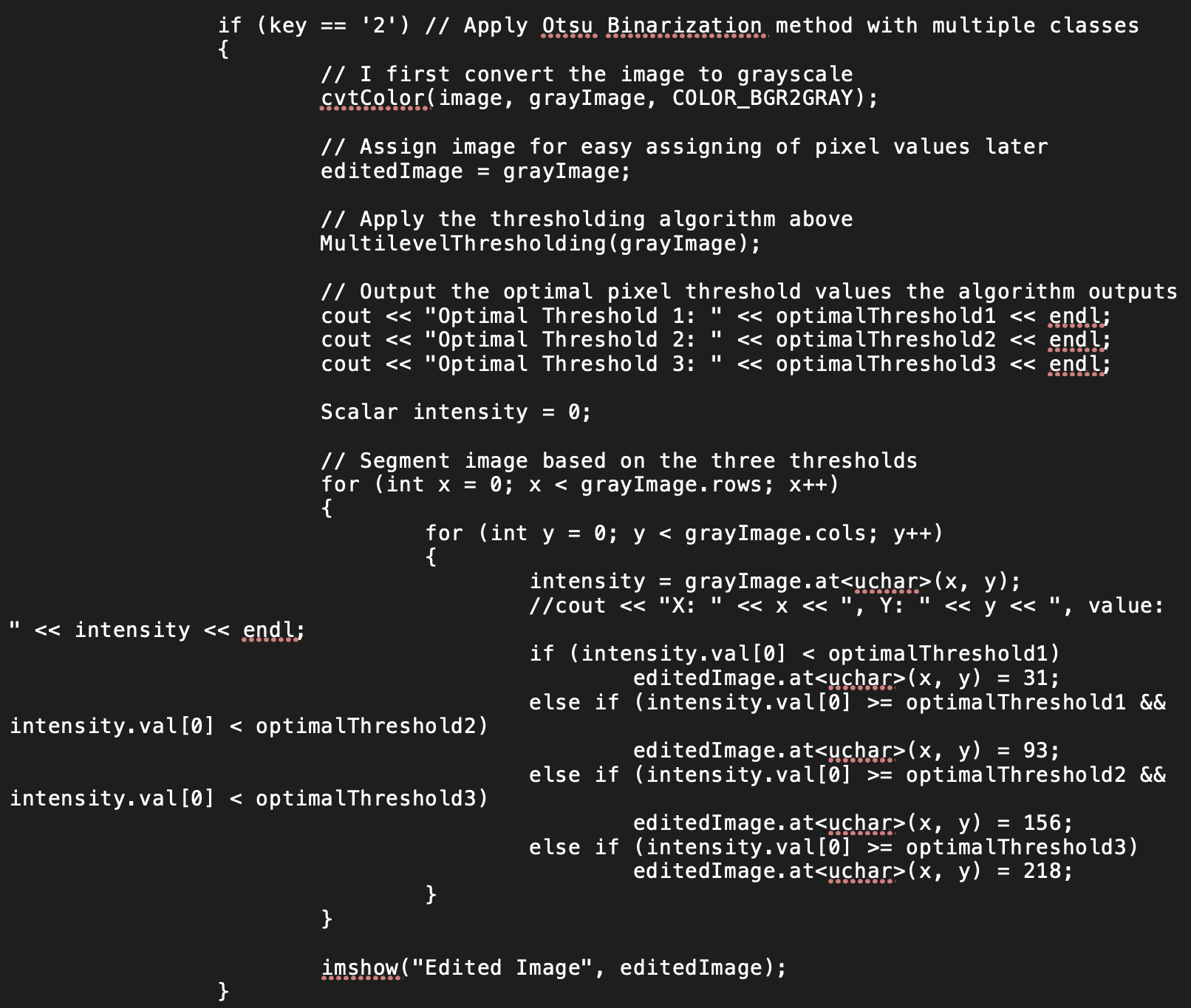
## Discussion

The method works as expected. The algorithm finds the optimal threshold value that minimizes weighted with-in class variance for each images and segments it accordingly. The color method works well on the color test image. The method is applied to each color channel of the image individually, then combined into a color image. Each output image has two colors: black and white (except for the color image), as expected.

# Otsu Method with Four Classes

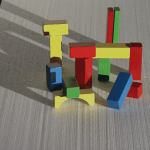
## Implementation

I researched online for a method of implementing Otsu’s method with four classes and found an algorithm on StackOverflow that worked with my program [3]. This algorithm, multilevelThresholding(), takes a one channel MAT image as a parameter, and outputs the three optimal threshold values that would minimize the weighted with-in class variance between all classes. Then, I segmented the image with these thresholds, by assigning all pixel values within certain ranges in the edited image to be new values and by using if-else statements. The math performed in the background is similar to the math involved in the Otsu Method with Two Classes, just for four classes.

