AERIAL IMAGE SEGMENTATION

by

Abdulkareem Ali Althwaini

A thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Science (MSc) in Computational Sciences

The Faculty of Graduate Studies

Laurentian University

Sudbury, Ontario, Canada

THESIS DEFENCE COMMITTEE/COMITÉ DE SOUTENANCE DE THÈSE
Laurentian Université/Université Laurentienne
Faculty of Graduate Studies/Faculté des études supérieures

Title of Thesis Titre de la thèse	AERIAL IMAGE SEGMENTATION	
Name of Candidate Nom du candidat	Althwaini, Abdulkareem	
Degree Diplôme	Master of Science	
Department/Program Département/Programme	Computational Sciences	Date of Defence Date de la soutenance July 25, 2016

APPROVED/APPROUVÉ

Thesis Examiners/Examinateurs de thèse:

Dr. Ralf Meyer
(Supervisor/Directeur de thèse)

Dr. Lorrie Fava
(Committee member/Membre du comité)

Dr. Julia Johnson
(Committee member/Membre du comité)

Dr. Abdellatif Serghini
(Committee member/Membre du comité)

Dr. Sabah Mohammed
(External Examiner/Examinateur externe)

Approved for the Faculty of Graduate Studies
Approuvé pour la Faculté des études supérieures
Dr. David Lesbarrères
Monsieur David Lesbarrères
Dean, Faculty of Graduate Studies
Doyen, Faculté des études supérieures

ACCESSIBILITY CLAUSE AND PERMISSION TO USE

Abstract

Image segmentation plays a vital role in applications such as remote sensing. For this example, remote sensing, aerial image segmentation is a special case of image segmentation. There are some unique features of aerial images, like noise in natural landscapes, which need to be addressed in order to obtain an optimal solution. Bushes and rocks are examples of landscape features with diverse and variable pixel values that need to be distinguished by the segmentation process. Smoothing filters present a common solution to address the problem of noise in images, as does aerial image segmentation. There are several image segmentation techniques used for aerial image segmentation. Some of these techniques are more sensitive to noise problems, and are necessary to discriminate between different smoothing filters. In this thesis, a number of different aspects of aerial image segmentation and their solutions are explained. In addition to this, a novel smoothing filter is introduced and compared with other methods using different segmentation techniques. Finally, all of the previous points are applied to a real world problem.

Keywords

Aerial image segmentation, smoothing filters, k-means classifying, c-means fuzzy classifying, eigenvalues, singular values and climate change detection.

*To you, Dr. **Ali**…*

I wished you were here today, hopefully someone is going to tell you.

*To you, Mrs. **Nabila**…*

I see you in each page of this work; please stay.

*To you, **Abdullah**…*

Thank you!

Acknowledgments

First and foremost, I offer biggest possible thanks and appreciations to my supervisor, Dr. Ralf Meyer, who gave me the chance, support, knowledge and patience during my Master's Degree studies. I believe no supervisor could be better than who God gifted to me.

A very special thanks goes out to Dr. Abdellatif Serghini. Without talking a course with him and benefitting from him in my committee, this thesis would not exist in its present form.

Also, I would like to thank the rest of my thesis committee: Dr. Fava Lorrie and Dr. Julia Johnson, for their motivation and encouragement.

Finally, and most significantly, I would like to acknowledge and thank the government of Saudi Arabia for giving me this chance to improve my knowledge and study abroad.

Table of contents

List of Figures

List of Tables

1. Introduction

Image segmentation, which is a process of partitioning a digital image into multiple meaningful parts, is a highly functional method for interpreting images. Differing from hand-drafted maps or advanced software for map drawing, aerial images provide realistic representations of the land surface, and produce accurate maps as well. Images in general, require human interpretation to be understood visually, yet, image segmentation techniques utilize images of the digital world to reveal and permit access to far more detailed and depthful data.

Since the 1960s, there have been several major changes in the field of aerial imaging and image processing. According to Baumann (Baumann, 2009), the spark for many of these changes occurred in the 1960s and 1970s. The term "remote sensing" was initially introduced in 1960. Also, the 1960s and 1970s saw the primary platforms carrying remotely-sensed instruments shift from airplanes to satellites. Satellites survey much more land space than airplanes, and can monitor areas on a more regular basis. Later, an imaging revolution began when images were created in digital, rather than analog, format. The digital format computerized the whole field of data imaging. Finally, sensors that recorded the Earth's surface simultaneously in several different portions of the electromagnetic spectrum became available. One could now view an area by looking at several different images, some in portions of the spectrum beyond what the human eye could see. This technology made it possible to see things occurring on the Earth's surface that could not be detected by looking at a normal aerial photograph.

Developments in image processing techniques continue. Researchers study computer vision and develop technologies that mimic human vision every day. Unlike human visual ability, computer

vision is continuously refined to reach higher benchmarks. To that end, imaging technologists define each vision task according to its specific requirements. For example, aerial images have various formats and information that can be represented according the desired results. For example, radiation images for land surfaces such as cosmic rays, gamma rays, and X-rays, can see and segment better than the human eye. Thus, changes and developments in both fields (computer vision and aerial imaging) produce further justification to study each technology independently and in combination.

One of the challenges in the field of aerial image segmentation is the common area variance. Image segmentation is typically affected by small, meaningless parts (in the original image). Small parts in the image are considered noise, since they lead to unwanted small parts in the results. For example, bushes need to be labeled all as one area, but may exhibit significant diversity between pixels (within the bush area itself). According to (Yu Qian, 2004), "Since pixels of the same land use/cover class may not have similar spectral property, methods based on spectral analysis can produce results that are "noisy" due to the high spatial frequency of the land covers. Moreover, the popular classification algorithms are based on single pixel analysis, producing a geometric outline of land covers that does not correspond to real spatial entity representation such as fields, roads, and streams". The diverse nature of landscape coverings may be interpreted as noise when it comes to image processing, since it allows for unwanted clusters. In this thesis, I consider any unwanted details in images as noise, and for the aerial images segmentation scenario our definition is refined to "any details in the image that lead to unwanted clusters". In other words, where there is a fair similarity in two segmentations of the same images, the less minutely detailed result is considered optimal. This circumstance suggests a

need to pre-process aerial images. Accordingly, a smoothing filtering is needed to process such images. (Rosin, 1994), have stated: “Depending on the classification system, further format conversion stages may be necessary. Inaccuracies in outlining the training areas on the image can cause boundary pixels with either incorrect labels or mixed class signatures to be introduced into the training set. Every step in the data preparation is a potential source for introducing errors. The spectral filtering technique we describe is one way of reducing the effects of these errors”. Although there are several solutions and approaches to aerial image filtering that achieve better segmentation results, users of the technology still argue about which filter to use, and how to apply filters to aerial image processing. However, user preferences are often due to variances in their objectives, and to the fact that there is no one correct approach, but potentially many. Still, the use of smoothing filters presents a highly advantageous approach to aerial image segmentation.

In the search for an optimized aerial image segmentation that is less noise, this project addresses the aerial image segmentation results based on different factors such as the segmentation method, image channels used, and filters adopted. I have built several components software to achieve this goal. The software supports the process of aerial image segmentation, channel selection and filtering. Also, the software includes several evaluation and analysis tools that help interpret the differences in aerial image segmentation results. For the experiment, I perform two tasks. Firstly, I address the different factors (segmentation methods, channels selected and filtering) that produce effects on noise reduction. Secondly, I perform a water shield measurement under the same conditions to address effects on climate change measurement.

2. Background

This chapter is a review of several aspects of aerial image segmentation based on related work. Starting from image representations (possible channels to use), the chapter proceed over spatial filters until the core process: image segmentation. This review provides a better understanding of what has thus far been accomplished and achieved.

2.1. Aerial Image Representation

Digital aerial images, like other digital images, might contain several channels. If not a gray image (one channel), primarily colour information is available. Also, artificial channels can be generated, such as texture channels, which result in texture analysis of an image. For each of the previous representations, pixel values reflect the domain of the image (or domains if multiple channels occurred). Below, most of the previous possibilities are described in detail.

Gray level, sometimes called intensity, images are the first images used in photography and digital images as well. These images represent the intensity of an observed image. In other words, they represent the values of an image as perceived by human eyes and brain. Although gray level images are the first digital images that were used in image processing, they are used widely and successfully for many purposes. Currently, there are many applications that adopt gray level images alone.

In this thesis, the gray channel of a standard JPEG or PNG image format that comes in multichannel is determined a linear combination of the RGB (Red, Green and Blue) colour intensities using in the following formula:

*Gray = (0.299 * R + 0.587 * G + 0.114 * B)*

Where R is the red, G is the green and B is the blue channel of an RGB based image.

If an image is not a gray level channel only, another type of representations is quite possible: the colour information channels. Those images (colour images) might have the colour information values mixed with the intensity between channels or separated as well in a channel. Examples of the mixed types are the RGB (Red, Green and Blue) colour spaces, which are quite similar to the CMY (cyan, magenta and yellow) colour space as well. In the RGB and CMY colour space, there are three intensities of channels and each of those channels correlate to the intensity of one colour. The three channels reflect the colour information of the image, and the intensity, at the same time. For example, in the float domain from 0 to 1, if all three channels in a pixel are the same (no difference between channels) and low, this pixel will look gray and dark, if not black. On the other side, it will look lighter, as the three channels are obtaining higher values at the same time (all channels together). Still, if the red and green channels are lower than the blue channel, the pixel will look bluish, and it will be light bluish if all channels are higher and the opposite (dark bluish) if all channels are lower. For the CMY colour space, it is somehow opposite of the RGB. In the CMY, channels are equal to one minus the RGB channels. That means the cyan will be equal to one minus the red, and the magenta with the green and yellow with the blue as well (Gonzalez, 1987) (Castleman, 1979) (Pratt, 1978).

A different way to represent colour information is giving by the Hue family colour models. This family contains the HSV (hue, saturation and value), HSI (hue, saturation and intensity) and the HSL (hue, saturation and lightness). These three models represent a cylinder-like model. The cylinders' axis represents the hue, and the distance from the axis represents the saturation. The distance along the axis is either lightness, value or brightness. What interests me in this thesis is the Hue channel, which is a wheel-like colours temperature scale, since it provides rich information about colours temperature in an image. The third channel of these three models is quite similar to the gray (Knudsen, 1999) (Bernice Ellen Rogowitz, 2007) (Yud-Ren Chen, 2005).

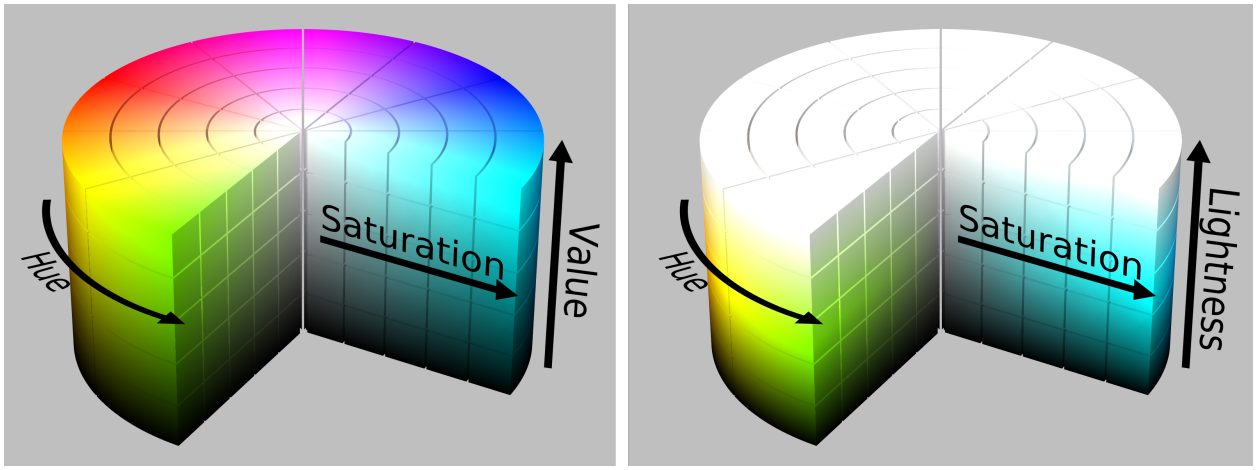

Figure 1 The HSV colour model mapped to a cylinder and The HSL colour model mapped to a cylinder. Image rights: Michale Horvath.

Also, there are artificial channels that can be generated for several purposes. Textures channels are examples of artificially generated channels of an image that reflect the texture analysis in the spatial domain. Another example is the co-occurrence matrix that generates several statistical

analysis values, such as the contrast, energy, and homogeneous values of the spatial domain (Pratt, 1978). There are several techniques to generate textures channels. This thesis uses a very basic process, which is the variance spatial filter. As with any spatial filter (Section 2.2), variance spatial filters assign each pixel with a new value based on the variance of the surrounding area (filter box). That means for a filter with a box size of 3*3 (which I use in this thesis), pixel in place 2,2 will be assigned with the variance value of 9 surrounding pixels. The variance filter is assigning the variance of the filter box values to the center pixel.

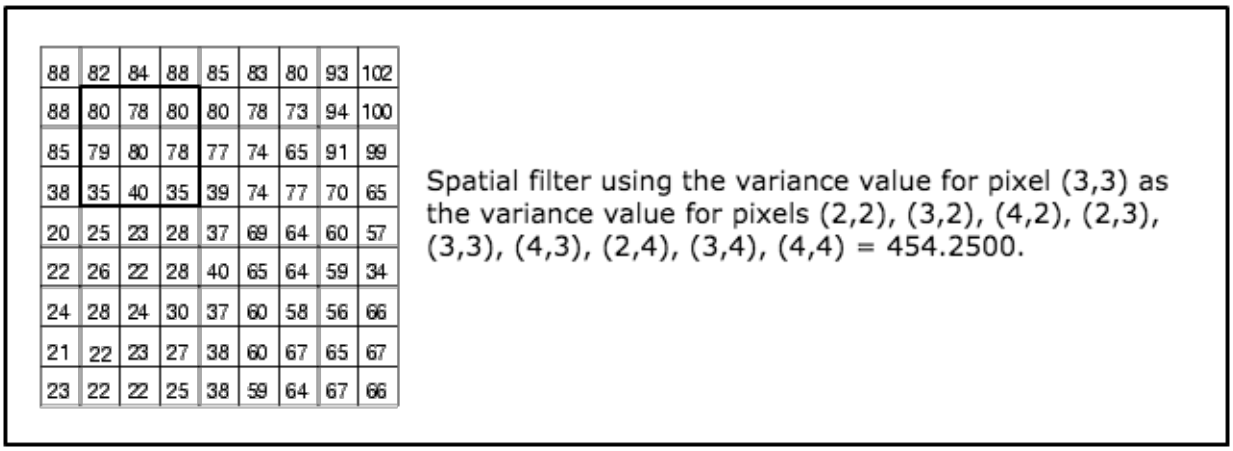

*Figure 2 Spatial filter with a filter box size of 3*3 using variance value.*

To assure that all images used are going to be in the scale from 0.0 to 1.0 and to get rid of unaffected borders, each filtering in this thesis will be followed with a normalization. The normalization starts by cropping the sides of the image by the size of the filter box, abstract the minimum value in the image from the whole image, divide the whole image by the maximum value of the image as well.

2.2. Spatial filtering

Digital image filtering is a common process for many different scenarios. Enhancement, editing and noise removal are cases where spatial image filtering is needed. This thesis deals with aerial images rich with noise, in other words, many unwanted details. More specifically, for image segmentations, we define noise as any detail that leads to unwanted segmentation results. In this thesis, I look to optimize an aerial image segmentation solution that gives a generalized idea of the image with respect to the main components in the original images. Simply stated, representing the main components of those images, regardless of small details that are noise, is an objective. Some applications seek to highlight those small details of the image such as sharpening. However, that is not the case in this thesis.

Reviewing digital image filtering reveals that there are two main ways of filtering. The first is pixel domain filtering, where pixels are modified by multiple mathematical or statistical operations, based on value of the pixel alone. This type of filtering applies the same operations to all pixels in the image regardless of the surrounding pixels values. An example of this kind of filter are mathematical log filters or inverse filters. These filters do not change images based on their components; they change all image pixels together. The other type of filtering, called spatial domain filtering, involves filters that conduct several operations with the spatial neighbors' data involved. (Pratt, 1978) (Castleman, 1979). There are two types of spatial domain filters: convolution and non-convolution (Bourne, 2010).

2.2.1. Convolution filters and non-convolution filters

“The linear filters are termed as convolution filters and non-linear filters are called non-convolution filters. Convolution filters use the neighborhood values in a linear fashion. Whereas non-convolution filters use the neighborhood values differently, which is discussed in the weightings of the masks used in convolution operation are also termed as kernels in image processing literature. Several commercial softwares allow the specification of the kernel by the user” (Ramesh, 2000).

Convolution filters are performed by kernels (filter box) applied to the image areas and resulting in new values. In other words, it acquires the sum of the kernelled area by kernel numbers (weights). Examples of convolutions are the mean filter and the Gaussian blur filter. In non-convolution filters, it is not necessary to apply mathematical operations such as those used by kernelling. Non-convolution can use statistical operation and select a value from the filter box. Examples of non-convolution are the statistical mode filter and the median filter.

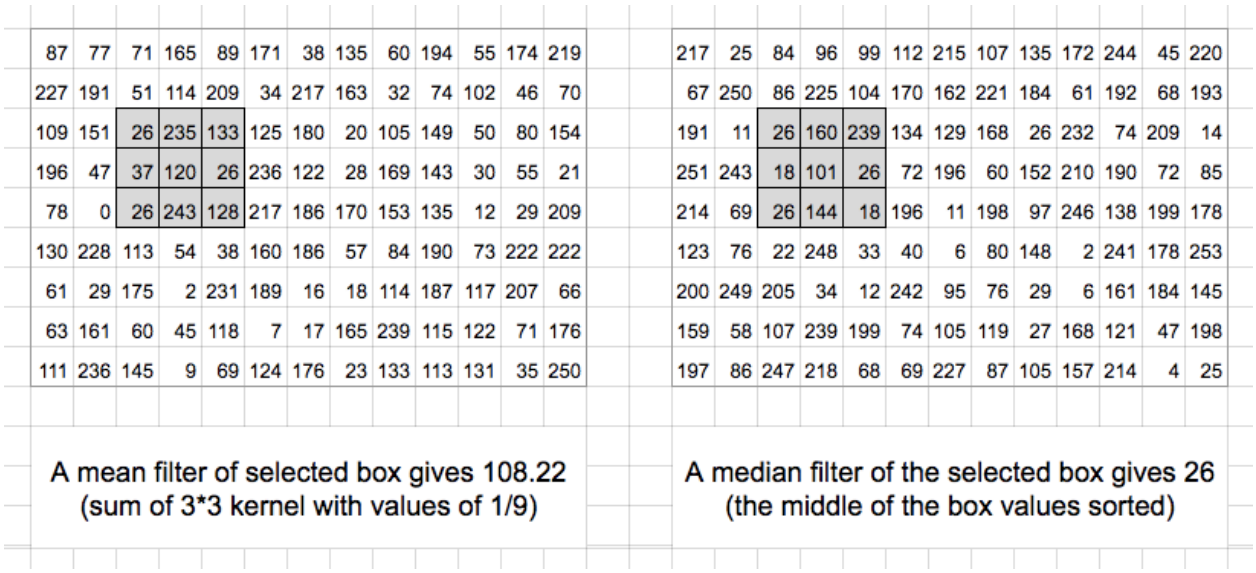

Figure 3 From left to righ: Example of a convolution filter (mean) and a non-convolution filter (median)

In this thesis, I propose a novel filtering technique using the largest singular value (LSV). Based on what reviewed so far, this technique has never been used as an image spatial filtering. The proposed filter is a neighborhood operator, in which the value of any given pixel in the output image is determined by applying the norm 2 of a matrix (interpreting the filter box as a matrix) whose entries are the values of the pixels in the neighborhood of the corresponding input pixel. This is equivalent to replacing the value of any given pixel with the largest singular value of the filter box.

If a matrix is a real symmetric matrix with non-negative eigenvalues, then eigenvalues and singular values coincide, but it is not generally the case. Since singular values of a m*n matrix are the square roots of the eigenvalues of the n*n matrix ATA (where AT is the transpose of A), I will use the largest singular values instead of the dominant eigenvalue. The main motivation for experimenting with the LSV filter beside the fact that it never being used for image spatial

filtering before, is its ability to determine the "size" of the filter box using the 2-norm (which is the largest singular value as well). The norm of a matrix is a real number which is a measure of the magnitude of the matrix. Matrix norms are ways to measure the "size" of a matrix, one way to do that is using the 2-norm or the so-called the largest singular value (LSV) norm. By the end of this thesis, it will be clear how this proposed filter behaves.

From a mathematical point of view, if A is symmetric, we say A is positive semidefinite if x^T A x >= 0 for all x, denoted A >=0. A >= 0 if and only if all eigenvalues are nonnegative. While in aerial imaging A is not necessarily square or symmetric, for x ∈ Rn norm(A x) / norm(x) gives the amplification factor or gain of A in the direction x, the maximum gain max norm(A x) / norm(x), is the spectral norm (norm 2) of A and is denoted norm(A), and norm(A) = sqrt(lambda_max (A^T A)). Note that A^T A is symmetric, A^T A >= 0 and norm(A) = sigma_1 (largest singular value of A).

Beside the LSV filter, I will use other filters to address the effect of filtering when compared to the one proposed. There are several filters to use for such a goal, but I will simplify this project by selecting only three filters. The filters I use are the statistical mode, median blur, and Gaussian blur. The statistics mode filter generalizes the image area by giving the filter box center the most frequently occurring value. Thus, it removes the rare and weakly occurring values, which give a more generalized idea of the whole image. The median filter is quite popular in image filtering, and it offers one of the best solutions for the well-known “salt and pepper noise” problem. A median spatial filter is a strong smoother, and is supposed to show a good result for

aerial image segmentation. The last filter examined is the Gaussian blur. A Gaussian blur spatial filter is not less known than the median (and may in fact be better known) and it has been widely and successfully employed for image noise reduction (Zhouping Wei, 2012). Mathematically, Gaussian 2d smoothing filters affect an image by convoluting the image with the Gaussian function.

For the mode filter is assigning the most occurred value in the filter box to the center pixel. For the Gaussian filter, I use a Gaussian blur filter provided by the OpenCV library (Opencv, 2016). Finally, for the median filter that it works by having all of the filter box values in an array, sort the array and select the middle value.

All of the previous filters can be applied individually to all image channels (one channel at a time). As mentioned in Section 2.1, to assure that all images used are going to be in the scale from 0.0 to 1.0 and to eliminate unaffected borders, each filtering in this thesis will be followed by a normalization and cropping that been mentioned before.

The largest singular value supposes to be normalized by dividing it by one of the filter box dimensions, but I have it scaled between 0.0 and 1.0 like other filters that should end with the same results.

By having the previous combination of filters (the new filter and the other three filters), I can perform segmentation scenarios to address the differences between filters used in seeking an optimized solution for the aerial image segmentation.

2.3. Image segmentation

Image segmentation is the partitioning of an image into meaningful parts. Meaningfulness varies between applications, but in this thesis I focus on natural landscape elements (mostly water, vegetation and rocks). There are multiple techniques for image segmentation, but for aerial image segmentation, not all of these techniques work well. In this thesis, I will use two terms for the segmentation results. Classes is the term for the number of partitions, and clusters is the term for individual partitions.

2.3.1. Edge based image segmentation

An image's edge refers to a border between different areas within the image. Edge detection is the process of highlighting areas by the discontinuity of its values, and is accomplished by detecting its boundaries. For many images, edges reveal important details about the component's borders. There are many successful approaches to image segmentation based on edge detection (Farag, 1992).

In edge detection, the main idea is to use filters that are sensitive to value changes in the image's spatial domain, which indicate that the area is changed. There are many techniques for performing edge detection such as Roberts Edge Detection, Sobel Edge Detection, Prewitt Edge Detection and Canny Edge Detection (Radha, 2011).

Aerial image segmentation does not always show an organized area, and it has a lot of naturally distributed components. For example, this may be result where bushes include variant values that cannot be defined by boundaries due to the occurrence of many boundaries within the area itself. When using edge detection methods for such aerial images cases, they will mostly be affected by the image textures and will be wrongly interpreted as edges. For example, bushes are likely to be full of small edges, and not organized border edges. Therefore, for images of the land surfaces, it is inadvisable to use any edge based image segmentation techniques.

Figure 4 Edge detection applied to a photograph, copyright: Jon McLoone at English Wikipedia

2.3.2. Region growing based image segmentation

Region growing is another common technique in image segmentation. The concept begins with selecting a point, and includes connecting all similar values into bigger clusters. This technique also works very well with edge detection, since all bounded areas can be filled after identifying borders (Pratt, 1978). Region growing might be an effective option to use after other segmentations to identify specific clusters.

The down sides of this technique are connectivity and supervision. This technique is a supervised technique, where the user has to be familiar with the areas to be detected. Such requirements make it difficult for aerial images, since the aerial image components are disconnected in between by values diversities. For this reason, the region growing technique will not be used in this thesis.

After segmenting an image, I identified clusters in the segmentation results. For example, if am image is full of small lakes, I segmented the image into two classes. If the segmentation was successful, multiple discontinued lakes (clusters) will share the same class label. In such a case, the discontinued clusters require a unique label, or the focus is on one cluster of the image. For these two cases, I will be using the Hoshen Kopelman algorithm and the flood fill algorithm as well. The Hoshen Kopelman algorithm performs a unique identification for all connected clusters of an image. For example, if there is an image with one value and it has four disconnected shapes with one value for the four shapes, and with a fifth value for the background, this algorithm outputs an image with 5 values for four shapes and a background.

The other algorithm I am going to use is the flood fill. The flood fill algorithm works by selecting a point within the region of interest. After, the selected point spoiled from all similar values to the starting point, which results in a selection of one connected area of interest. As discussed above, the region growing method is not ideal to apply to aerial images directly. Yet, it is quite appropriate for the segmented result of aerial images.

2.3.3. Thresholding based image segmentation

Thresholding is one the simplest and well-known image segmentation techniques. It works by defining a point of threshold values and it results in any pixel that fits between the thresholds in one class. Thresholding is usually accomplished with the help of the image histogram, which shows the occurrences of the values. This gives an idea of the image values in general. Such an understanding of the image can help find which thresholds of the values will cluster the region of interest. For several areas, the algorithm must be implemented several times, and this algorithm requires the user to be knowledgeable of the selected images (Gonzalez, 1987).

One of the challenges in thresholding is that it is a supervised method and calls for previous knowledge of an image to achieve segmentation. A user has to know the image values, and to be familiar with the imaging system used, in order to select the thresholds. This challenge makes such a technique inappropriate for this thesis, as well as the noisy nature of the aerial images and the huge diversity and changeability of its values.

2.3.4. K-means classifying based image segmentation

The k-means algorithm falls between a semi-supervised and an unsupervised method. In some works, it is called k-means clustering algorithm. It classifies data based on its distance from k centroids. It is an iterative technique that works by measuring the data value's distance from the initialized centroids iteratively, assigning the values to the closest centroid and updating the centroids to the mean of its values in each iterate. This algorithm can be used in image segmentation and it shows good results, and also helps recognize image parts without previous

knowledge. K-means image segmentation considers image values as any other data; the user defines k number of centroids and the algorithm starts by measuring the distance of each pixel in the image from all of the centroids. Each pixel is tagged to the centroid with the least distance. After, the centroids are updated by accessing the mean value of all centroids tagged to that centroid. After a couple of iterates, the changes in the centroids converge to zero, which is a sign that the pixels have been segmented fairly.

The k-means algorithm is quite appropriate for this thesis, since it is largely an unsupervised method (except centroids initialization). Deciding upon the number of clusters is the only semi-supervised component. The k-means algorithm does not require a previous knowledge of images, and it can handle significant changes between images. It is an accurate method for discriminating between different filters that are used in the processing of aerial images. I will be using this method in this experiment.

There are two challenges in k-means image segmentation that need to be addressed. The first is selecting the right initial values for the centroids. The k-means algorithm is quite sensitive to the initial centroid values. Having the same image segmented with different centroids values could lead to different segmentations results. Also, careful seeding for initial class centers to improve segmentation results is quite important (Pavan, 2011). This challenge is quite sensitive, since I am looking for a stable segmentation to assure a fair comparison between filters.

There are several different methods to select the initial centroids instead of selecting them randomly. Also, having static criteria to select the centroids will assure the same segmentation

factors over images and filters. In this thesis, I will use a density method that attempts to create the most distance between seeds based on the values density over an image. This method uses a median division between the image data scale with a respect to the data density. If the image's values density is the same all over (a flat histogram), an example of three centroids in the scale of 0.0 to 1.0 will results centers (0.0, 0.5 and 1.0). If the density is uneven the centroids will move accordingly. For example, if the lower values occur more frequently in the image, the centroids are going to shift to the left.

The second challenge of the k-means algorithm in image segmentation is to decide how many divisible classes are possible in the image. For example, if a one-value image is used and we attempt to segment it into two clusters, the result will be meaningless. In other words, having high k numbers for segmentations will lead to some confusion in the results. Since I am focusing on the main components, I use the centroids k=3.

In this thesis, I will be using a standard k-means algorithm for image segmentation as follows:

1- Initialize:
- centroids[0.. k-1] = 0 where k = number of centroids
- results[0.. n-1][0.. m-1] = 0 where n and m are the dimensions original image.
- distances[0.. n-1][0.. m-1][0.. k-1] = 0 where n and m are the dimensions original image.

2- For each pixel in image(i,j), find distances to centroids[0.. k] as:

```
for i = 0 to n-1
  for j = 0 to m-1
    for l = 0 to k-1
      distances[i][j][l] = abs(image(i, j) - k[i])
    results[i][j] = l where min(distances[i][j][l]
```

3- Update centroids as:

```
initial sum[0.. k-1] = 0 and count[0.. k-1] = 0
for l = 0 to k-1
  for i = 0 to n-1
    for j = 0 to m-1
      if( results[i][j] = centroids[l] )
        ++count[l]
        sum[l] += image(i,j)
  centroids[l] = sum[l] / count[l]
```

4- If there is a change in the centroids values, go to step 2

5- Else return image result(x, y)

Formula 1 K-means classifying algorithm for image segmentation

2.3.5. Fuzzy c-means classifying based image segmentation

Since the introduction of fuzzy set theory in 1965 (Zadeh, 1965), it has been used in many computer applications. The theory provides a way to implement reality into a binary machine. Fuzzy set theory provides for overlapping and partially belonging to a set. The k-means algorithm is one of the algorithms affected by fuzzy set theory. Fuzzy c-means algorithm is an application that joins fuzzy set theory and k-means classifying.

The first fuzzy c-means classifying algorithm was presented by James C. Bezdek and others (James C. Bezdek, 1984). In their paper, “FCM: The Fuzzy c-means Classifying Algorithm”, they applied fuzzy set theory to the traditional k-means algorithm. As the traditional k-means was successfully used for image data, the fuzzy c-means classifying algorithm has been used successfully in image segmentation.

In the fuzzy c-means, the centroid’s distance to the data points is measured by memberships. In other words, each pixel belongs to every one of the centroids by a membership that lies between 0.0 and 1.0. The sum of all relationships must be 1.0 as well. Like the k-means, it is an iterative method. Within iterations the centroids are updated until they achieve more stable and satisfying changes. Also, the c-means algorithm uses a fuzziness degree, which starts from 1 (not fuzzy) and higher (fuzzier).

One of the biggest drawbacks of the fuzzy c-means is its sensitivity. Fuzzy c-means is quite sensitive to the unique details and outliers that in aerial images are considered noise. Such sensitivity is quite beneficial in some applications, yet not in aerial images since it highlights small details in the image, which are not required for this thesis (Yang, 2007) (Zaixin, 2014).

However, the sensitivity problem is quite handy for this thesis in different ways. Since I am trying to measure differences between several spatial filtering techniques on the aerial image noises, the fuzzy c-means is a useful highlighter. In conjunction with the k-means, c-means can be the “sensitive version of the k-means” that reveals finer differences between the filters.

In this thesis, the fuzzy c-means algorithm is implemented as:

1- Initialize:
- centroids[0.. k-1] = 0 where k = number of centroids
- results[0.. n-1][0.. m-1] = 0 where n and m are the dimensions original image.
- memberships[0.. n-1][0. m-1][0.. k-1] = 0 where n and m are the dimensions original image.
- d = fuzziness degree.
- delta = the termination condition.

2- For each pixel in image(i,j), find memberships to centroids[0.. k-1] as:
for i = 0 to n-1
for j = 0 to m-1 $\quad \text{memberships[i][j][l]} = \frac{1}{(\sum_{l=0}^{l=k-1} \frac{abs(image(i,j)-centroids[l])}{abs(image(i,j)-centroids[l])})^{\frac{2}{d-1}}}$

3- Update centroids as:
for l = 0 to k-1
$\text{centroids[l]} = \frac{\sum_{i=0}^{i=n-1}\sum_{j=0}^{j=m-1} image(i,j) \times memberships[i][j][l]}{\sum_{i=0}^{i=n-1}\sum_{j=0}^{j=m-1} memberships[i][j][l]}$

4- If changes in centroids values greater then delta, go to step 2

5- Else return a de-fuzzified result as:

for i = 0 to n-1
for j = 0 to m-1
result[i][j] = l of max (memberships[i][j][l])

Formula 2 Fuzzy c-means classifying algorithm for image segmentation

2.3.6. Neural Network image segmentation

An artificial neural network is a distributed parallel processor that can be used for storing experiential knowledge. This type of network provides suitable solutions for problems, characterized by non-linear ties, high dimensionality noisy, complex, imprecise, and imperfect or error prone sensor data, and lack of a clearly stated mathematical solution or algorithm. A key benefit of neural networks is that a model of the system can be built from the available data (M.SEETHA, 2008).

Neural network image segmentation is also a successful image segmentation technique. It uses the technique of artificial neural network to segment images. The artificial neural networks are computationally very expensive. An advantage however, is that they are susceptible to machine learning, so that a computer can be trained. Neural network image segmentation is not used in this thesis because it is computationally expansive and the fact that machine learning is not within the goals of the project.

2.4. Analysis (measuring tools)

After reaching final segmentation results, it is necessary to measure those results numerically. This is because it is difficult to find differences between several channels, filters and segmentation techniques visually. Therefore, it is more effective to have numbers that explain the differences. Additionally, we are trying to optimize an aerial image segmentation solution. The optimization in this thesis is to avoid noise effects on the segmentation. In other words, I will consider a reduction in smaller details in the segmentation results as a better solution with a respect to the main large components of the aerial images.

2.4.1. Clusters counting (qualitative measuring)

The first measuring task in this thesis is qualitative. I need to measure how much the noise in aerial image segmentation has been reduced. For this, it is necessary to know the number of clusters found and analyze them by their sizes. If an image of lakes, bushes and rocks is

segmented using the k-means classifying with centroids k=3, this will end with a result image that includes many clusters having one of the three classifying numbers (zero, one and two). If the image is known, for example, to have three lakes, five areas of bushes and eight areas of rock, an ideal scenario will find sixteen clusters in the result image, but this will be mostly not the case due the noise sensitivity and diversity in the values for each of the natural areas. So, there is an assumption that the resulting image will have more clusters. This problem requires us to build a tool to count the clusters found in image, which is the Hoshen Kopelman algorithm mentioned before (Section 2.3.2). The Hoshen Kopelman algorithm is a useful means to count the clusters found in a results. Identifying the clusters opens the way to analysis of those clusters based on their sizes (large meaningful clusters or small noises).

2.4.2. Cluster sizes (quantitative measuring)

The second measuring task in this thesis, aside from noise reduction, is to measure changes over time. Climate change effects on land surfaces can be measured by image processing. Under multiple conditions (multiple channels, filters and two segmentation methods) I am going to address the changes in the climate changes. To begin, I need a solution to select a specific area within the segmentation results, which is the area to be measured. The key point here is to identify a selected point cluster (the cluster of the area of interest). The flood fill algorithm is an effective tool to address this problem (mentioned in Section 2.3.2). Flood fill is an algorithm to select the connected area (cluster) by selecting a point within it (Vandevenne, 2004). After selecting a specific cluster, I am able to analyze it over multiple effects.

This chapter reviewed several approaches and aspects of aerial image segmentation. Designing an improved solution depends upon the types channels in selected images (grey, coloured or textured). Also, the aforementioned smoothing filtering step is important for image segmentation. Filtering could change the segmentation process significantly. For the segmentation process, I use several options. I will use the k-means and the fuzzy c-means for the aerial image segmentation. Finally, I have reviewed the analysis methods that will be used as well. The main idea here is to answer which techniques are going to serve this thesis goal most.

3. Methodology

This chapter explains methods used to examine, evaluate and analyze aerial image segmentation techniques under multiple conditions. The experiment is done using several aerial images and employing software that performs channels selections, filtering, segmentation and analysis. Also, the software was developed using C++ and OpenCv library (Opencv, 2016).

3.1 Images used

Two tasks were performed in this thesis. The first task examined the segmentation of three highly detailed images. Those highly detailed images contain main components such as vegetation, water and rocks, so several naturally or artificially occurring structures. I have used the Google Earth application and selected three aerial images from the Northern Ontario region. The first image is from a farming area south of North Bay, Ontario. It contains a small river, and a couple of different farms. The second image is from the same area and contains a couple of different farms and bushes as well. The last image is for a fully natural area, containing bushes and rocks and several lakes.

Figure 5 From 1 to 3, first Image, second image and third image. Images courtesy of Google.