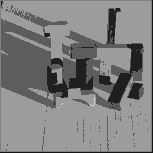


**Figure 5: Otsu Method with Four Classes Implementation**

## Testing



**Figure 6: Otsu Method with Four Classes Test Images**



**Figure 7: Otsu Method with Four Classes Output**

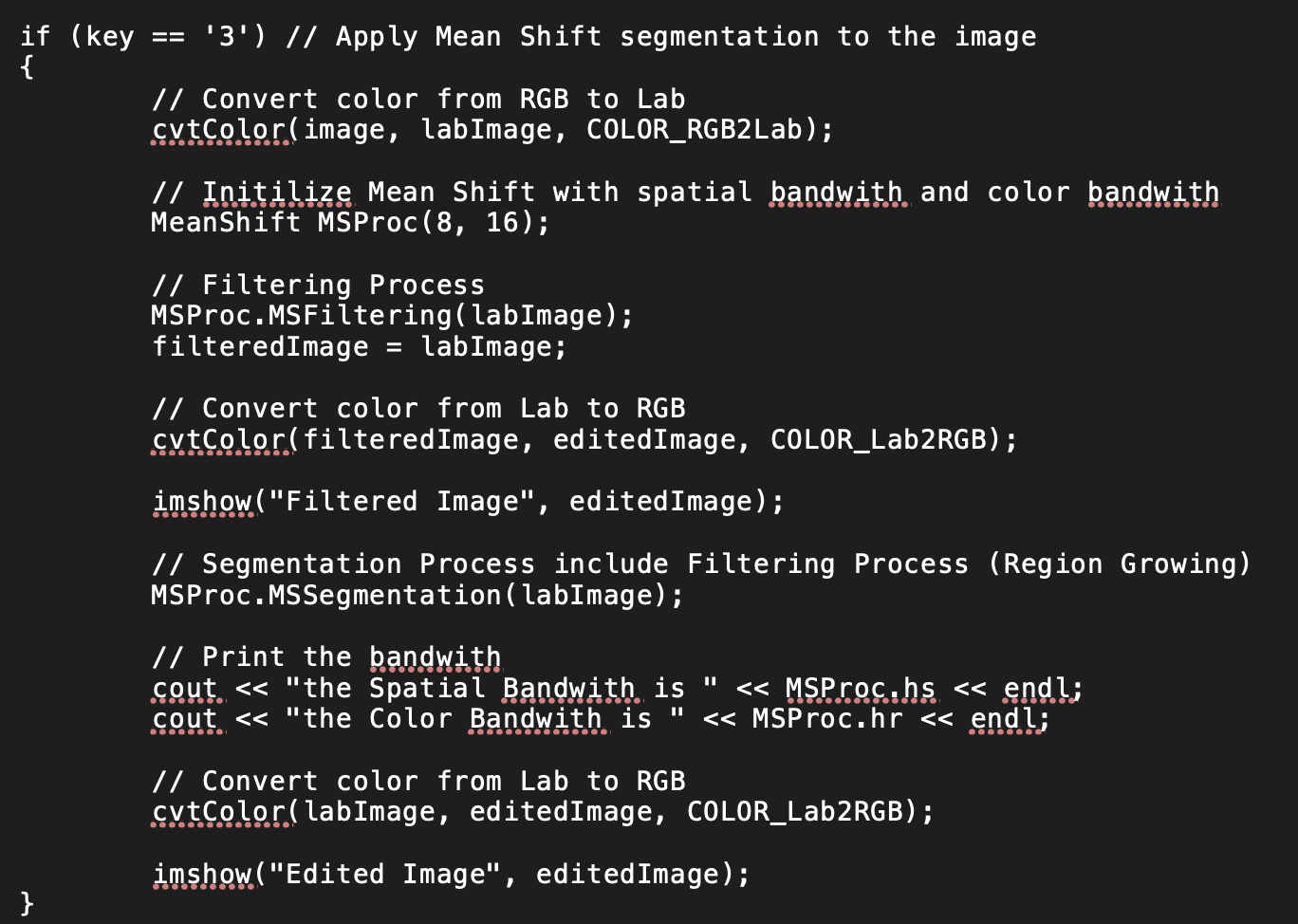
## Discussion

These results are as expected. In each image, pixels are reassigned to one of four ranges depending on their intensity, which clusters similar colors and results in four shades of gray in the image. This can be observed in each of the output images above.