As can be seen in Figure 3, each image has a number of areas that can be labeled and each of those areas has a significant diversity of value. Such images address a common segmentation challenge: result clusters of main objects, which can be by eliminating small clusters (noises) that come within main objects.

The second group of experiments addresses a common need in aerial image processing, which is to measure changes over time. Climate change on land cover is a challenge to measure. Image processing provides an intelligent solution. By segmenting the image and having the area of interest (water shield) as one class, and then measuring differences in the same area in different images taken over time, a user can get a good impression of what has been changed. By using some images that show climate changes over time, I examined the factors that reveal changes. The United States Geological Survey "USGS" provide a valuable collection of climate change images for several locations in the world (USGS, 2015). I have chosen three images of a lake called "Bahr al Milh". Bahr al Milh (also called Lake Razazah) is located in Iraq, and fed by the Euphrates River via canal. Water levels of the shallow lake vary with the seasons; however, levels have been drastically low in the past decade, as can be seen in these Landsat images from 1995, 2003, and 2013. Such images are excellent examples of measuring changes over 18 years.

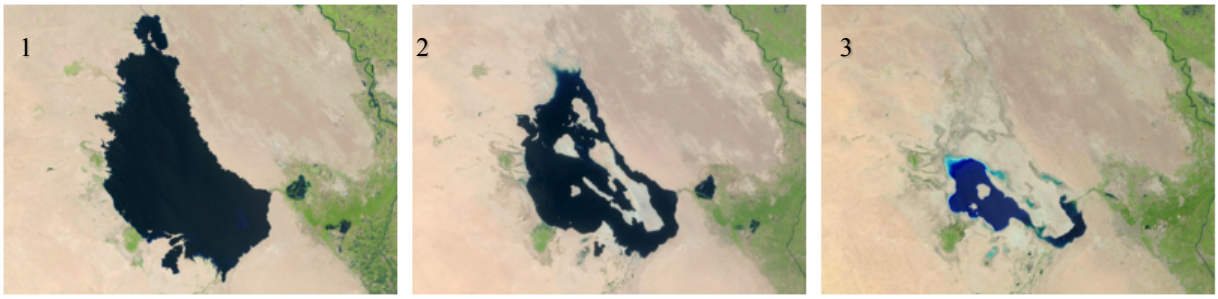

Figure 6 From 1 to 3, Bahr Al Milh 1995, 2003 and 2013. Images courtesy of the U.S. Geological Survey.

A second set of sample of images for change detection shows two images from the USGS. They are for Shasta Lake, California and show changes to the lake over a span of three years. According to the USGS: "As a 3-year drought continues in the western United States, water levels have been dropping in many California reservoirs, leading to emergency water use restrictions across the state. These two Landsat images show the changing shoreline of Shasta Lake reservoir in northern California over the past three years. The first image was collected in September 2011 and shows the shoreline when the reservoirs water levels were at 77 percent of total capacity. The tan colours in the September 2014 image show the change in shoreline. Even though snowmelt slightly increased the lake level earlier in 2014, the reservoir was still at only 27 percent capacity when this more recent image was acquired. The lower right portion of the second image also shows a recent burn scar from the Gulch Fire. This fire was officially contained one day before the September 17 image was collected". These two images (Shasta Lake) represent a good challenge for aerial image segmentation due to narrow areas of water (line like), which are sometimes lost in processing.

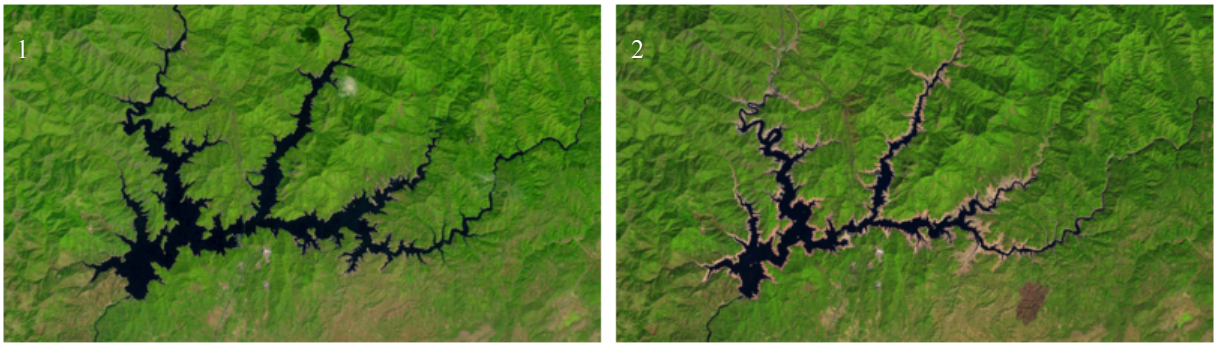

Figure 7 From 1 to 2, Shasta Lake, California 2011 and 2014. Images courtesy of the U.S. Geological Survey.

The three sets of images used in this work present useful challenges for two tasks: noise reduction (task one: Google images) and change over time (task two: USGS images). My goal in the first experiment is to examine the effect of multiple channels, filters and segmentation methods on the results of the images to perform a qualitative measuring (noise reduction). For the USGS images, I am doing a quantitative measuring of the changes over time under the same conditions as well (filters and segmentation methods).

3.2 Image smoothing filtering

As mentioned in Chapter 2, spatial filtering is quite important for image segmentation. This is especially true for aerial image segmentation, where an image's content may be quite noisy. There are two important factors to consider in filter use: the filter type, and the filter box size. Filters are different in their behavior, and bigger filter boxes give the filter greater ability to affect the result than a smaller filter box. A stronger filtering effect on an image may cause more loss the original image's information. For each experiment, chosen images channels are represented by the segmentation function as not filtered, mode filtered, median filtered, Gaussian

blur filtered and LSV filtered. For the LSV filter, I will use the SVD function for singular values calculation from the Opencv. Also, the same library is used for the Gaussian blur filtering as well. All filters will use a 3*3 filter box size.

3.3 Image segmentation

This step is the core phase of this thesis experiment. As mentioned in Chapter 2, there are several methods and techniques to perform image segmentation. Different from other methods, k-means image segmentation is a simple and powerful method at the same time depending on the vector quantization. Also, it is mostly an unsupervised method (no more then selection of centroids). The fuzzy version of the k-means is also beneficial for the experiment as has been mentioned as in Chapter 2.

In this thesis, I will use the k-means classifying for image segmentation and the fuzzy c-means classifying for image segmentation. For task 1, both methods are used with three combinations of channels (gray level, gray level with hue and gray level textures) as input with number of centroids k=3 and k=5. For task 2, both methods are used with the gray level alone as input with centroids k=3. Before using the two algorithms, there are two issues to address.

A challenging issue in using the k-means classifying or c-means classifying methods is the sensitivity for the initial seeds (initial centroids). According to (Pavan, 2011), "K-means is the most popular partitional classifying technique for its efficiency and simplicity in classifying large data sets. One of the major issues in the application K-Means type algorithms in class analysis is,

these are sensitive to the initial centroids or seeds. Therefore, selecting a good set of initial seeds is very important". In this thesis, I am looking for a method that provides stable seeds each time (longest possible distance between initial seeds). The median method for initializing the k-means seeds considers the value's accuracy and densities when selecting its numbers. The algorithm for the median based seeds that I am using is:

$for\ image\ X\ with\ dimentions\ M\ and\ N, clusters\ K,\ dense\ D\ =\ (m * n)\ /\ (k - 1)$

$Build\ \ a\ histogram\ of\ the\ image\ X$

$initialize\ an\ empty\ array\ seeds\ Se\ and\ variable\ sum\ Su = 0$

$for\ i\ =\ 0;\ i\ <\ hist.size(\ ) - 1;\ ++i$

$if(Su\ >=\ D * Se.size())$

$Se.push(i)$

$Su = Su + hist(i)$

Also, there will be a special case issue for the used k-means and c-means algorithms, where one of the channels is the hue. Unlike intensity values or RGB values, hue uses a periodic boundary. For this reason, the distance between its value and the centroid will be calculated as follows:

$if(distance\ >\ 0.5)$

$distance\ =\ 1.0\ -\ distance$

Addressing the previous issue assures that there will be no unnecessary changes between different implementations over different images and filters.

3.4 Analysis (measuring tools)

After completing the aerial image segmentation, it is necessary to analyze the results. As mentioned previously (Section 2.4), it is hard to find differences between many segmentation results visually, and it is more robust to have numbers that describe those results. For this reason, I use a couple of the analysis techniques previously reviewed.

I use the Hoshen Kopelman algorithm that Fricke provided (Fricke, 2000) to determine the number of clusters found in an image and to label them as well. Each segmentation result is processed using the Hoshen Kopelman algorithm. I use the output of the Hoshen Kopelman algorithm to measure the sizes of the clusters by initializing a histogram for the sizes of the clusters. Since the Hoshen Kopelman algorithm has labeled each cluster with a unique identifier (from 0 to the number of clusters), I can obtain the sizes of all those clusters. After having a histogram of all clusters sizes, I will find the count of sizes starting from the smallest possible and through to the maximum possible size (the whole image size). Finally, I have an accurate measure of how many smaller (meaningless) clusters and how many larger (meaningful) clusters the result possesses.

After observing the previous information of the aerial image segmentation results, it becomes clear how to discriminate between the different scenarios used (channels, filters, and segmentation techniques). One more task is left, which is observing the climate change effects over time on water shield. For this task, we have a specific area of interest, so it is necessary to use the flood fill algorithm. In this task, I start by identifying a specific area in an image and measuring its size over several other images based on the identified area.

After I collect, prepare, process and analyze several scenarios for aerial image segmentations in terms of the structure of functions, it is time to build a clear visualization for this data. I have built a web tool that visualizes the previous function's outputs and displays it in flexible graphs beside the results tables as well.

4. Results and Discussion

This chapter proceeds from the methods listed in Chapter 3. The implementation starts by select channels of each of the images (Google and USGS images). Secondly, performing different filtering to the selected channels results in passing the filtering results to the core step, which is segmentation. There are two segmentations techniques used: the k-means image segmentation and the fuzzy c-means image segmentation. The project ends by analyzing the segmentation results to reveal the results for the different scenarios.

4.1. Channels selection

The experiment starts by selecting the suitable channels from images. The point is to utilize differences between channels, so I can use the most dissimilar channels in different scenarios.

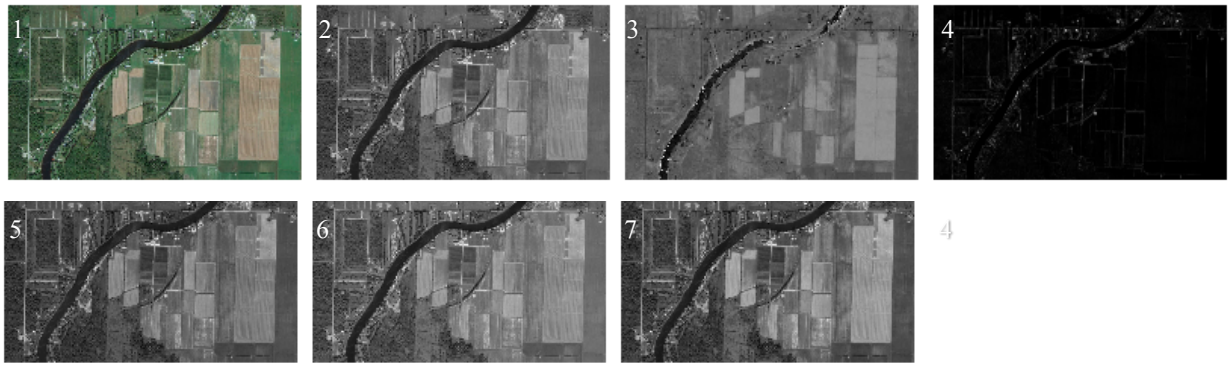

Figure 8 Image 1 channels from 1 to 7, original, gray, hue, variance textures, R (RGB), G (RGB) and B (RGB). Original image courtesy of Google.

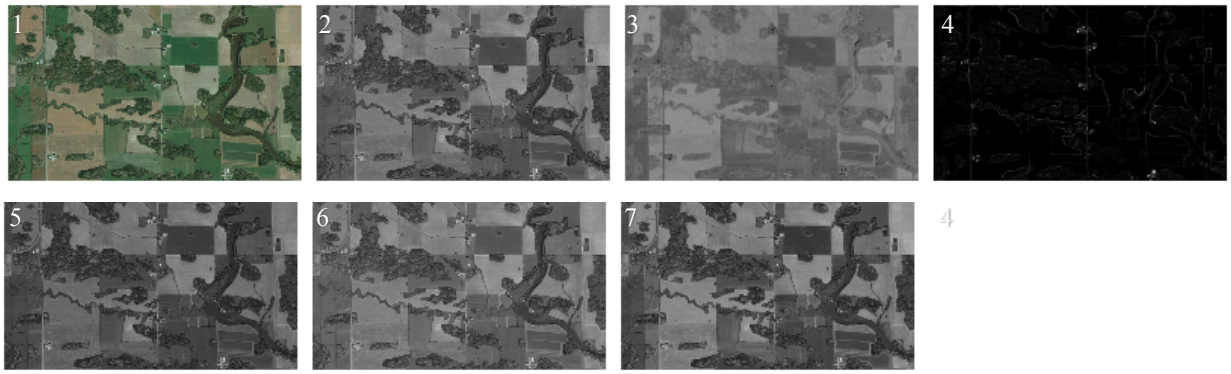

Figure 9 Image 2 channels from 1 to 7 , original, gray, hue, variance textures, R (RGB), G (RGB) and B (RGB). Original image courtesy of Google.

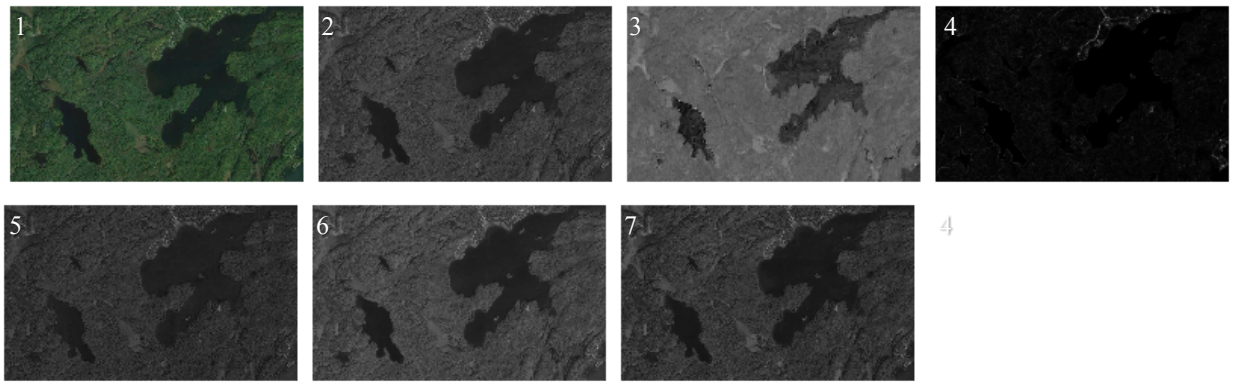

Figure 10 Image 3 channels from 1 to 7 , original, gray, hue, variance textures, R (RGB), G (RGB) and B (RGB). Original image courtesy of Google.

As seen in Figures 6, 7 and 8, gray, R, G and B are similar to each other since they all represent intensities. The differences between them vary according to the image's content. On the other hand, the hue and the variance texture are quite different. To utilize the differences, this experiment uses only three channeling scenarios: gray alone, gray with hue and gray with variance texture. Those channels are for task 1. For task two, I will use only the gray channel alone, since the gray alone is going to be enough to measure the water shield.

4.2 Spatial smoothing filters

After selecting combinations of channels, it is time to implement an important step: the filtering. As mentioned in Chapter 2, this step has a strong effect on the image segmentation results. The filters I use are statistic mode, median blur, Gaussian blur and the LSV, no filtering as well. All filters will be applied using a 3*3 filter box size. Both tasks (task 1 and task 2) are filtered by the same four filters.

Although there might be no clear differences between filtering results visually, Figures 9, 10 and 11 shows that the LSV and median results are slightly lighter (higher values). The next step (segmentation) is expected to show more discrimination between those results.

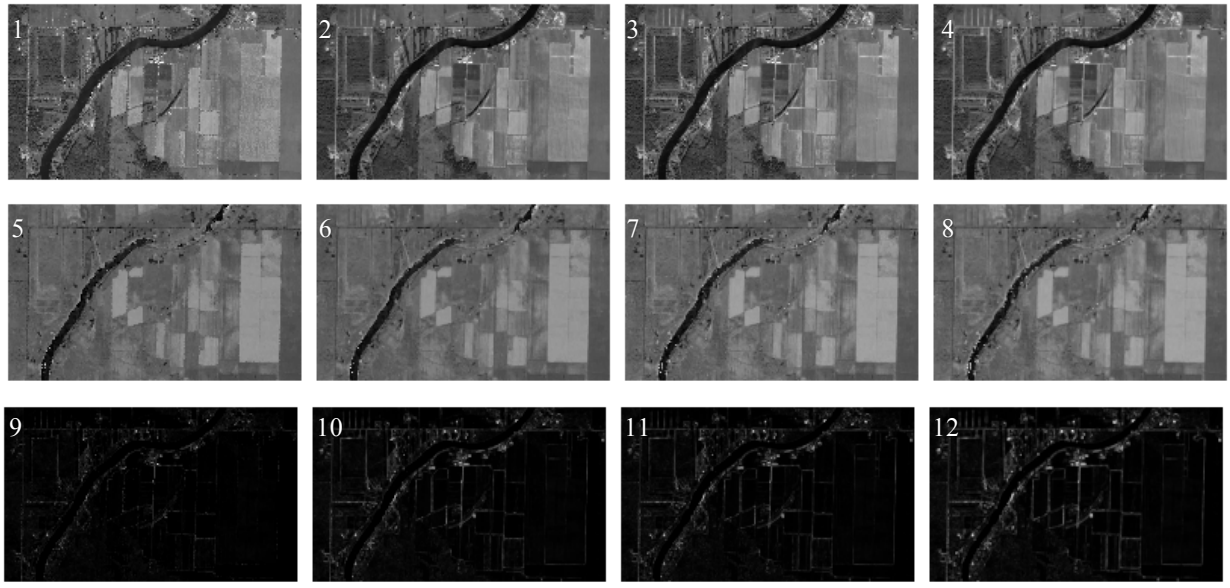

Figure 11 Image one after filtering. From 1 to 12: gray mode, gray median, gray Gaussian, gray LSV, hue mode, hue median, hue Gaussian, hue LSV, textures mode, textures median, textures Gaussian and textures LSV

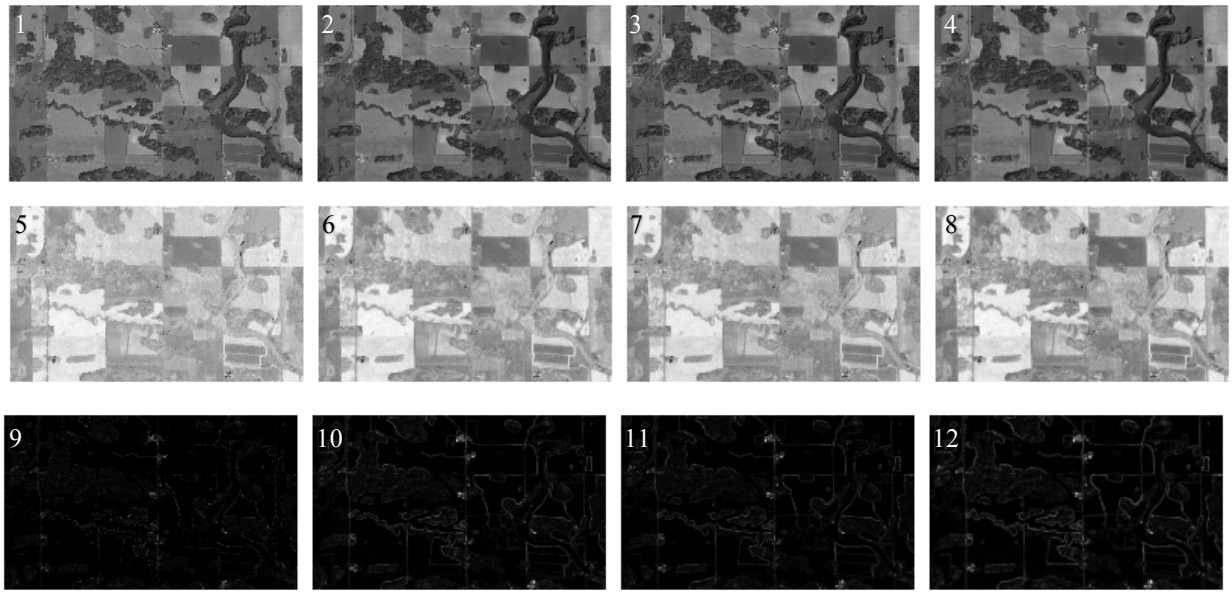

Figure 12 Image two after filtering. From 1 to 12: gray mode, gray median, gray Gaussian, gray LSV, hue mode, hue median, hue Gaussian, hue LSV, textures mode, textures median, textures Gaussian and textures LSV

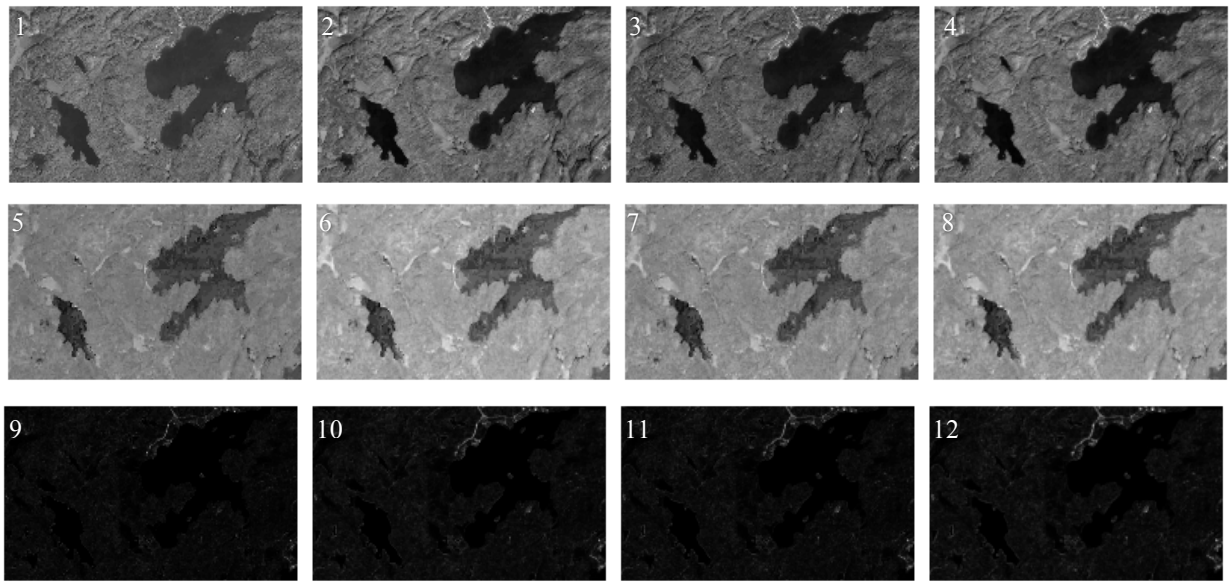

Figure 13 Image three after filtering. From 1 to 12: gray mode, gray median, gray Gaussian, gray LSV, hue mode, hue median, hue Gaussian, hue LSV, textures mode, textures median, textures Gaussian and textures LSV

4.3 Image segmentation - Task 1

After obtaining the filtered results of the sets of channels, it is necessary to start the segmentation, which should address the differences more clearly. The first task includes segmenting the three Google images, which contain three channels and four different filtering techniques (with no filtering as well) for each image. I use two segmentation techniques, k-means classifying and fuzzy c-means classifying with centroids number k=3 and k=5 for each segmentation method.

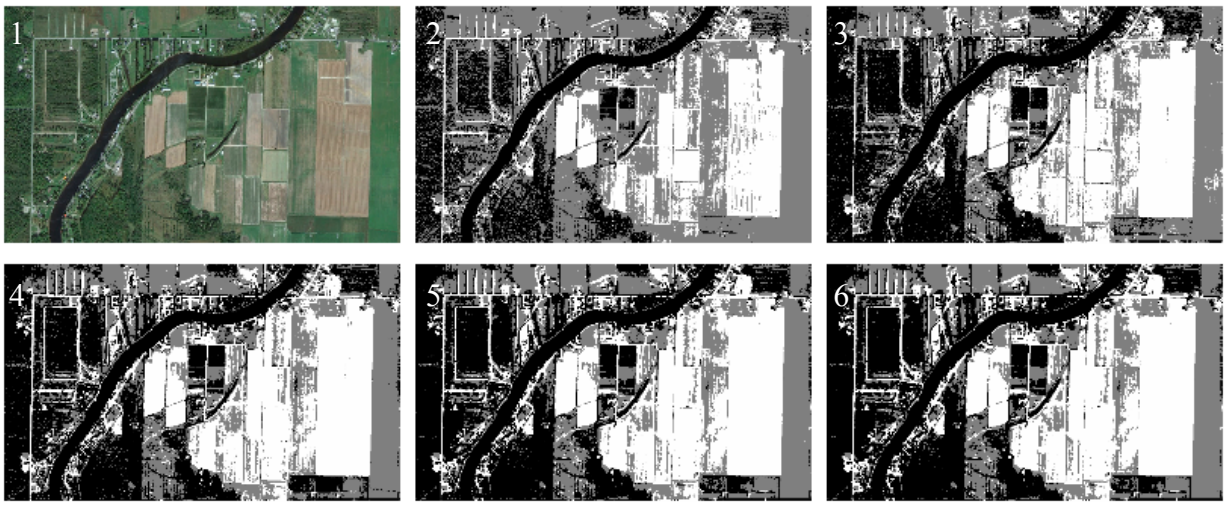

Figure 14 From 1 to 6, First Image 1 (gray by k-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

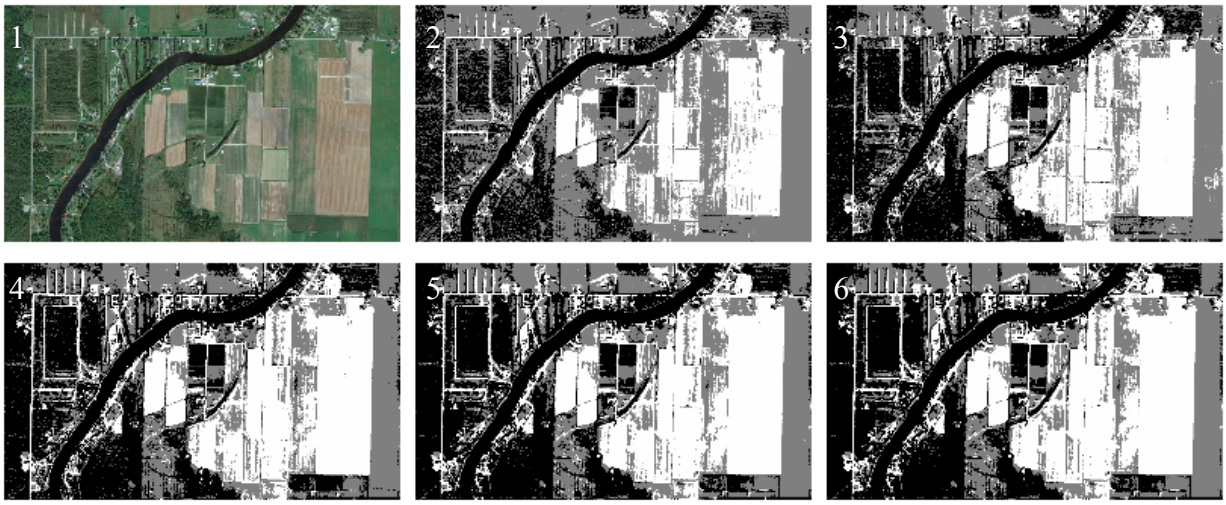

Figure 15 From 1 to 6, First Image1 (gray and hue by k-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

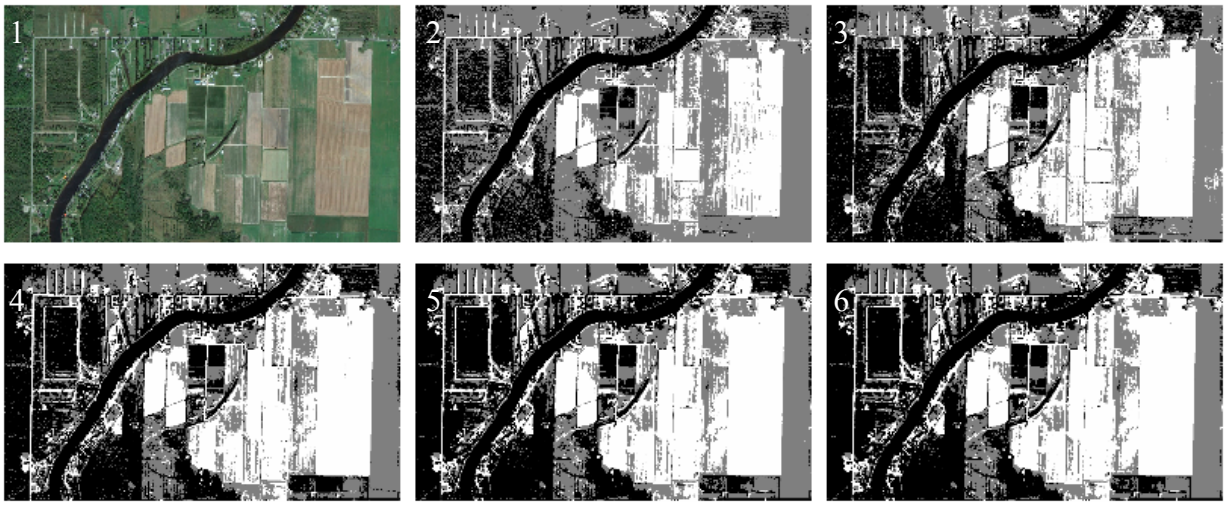

Figure 16 From 1 to 6, First Image 1 (gray and textures by k-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

Figures 12, 13 and 14 show segmentation results for the first image of task 1 using the k-means method. Those figures show the gray channel alone has more noise (small clusters), while the gray with hue channels and gray with variance texture are less noisy. The hue and variance

texture channels show less detail, and involving them leads to more smoothed results. When I have a hue channel for a bush, it will mostly be one colour (green). The bush details are mostly reflected by the intensity. That explains how the hue added to the gray as a segmentation input leads to less noise. Beside that, I can see the variance textures also waiving some noise, but highlighting the main object's borders. The borders between components, when I have the texture channel with the gray, are clearer and highlighted. This is one of the texture analysis features, which is highlighting details.

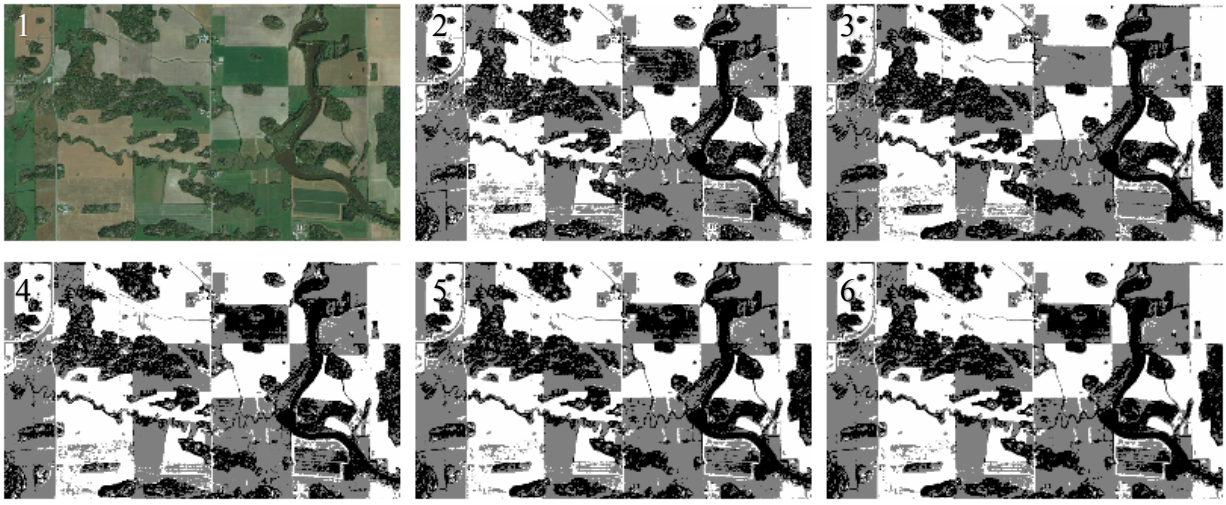

Figure 17 From 1 to 6, Second Image (gray by k-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

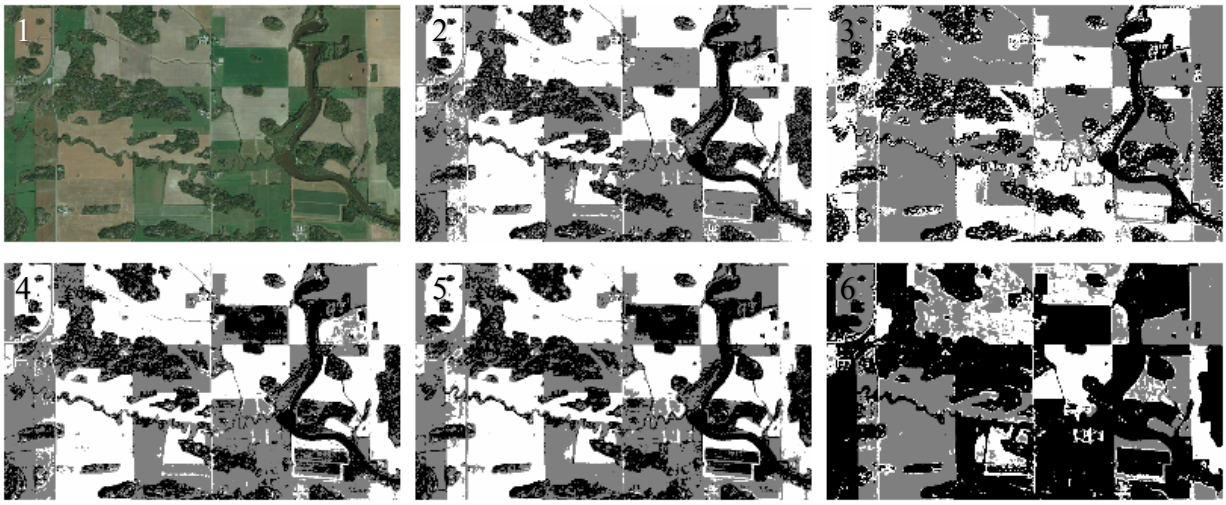

Figure 18 From 1 to 6, Second Image (gray and hue by k-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

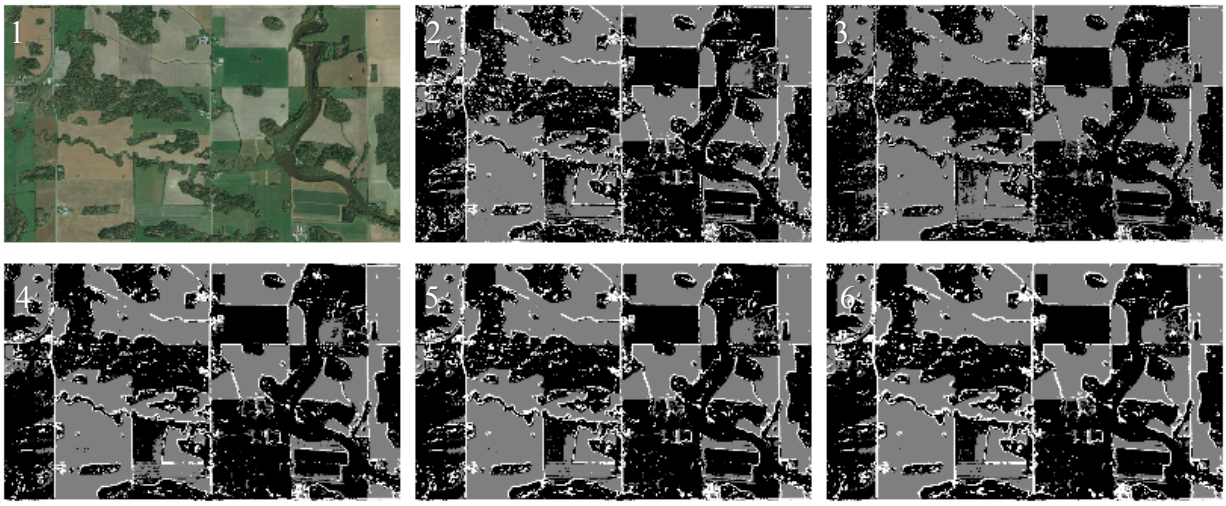

Figure 19 From 1 to 6, Second Image (gray and textures by k-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

Figures 15, 16 and 17 show segmentation results for the second image of task 1 using the k-means method. Those figures give the same notes found in Figures 12, 13 and 14. The hue and

the variance texture channels affect the segmentation results by giving less noise than the gray alone channel.