# Mean Shift Method

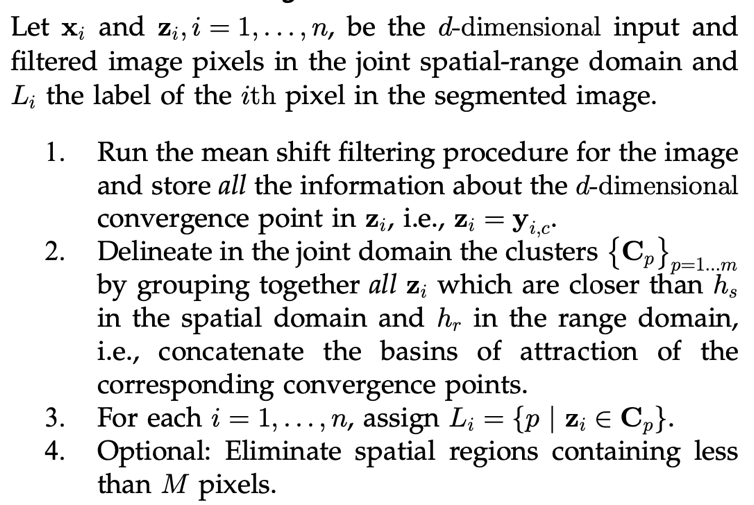
## Implementation

I read about this algorithm in the lecture slides and on the internet. I found an implementation of mean shift using OpenCV [4]. I added the MeanShift.c and the MeanShift.h files to my program and included MeanShift.h file in my main.cpp file. My code is very similar to the code on GitHub, I did not need to change it much.

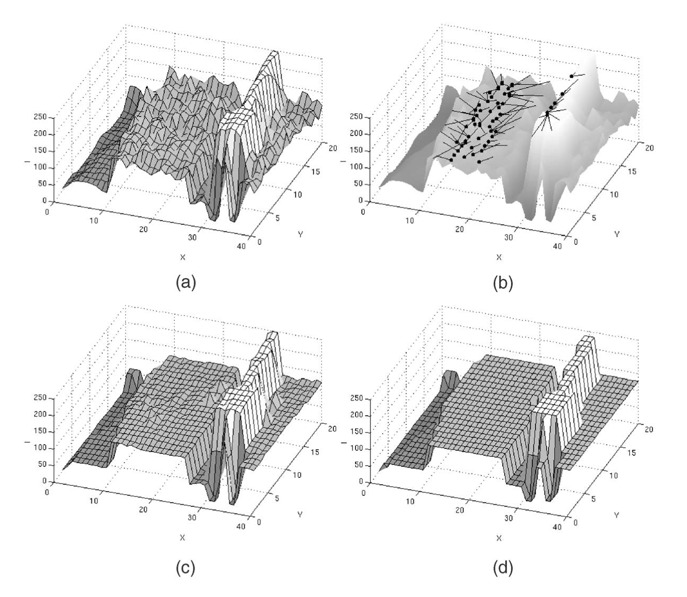


**Figure 8: Mean Shift Method Implementation**

The mean shift segmentation algorithm works by analyzing a complex multimodal feature space and determining clusters of the data [5]. Mean shift is based on an old pattern recognition procedure, and mean shift is application recursively on the image until every pixel is a part of at least one segment. Researchers proved that recursive mean shift converges at the nearest feature stationary point of the probability density function, which means this procedure is very useful for finding dense modes in the image [5]. This is what allows segmentation to occur.



**Figure 9: Mean Shift Segmentation Method** [5]

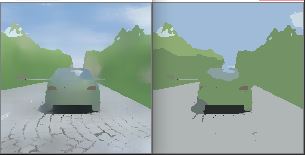
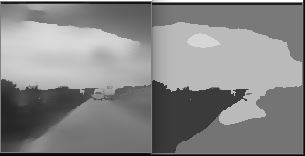
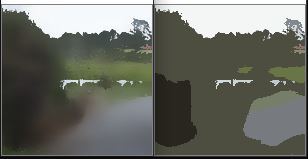


**Figure 10: Mean Shift Method Visualization** [5]

## Testing



**Figure 11: Mean Shift Method Test Images**



**Figure 12: Mean Shift Method Output**

## Discussion

These results are as expected. The first image of each pair is the image that is filtered before being processed. Each image is clustered based on it is pixel values which is useful for computations and other applications. The clusters are formed logically and not randomly. The mean shift segmentation algorithm is working correctly.

References

[1] “Miscellaneous Image Transformations,” *OpenCV*. [Online]. Available: https://docs.opencv.org/master/d7/d1b/group\_\_imgproc\_\_misc.html#ggaa9e58d2860d4afa658ef70a9b1115576a95251923e8e22f368ffa86ba8bce87ff. [Accessed: 30-Oct-2019].

[2] “Image Thresholding,” *OpenCV*. [Online]. Available: https://docs.opencv.org/master/d7/d4d/tutorial\_py\_thresholding.html. [Accessed: 30-Oct-2019].

[3] natan337natan337 2666 bronze badges, “multi-level (4) Otsu thresholding,” *Stack Overflow*, 01-Apr-1966. [Online]. Available: https://stackoverflow.com/questions/35056760/multi-level-4-otsu-thresholding. [Accessed: 30-Oct-2019].

[4] "L. Bingyang, “bbbbyang/Mean-Shift-Segmentation,” *GitHub*, 11-Feb-2019. [Online]. Available: https://github.com/bbbbyang/Mean-Shift-Segmentation. [Accessed: 30-Oct-2019].

[5] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.