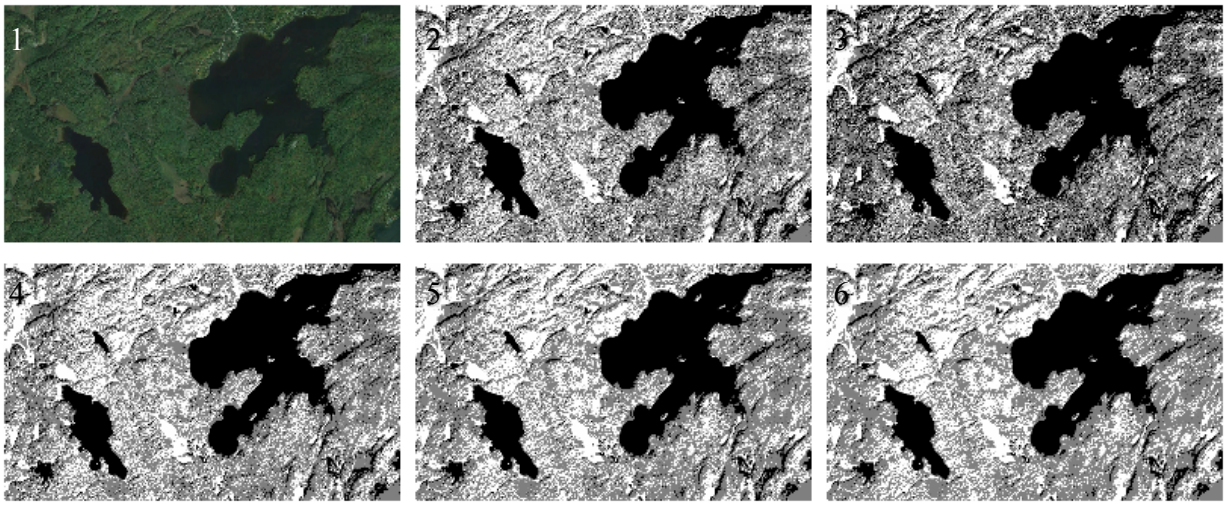

Figure 20 From 1 to 6, Third Image (gray by k-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

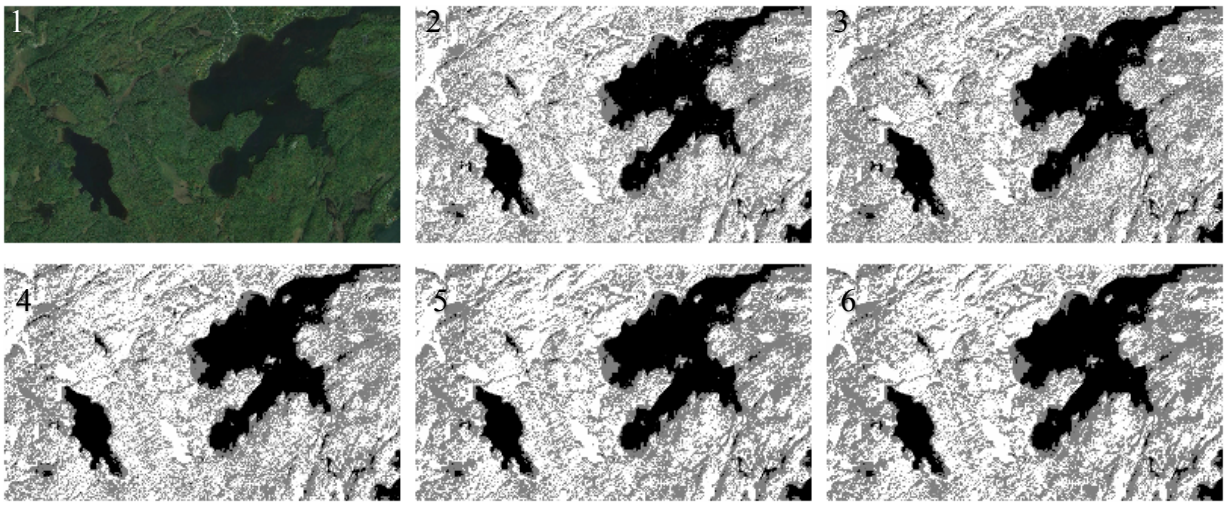

Figure 21 From 1 to 6, Third Image (gray and hue by k-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

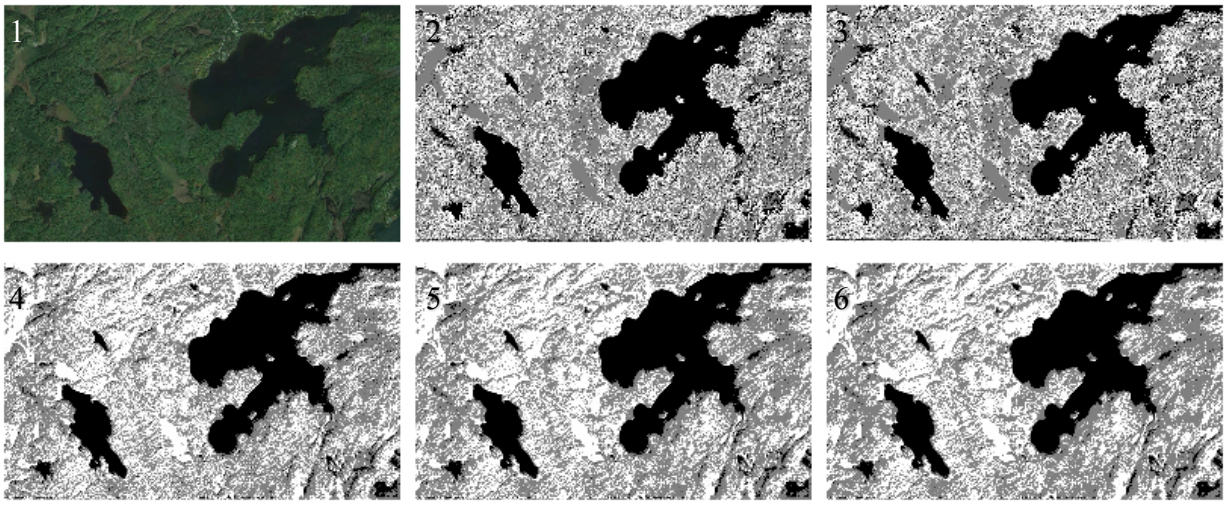

Figure 22 From 1 to 6, Third Image (gray and textures by k-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

Figures 18, 19 and 20 show the segmentation results for the third image of task 1 using the k-means method. As the first and second images, those figures give the same notes found in Figures 12, 13, 14, 15, 16 and 17. The hue and the variance texture channels affected the segmentation results by giving less noise then the gray alone channel, but slightly stronger (less noise). The third image has more natural details since it includes more bush areas.

Over the first, second and third images segmentation using the k-means, it is noticeable that filtering has an effect on segmentation results. Except the statistical mode filter, which shows the least noise reduction, Gaussian blue, median and the LSV show more noise reduction.

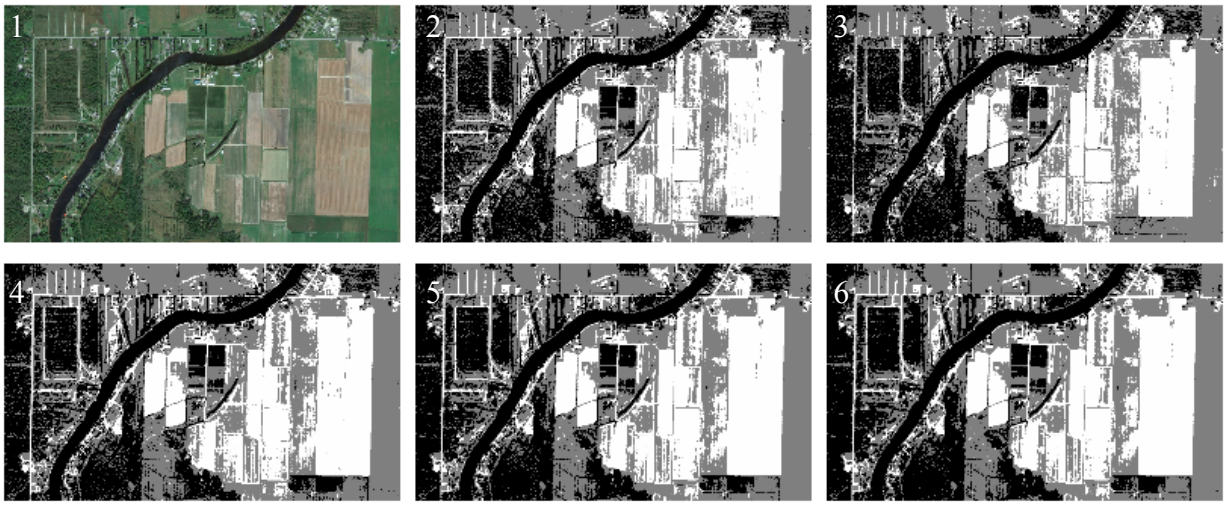

Figure 23 From 1 to 6, First Image (gray by c-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

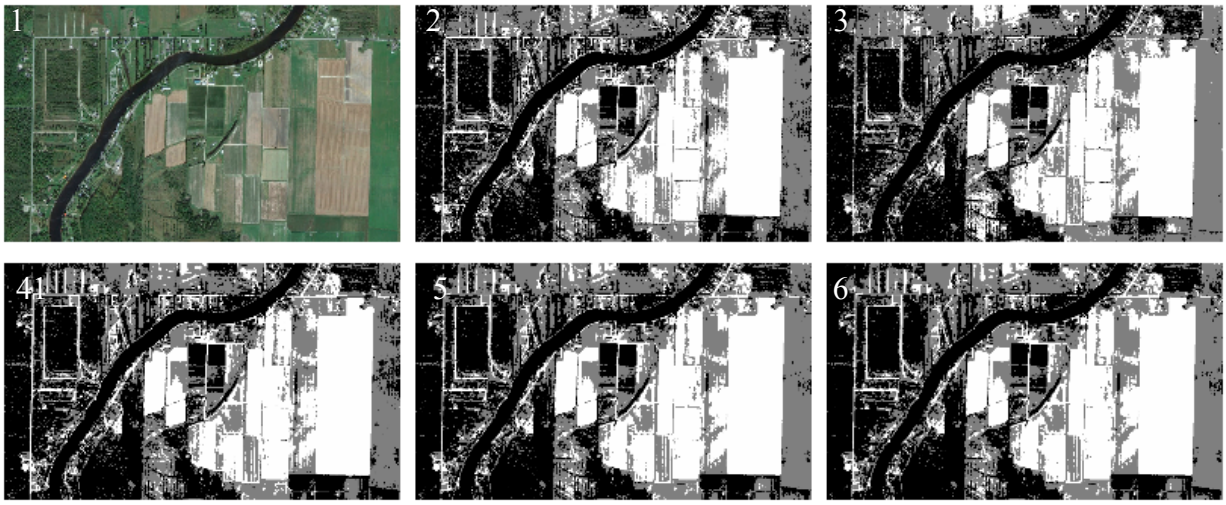

Figure 24 From 1 to 6, First Image (gray and hue by c-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

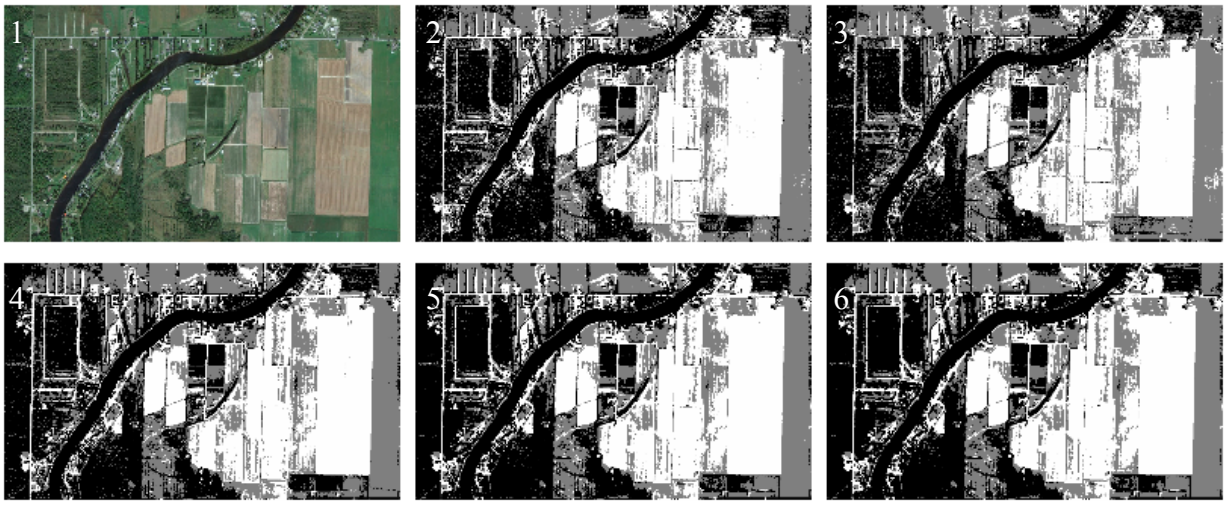

Figure 25 From 1 to 6, First Image (gray and textures by c-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

Figures 21, 22 and 23 show segmentation results for the first image of task 1, but using the fuzzy c-means method. Those results (using the fuzzy c-means) shows more noise than the k-means results. This confirm that the fuzzy c-means is sensitive to the noise. Also, those results that show the gray channel alone has more noise (small clusters), while the gray with hue channels and gray with variance texture are less noisy. The hue and variance texture channels have less detail, and involving them leads to more smoothed results as well.

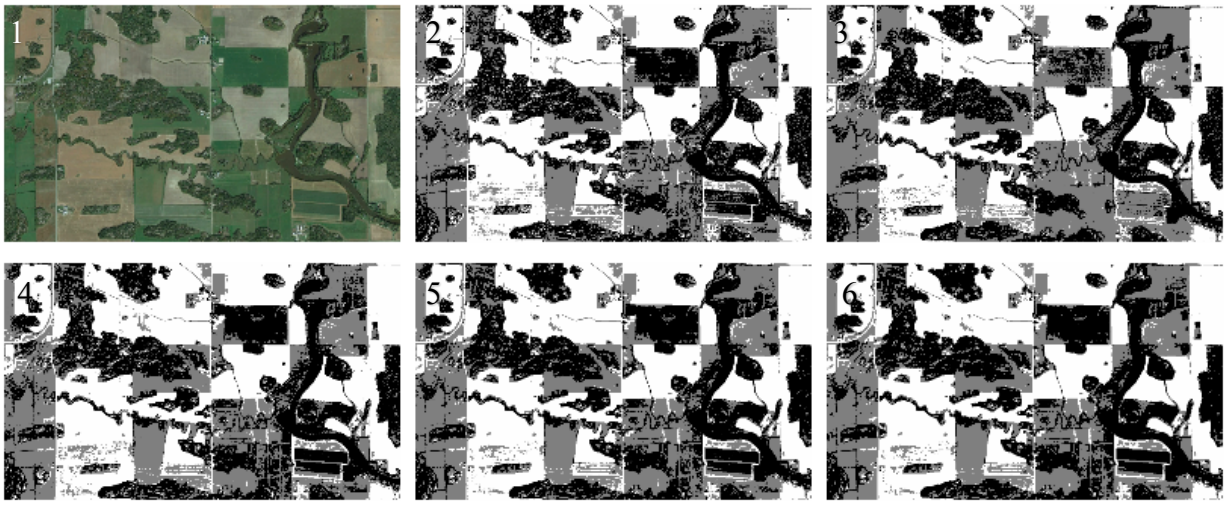

Figure 26 From 1 to 6, Second Image (gray by c-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

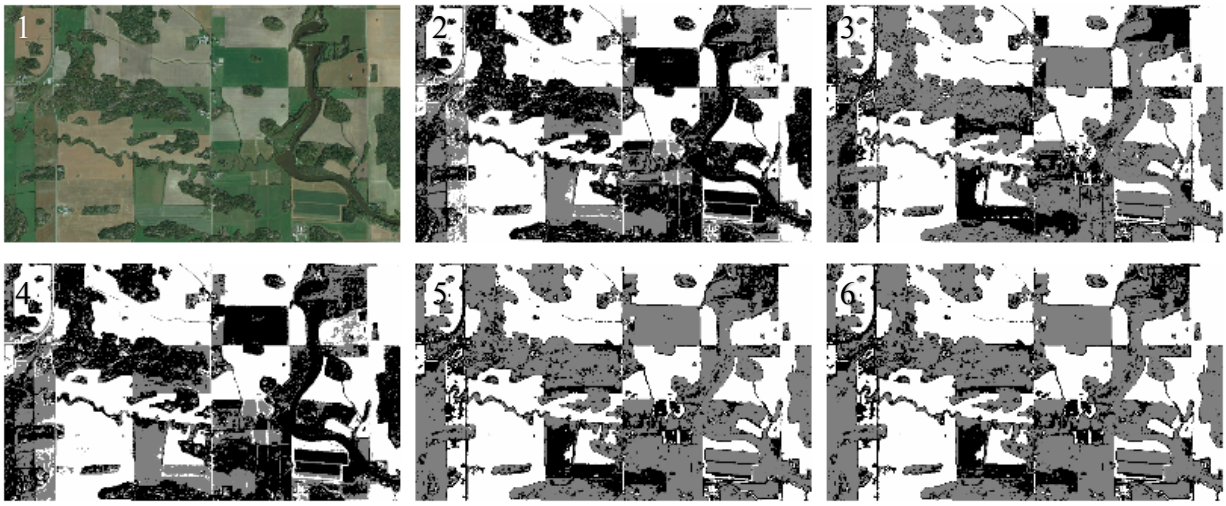

Figure 27 From 1 to 6, Second Image (gray and hue by c-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

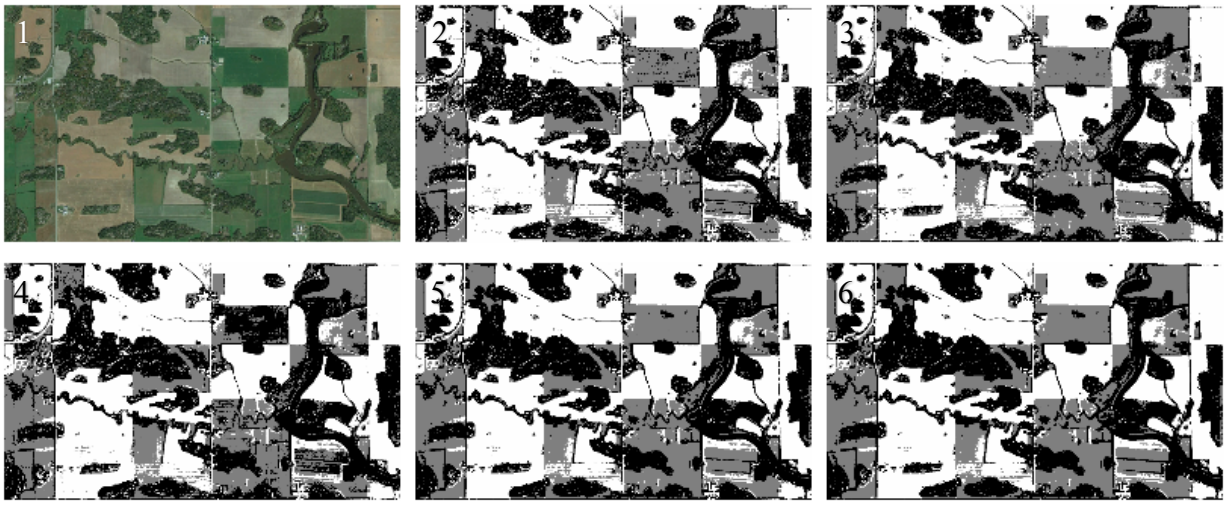

Figure 28 From 1 to 6, Second Image (gray and textures by c-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

Figures 24, 25 and 26 show segmentation results for the second image of task 1 using the fuzzy c-means method. Those results show more noise than the k-means results, as explained before. Also, those results show the gray channel alone has more noise (small clusters), while the gray with hue channels and gray with variance texture are less noise. The hue and variance texture channels are less detailed, and involving them leads to more smoothed results as well.

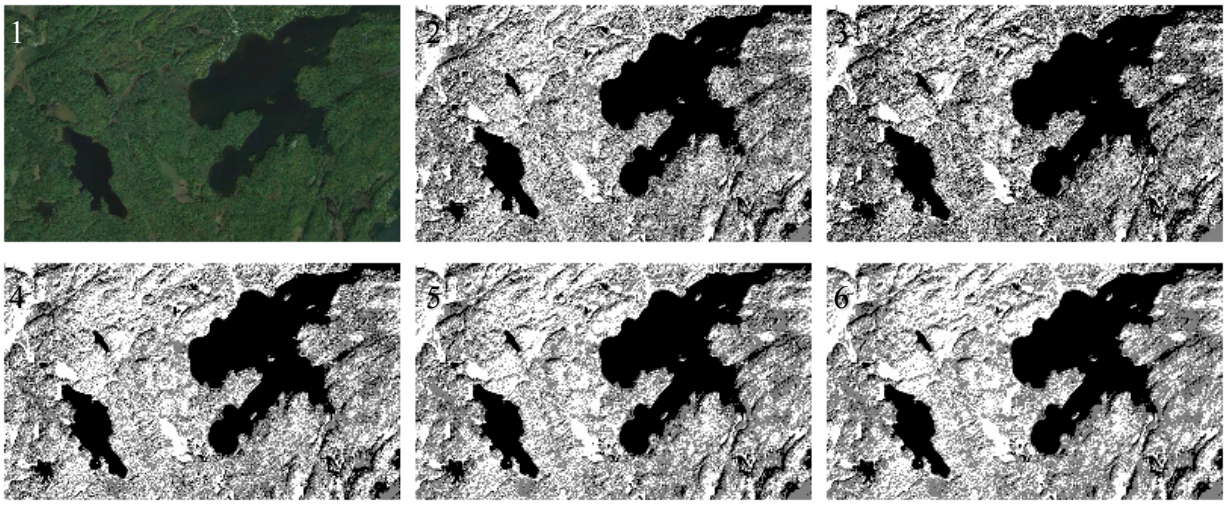

Figure 29 From 1 to 6, Third Image (gray by c-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

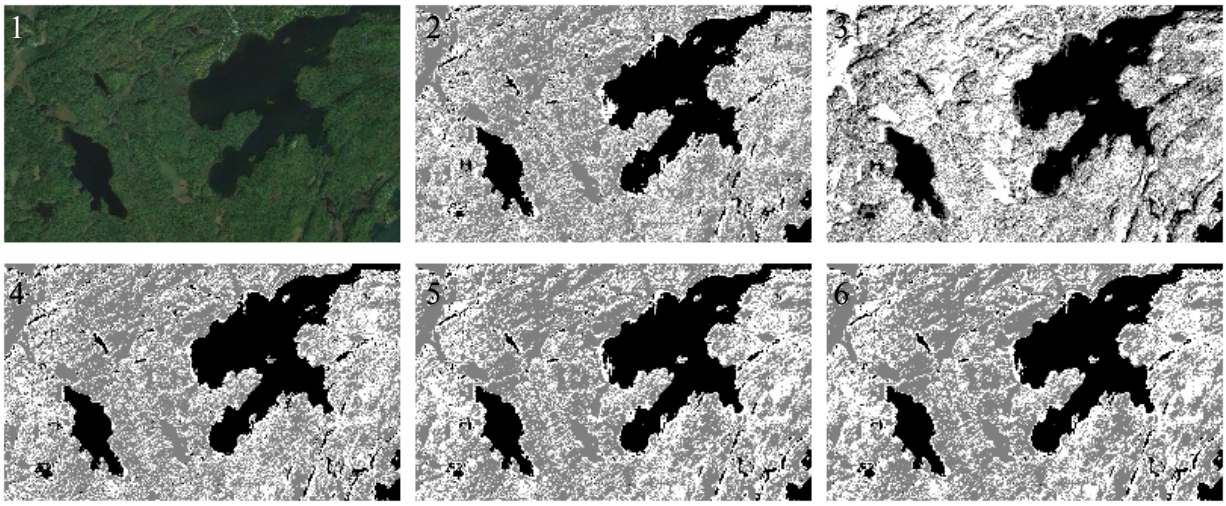

Figure 30 From 1 to 6, Third Image (gray and hue by c-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

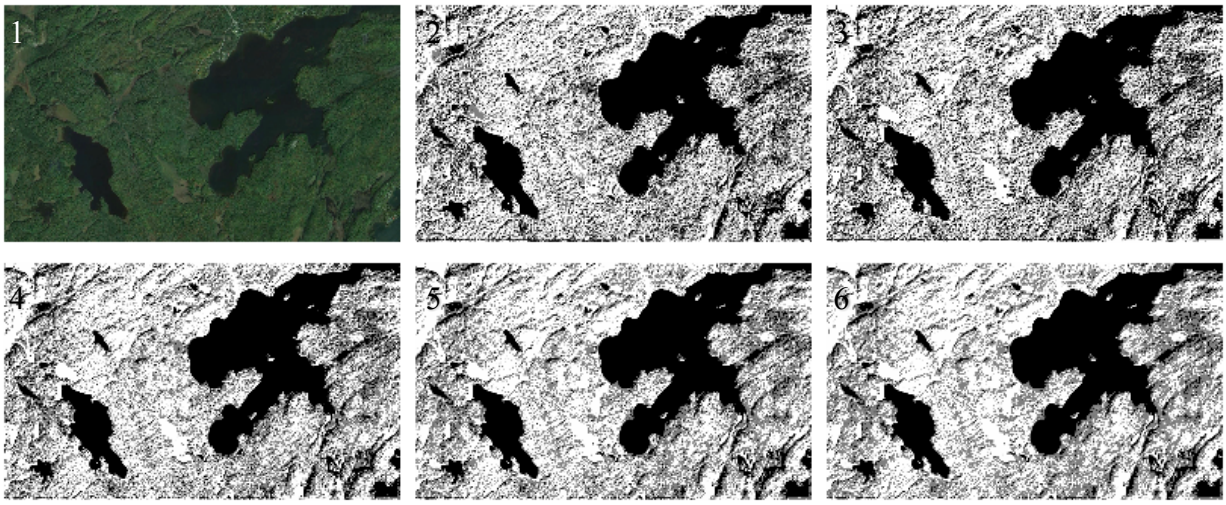

Figure 31 From 1 to 6, Third Image (gray and textures by c-means with k of 3): Original, no filter, mode, median, Gaussian and LSV. Original image courtesy of Google.

Figures 27, 28 and 29 show segmentation results for the third image of task 1 using the fuzzy c-means method. The first and second image results show more noise than the k-means results, which has been explained before. Also, those results show that the gray channel alone has more noise (small clusters), while the gray with hue channels and gray with variance texture are less noisy. The hue and variance texture channels are less detailed, and involving them leads to more smoothed results.

Like the k-means segmentation results, over the first, second and third images segmentation using the c-means, it is noticeable that filtering has an effect on segmentation results. Except the statistical mode filter, which shows the least noise reduction, Gaussian blue, median and the LSV show more noise reduction as well.

4.4 Image segmentation - Task 2

The second task is not including any noise reduction measuring. As mentioned before, my objective is to do a quantitative measure the surface area over time using different filtering. This task includes segmenting the USGS two set of images using the same two segmentation techniques, k-means classifying and fuzzy c-means classifying, but with centroids number k=3 only. Also, this task is done using the gray level channel alone. Since task 2 is focusing in one object, which is the water shield, centroids number k=3 and gray channel are suitable.

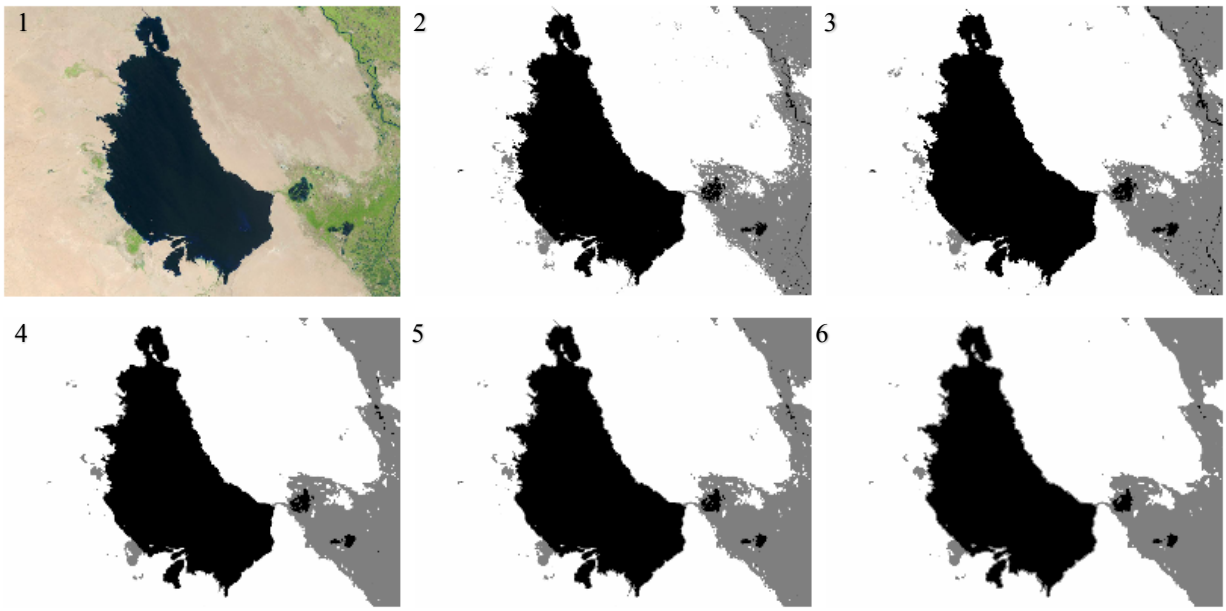

Figure 32 Bahr Al-milh first image segmentation results using k-means. From 1 to 6, original, no filter, mode, median, Gaussian and LSV. Original image courtesy of U.S. Geological Survey.

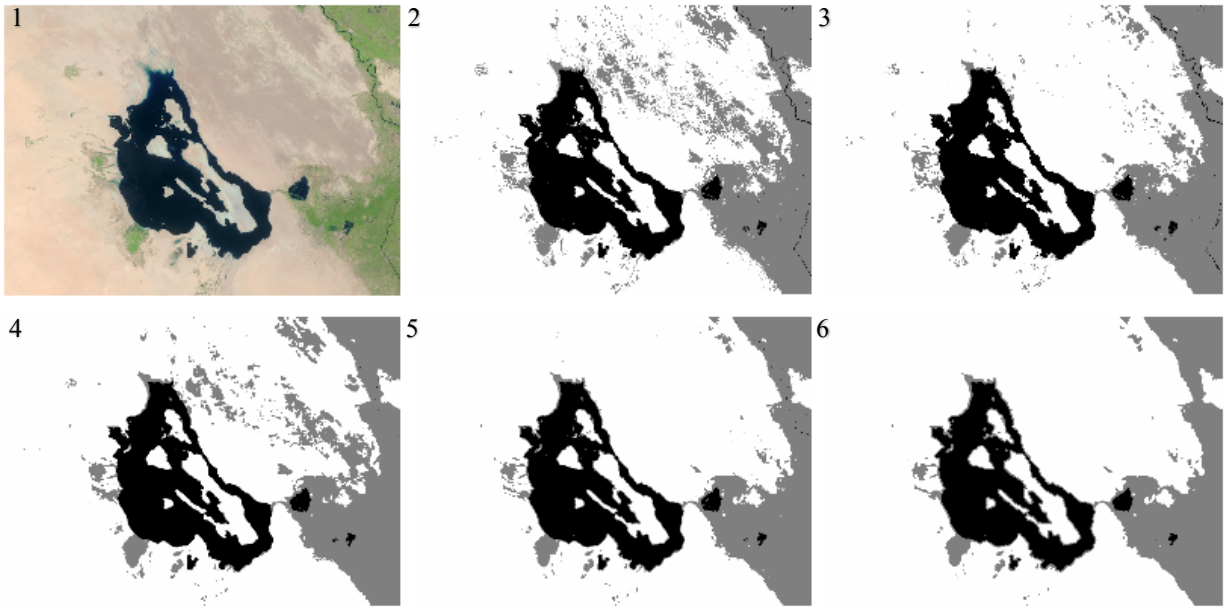

Figure 33 Bahr Al-milh second image segmentation results using k-means. From 1 to 6, original, no filter, mode, median, Gaussian and LSV. Original image courtesy of U.S. Geological Survey.

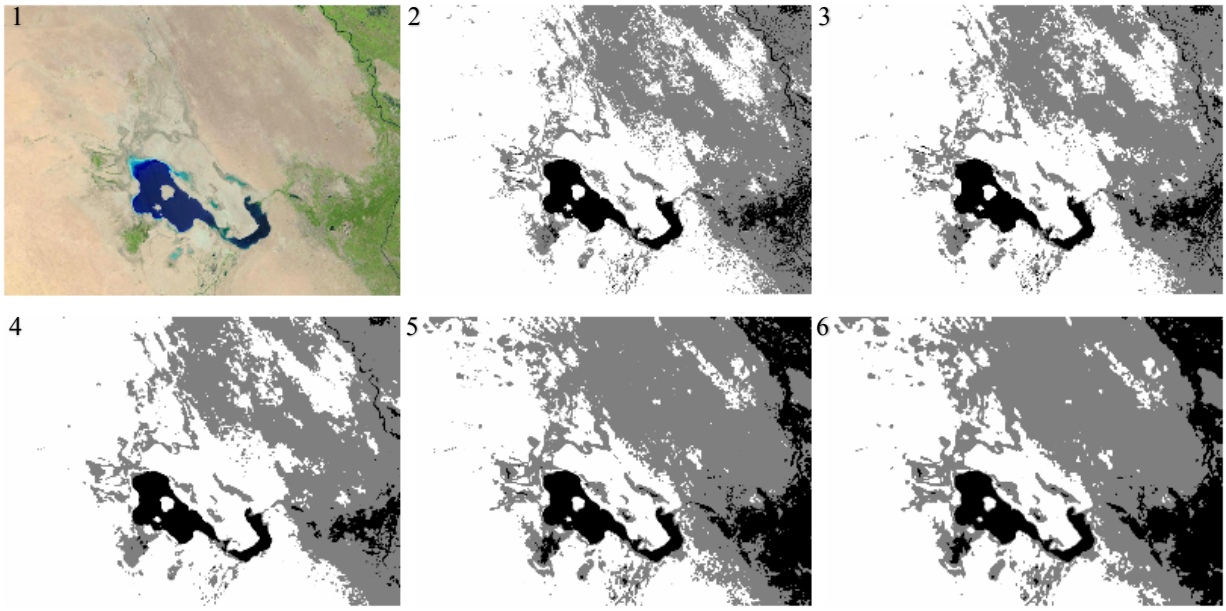

Figure 34 Bahr Al-milh third image segmentation results using k-means. From 1 to 6, original, no filter, mode, median, Gaussian and LSV. Original image courtesy of U.S. Geological Survey.

Figures 30, 31, and 32 show a satisfying k-means segmentation as the water shield is in one cluster over all the filters. Those figures show no difference between filters in the water shield (at least visually).

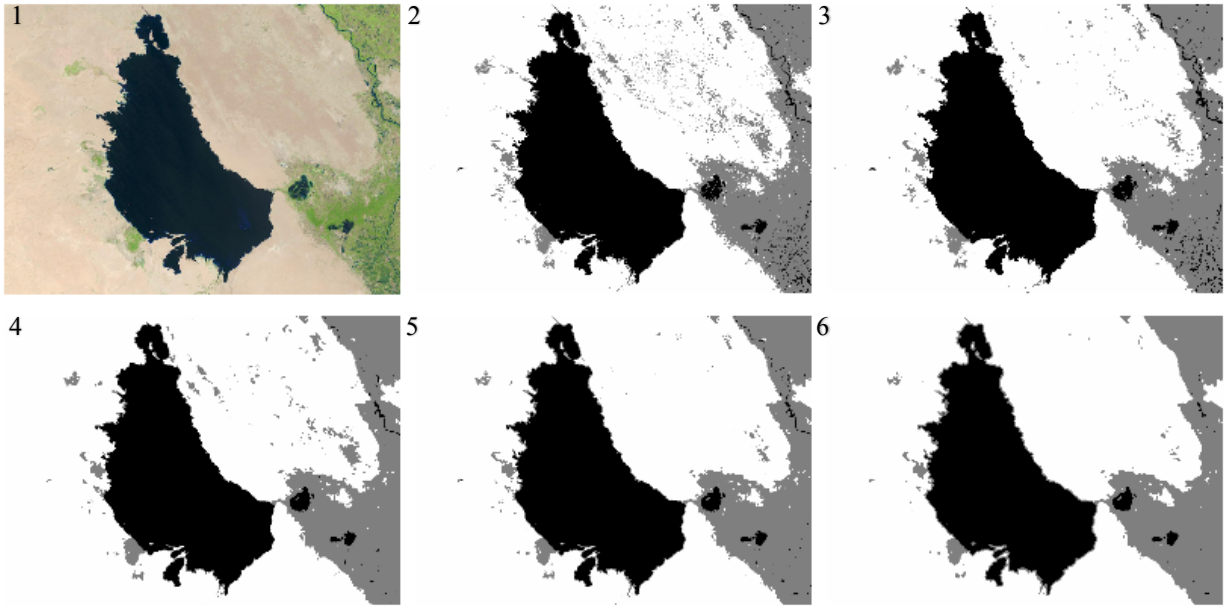

Figure 35 Bahr Al-milh first image segmentation results using c-means. From 1 to 6, original, no filter, mode, median, Gaussian and LSV. Original image courtesy of U.S. Geological Survey.

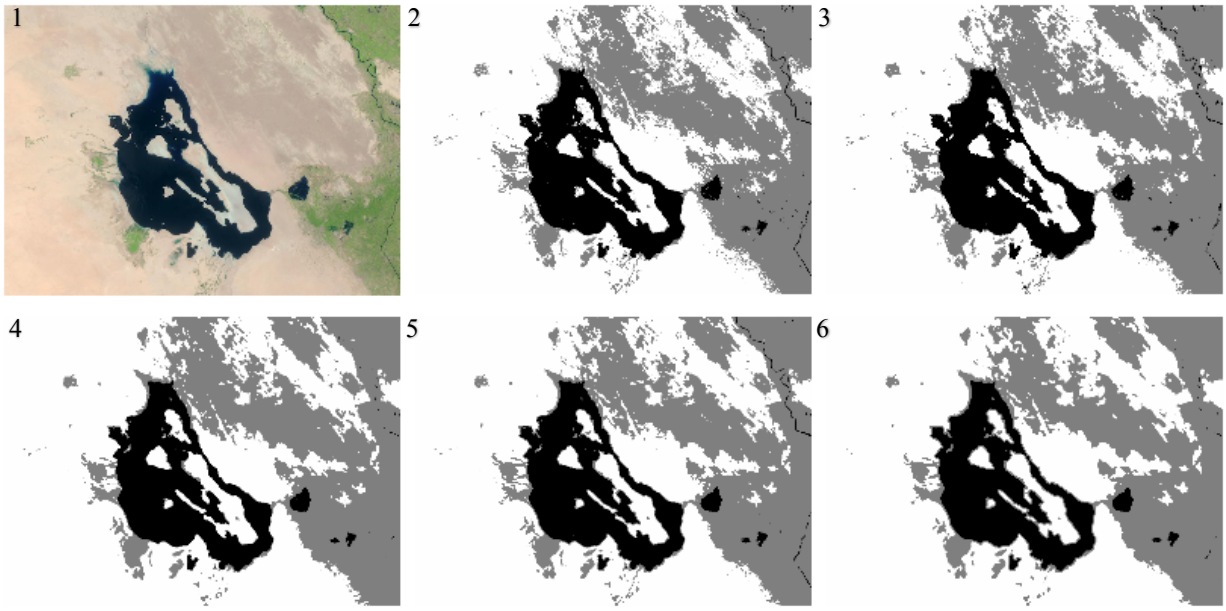

Figure 36 Bahr Al-milh second image segmentation results using c-means. From 1 to 6, original, no filter, mode, median, Gaussian and LSV. Original image courtesy of U.S. Geological Survey.

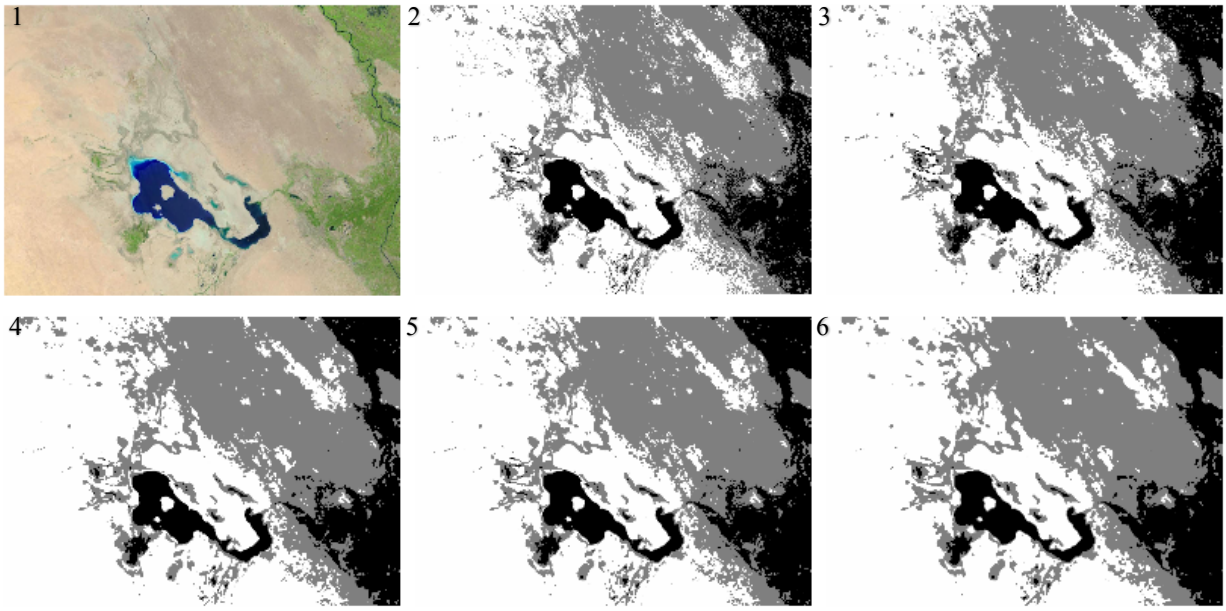

Figure 37 Bahr Al-milh third image segmentation results using c-means. From 1 to 6, original, no filter, mode, median, Gaussian and LSV. Original image courtesy of U.S. Geological Survey.

Similar to Figures 30, 31 and 32, Figures 33, 34 and 35 show a satisfying c-means segmentation as the water shield is in one cluster over all the filters. Those figures show no difference in the water shield between filters (at least visually).

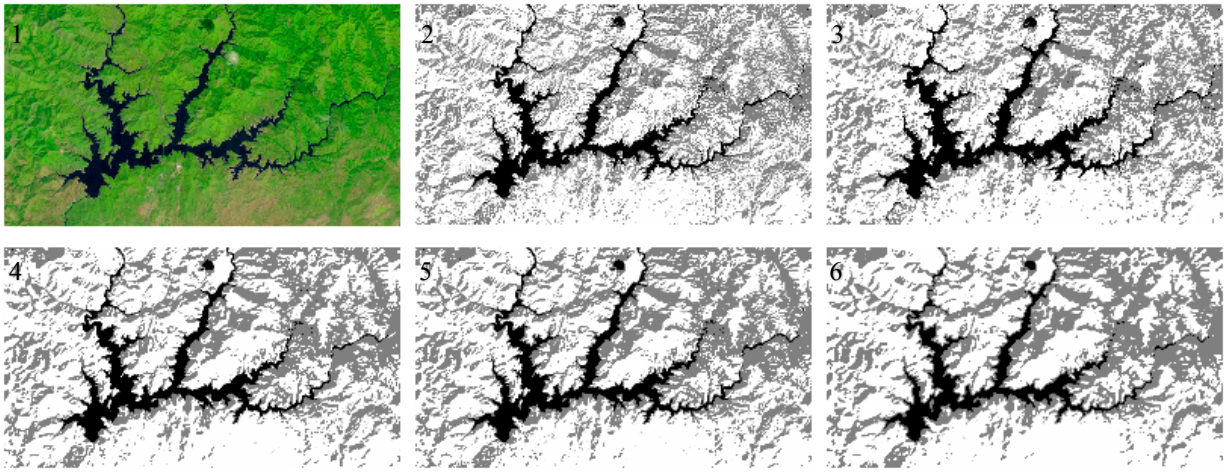

Figure 38 Shasta Lake first image segmentation results using k-means. From 1 to 6, original, no filter, mode, median, Gaussian and LSV. Original image courtesy of U.S. Geological Survey.

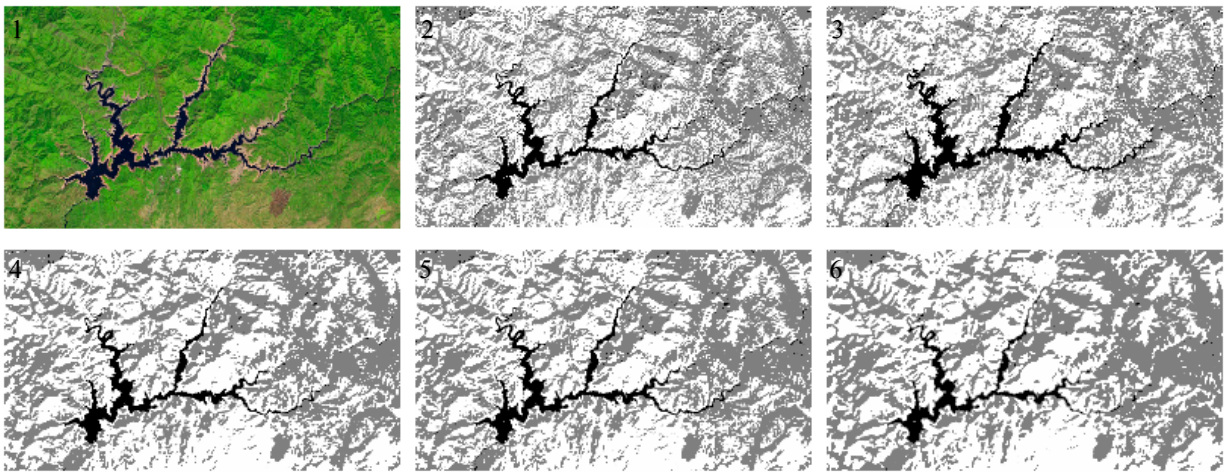

Figure 39 Shasta Lake second image segmentation results using k-means. From 1 to 6, original, no filter, mode, median, Gaussian and LSV. Original image courtesy of U.S. Geological Survey.

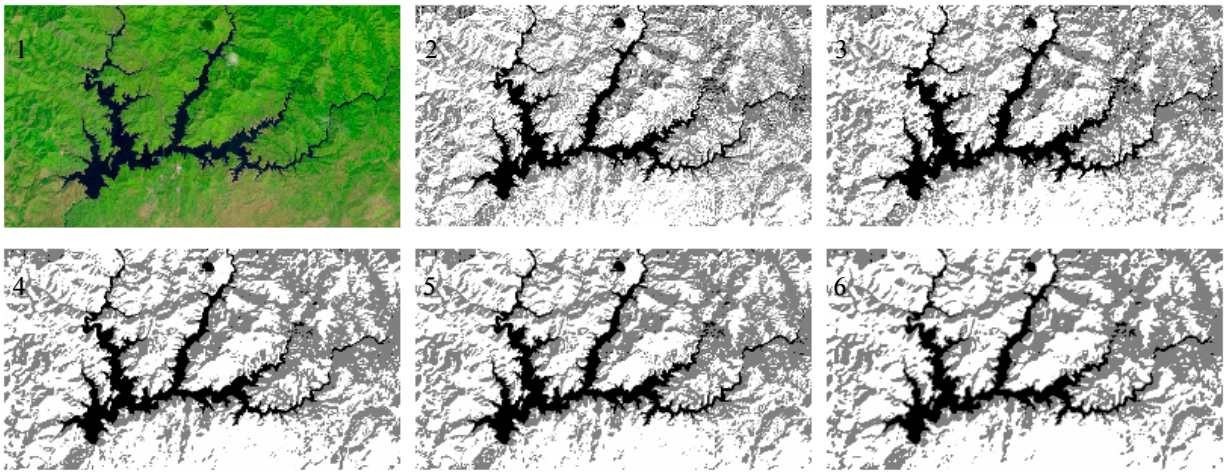

Figure 40 Shasta Lake first image segmentation results using c-means. From 1 to 6, original, no filter, mode, median, Gaussian and LSV. Original image courtesy of U.S. Geological Survey.

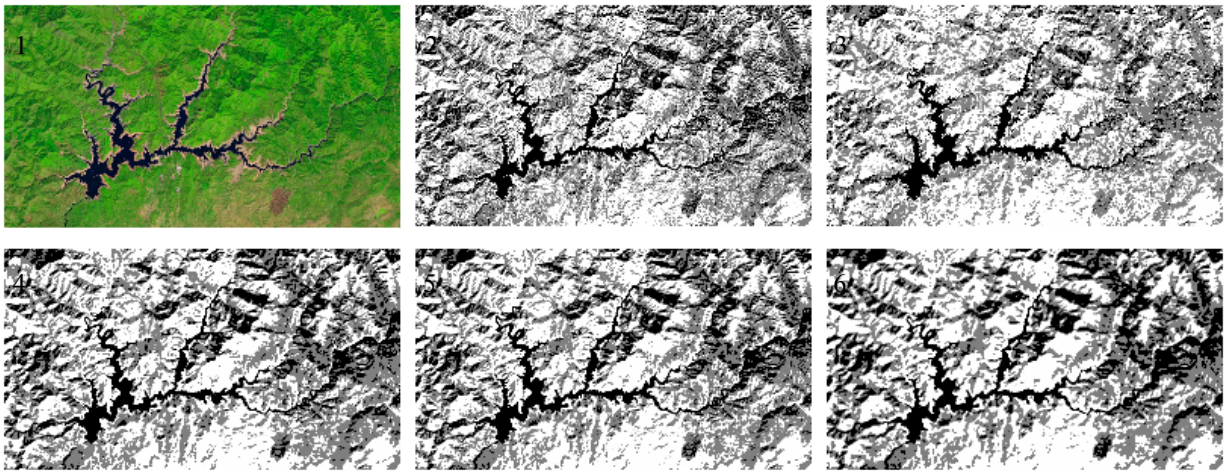

Figure 41 Shasta Lake second image segmentation results using c-means. From 1 to 6, original, no filter, mode, median, Gaussian and LSV. Original image courtesy of U.S. Geological Survey.

Different from Bahr al-milh images, Shasta Lake shows noticeable effects for the filtering. Shasta Lake images have a line like water shields, which been erased in some filters such as median and LSV. Fuzzy c-means segmentation shows more filtering effects, which means more loss of narrow water shield.

Since this task (task 2) is a supervised task (select area of interest manually), there is a chance for error. Also, Shasta Lake shows that the whole water shield could be disconnected in some segmentations. To avoid errors in measuring the changes over time, I will do multiple selections for multiple areas of interest and add all selected areas results together using the bitwise OR operation. Also, I will list the selections results to ensure the selection is working well.

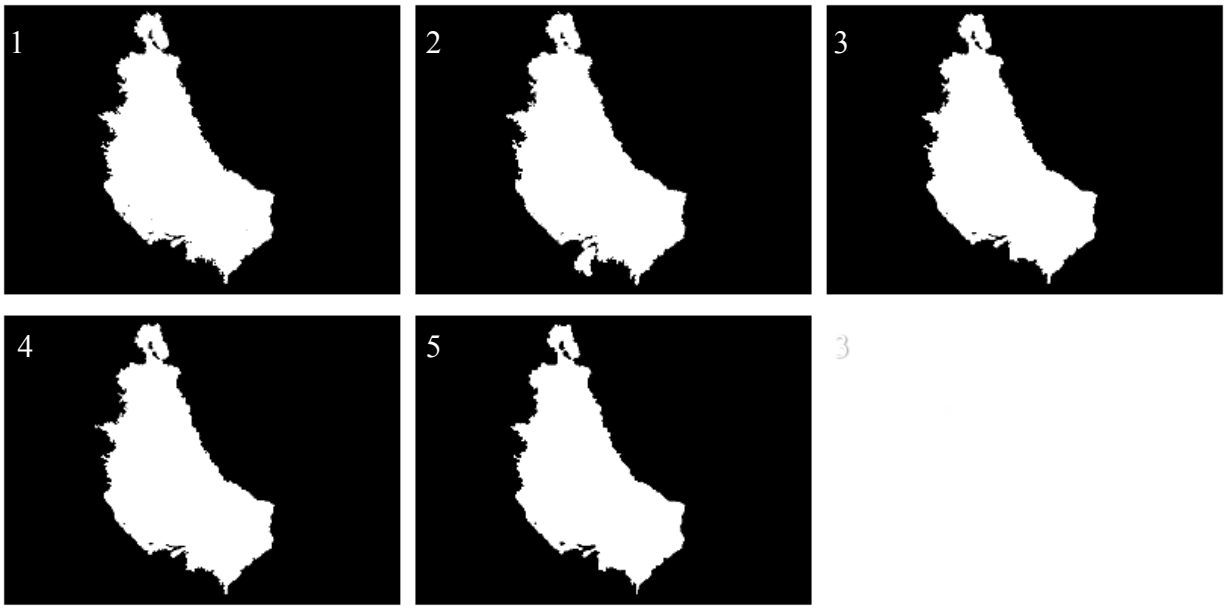

Figure 42 Bahr Al-milh first image area of interest selection using k-means results. From 1 to 5: no filter, mode, median, Gaussian and LSV.

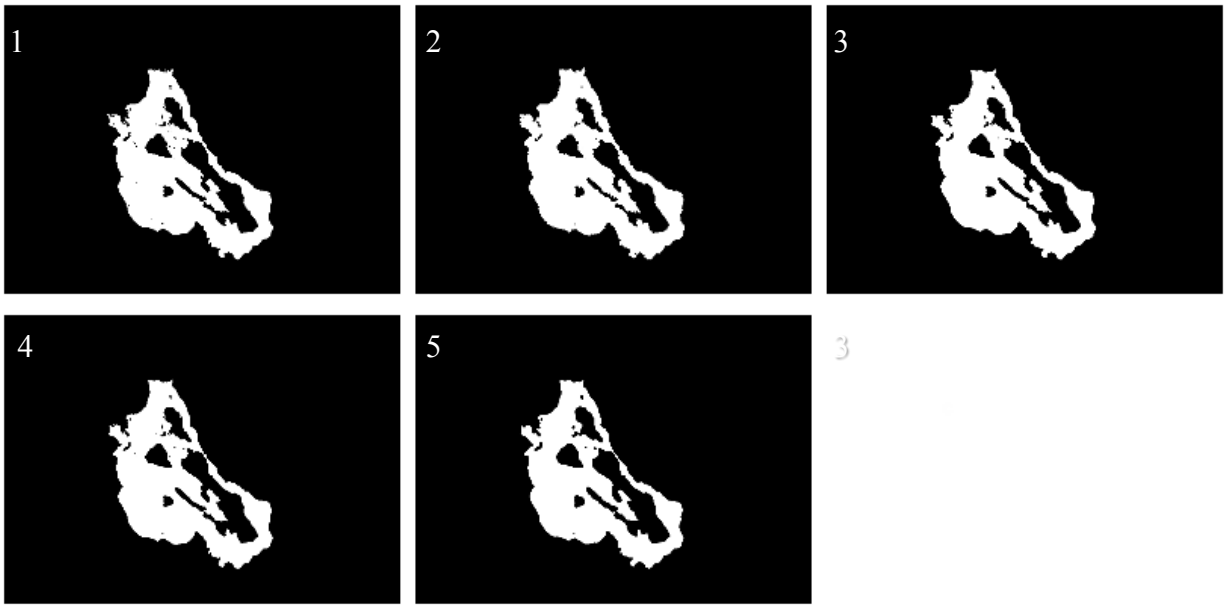

Figure 43 Bahr Al-milh second image area of interest selection using k-means results. From 1 to 5: no filter, mode, median, Gaussian and LSV.

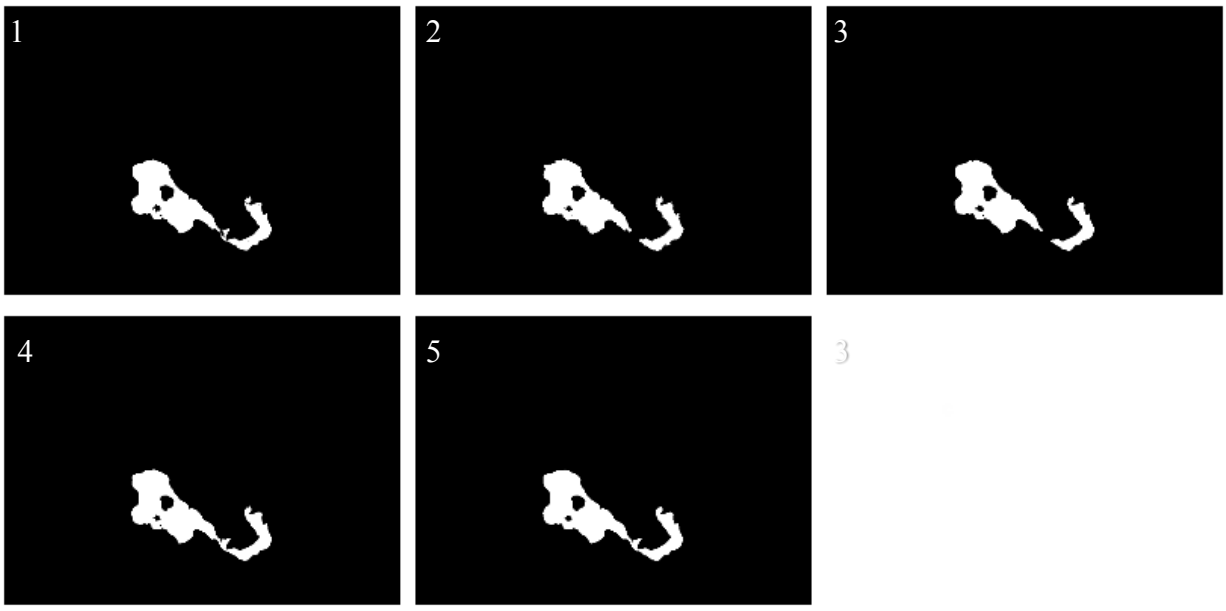

Figure 44 Bahr Al-milh third image area of interest selection using k-means results. From 1 to 5: no filter, mode, median, Gaussian and LSV.

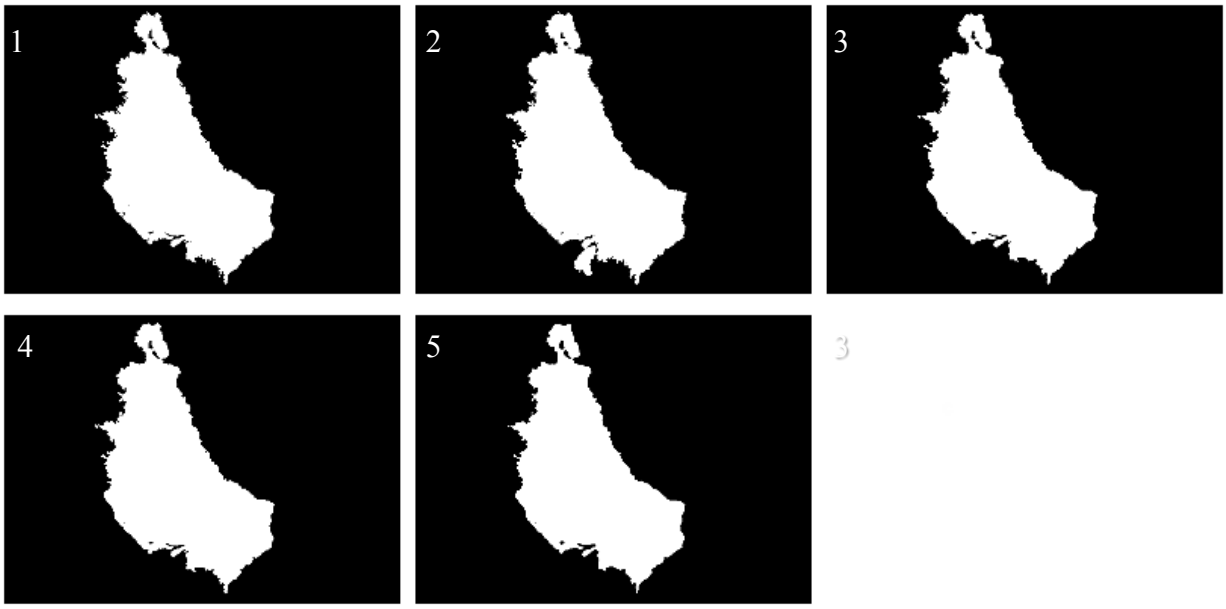

Figure 45 Bahr Al-milh first image area of interest selection using c-means results. From 1 to 5: no filter, mode, median, Gaussian and LSV.

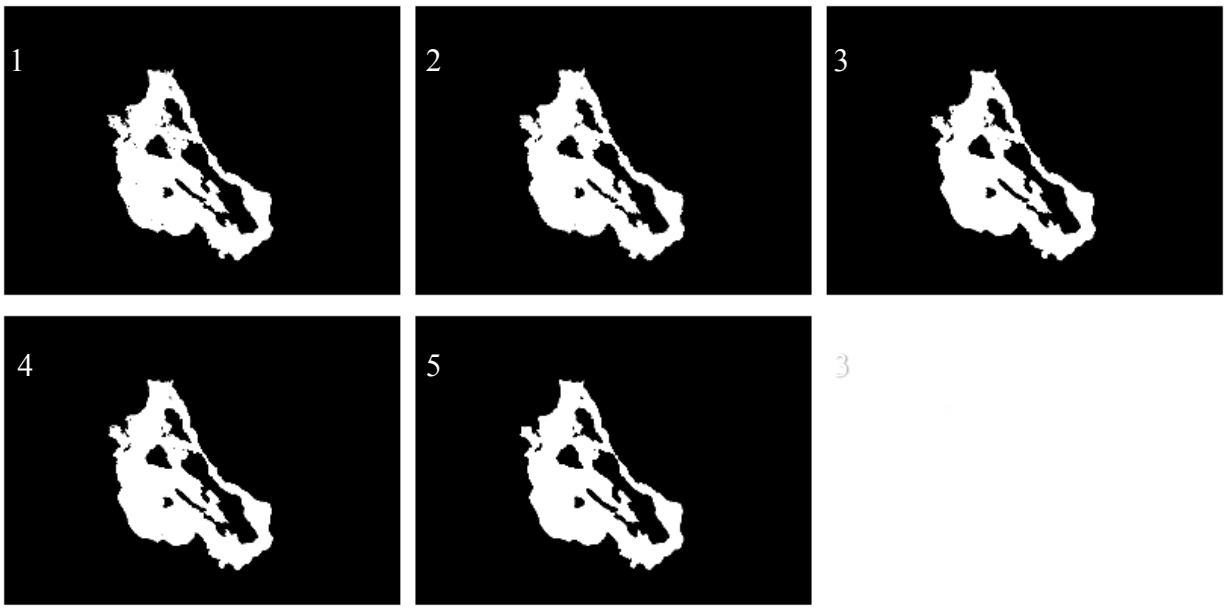

Figure 46 Bahr Al-milh second image area of interest selection using c-means results. From 1 to 5: no filter, mode, median, Gaussian and LSV.

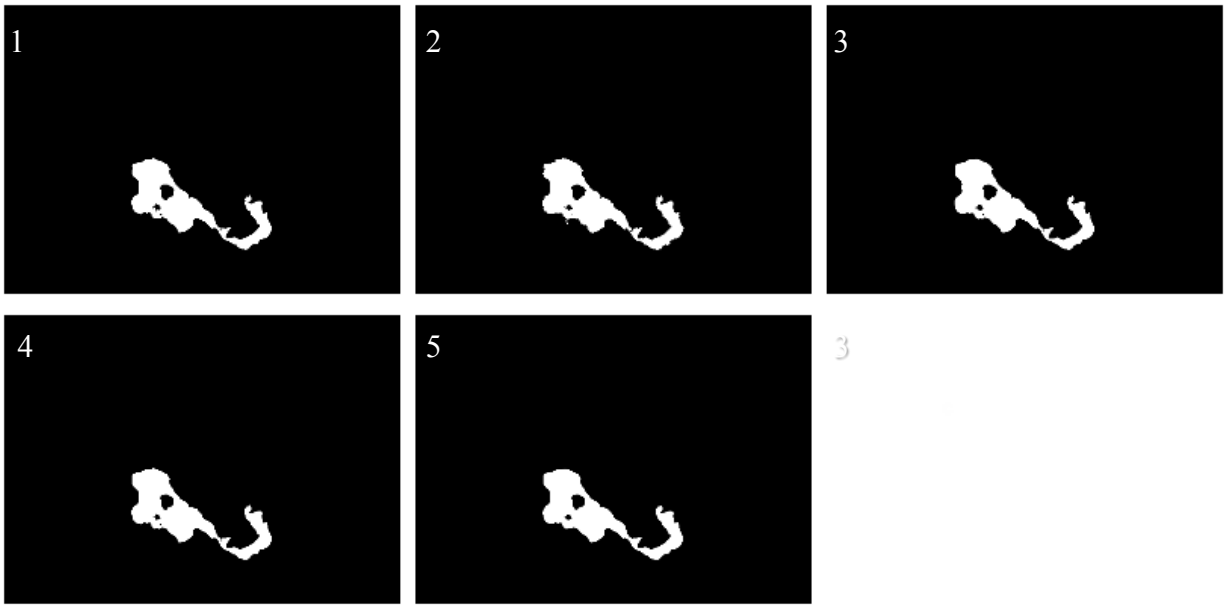

Figure 47 Bahr Al-milh third image area of interest selection using c-means results. From 1 to 5: no filter, mode, median, Gaussian and LSV.

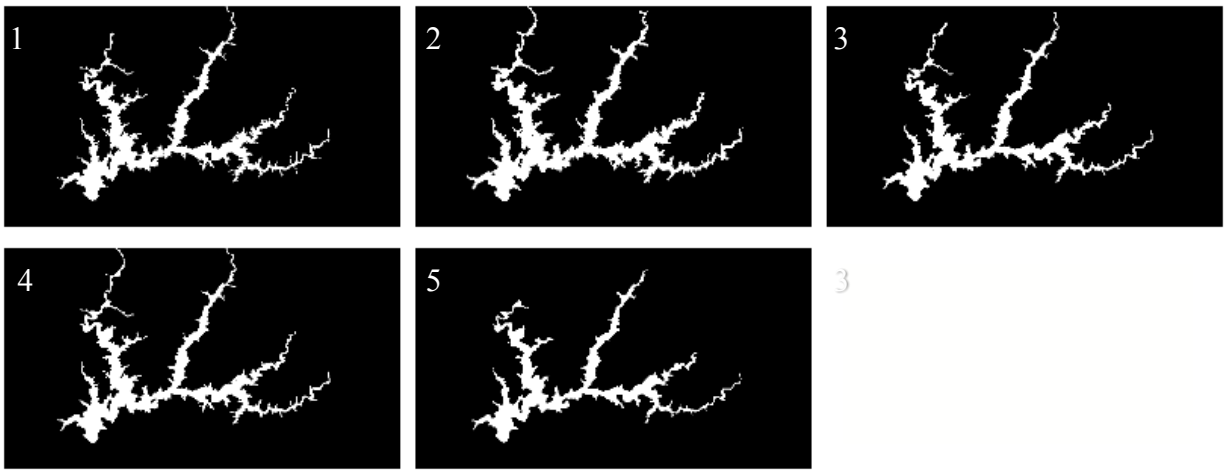

Figure 48 Shasta lake first image area of interest selection using k-means results. From 1 to 5: no filter, mode, median, Gaussian and LSV.

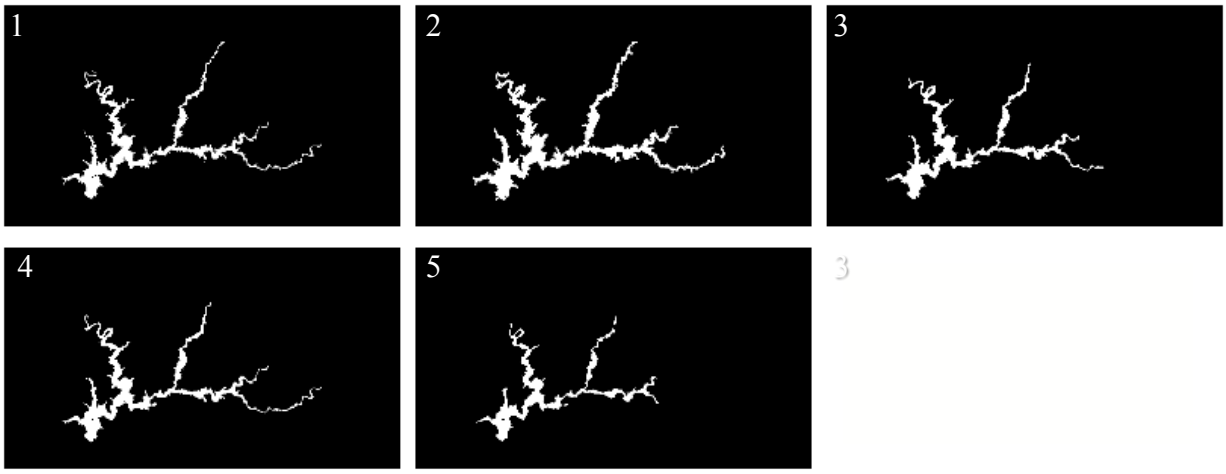

Figure 49 Shasta lake second image area of interest selection using k-means results. From 1 to 5: no filter, mode, median, Gaussian and LSV.

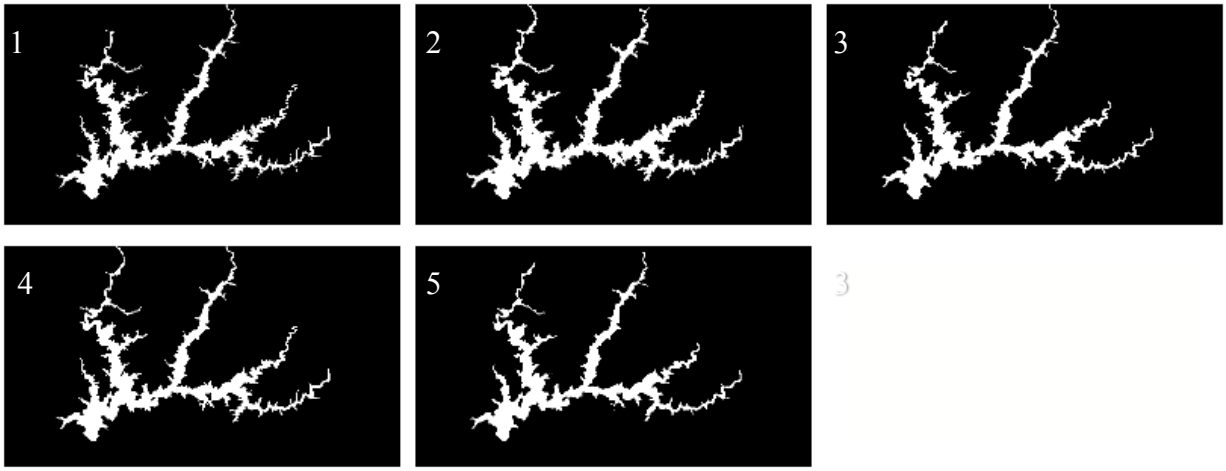

Figure 50 Shasta lake first image area of interest selection using c-means results. From 1 to 5: no filter, mode, median, Gaussian and LSV.

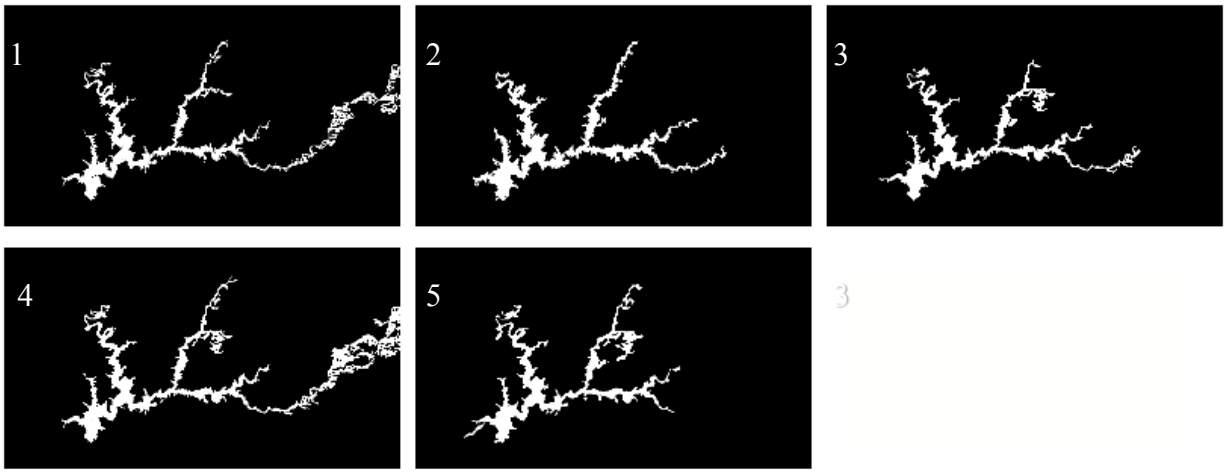

Figure 51 Shasta lake second image area of interest selection using c-means results. From 1 to 5: no filter, mode, median, Gaussian and LSV.

Figures 40 to 49 show the areas of interest (the water shield) is undamaged and ready for the next step. There are few differences, which are due to the filtering effects. The following analysis clarifies those differences.

4.5 Analysis - Task 1

The previous section visualizes the segmentation results that may not be very clear. In this section I am going to use the analysis methods explained Chapter 3. For task 1 (Google images), I use the size count method to produce a qualitative analysis of the cluster sizes and counts. Less small clusters (noise) are considered more optimized and means more noise reduction been achieved.

The following tables list the number of clusters occurring in a segmentation results divided into size groups. The sizes start from less then 8-pixel clusters and the sizes increases four times until 600000 pixels (the maximum possible for the image size). In each table, the letter g refers to gray

channel, g&h refers to gray with hue channels and g&t refers to gray and variance texture. Also, there is a chart of the size counts as well.

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
1	g	3	kmeans	noFilter	1213	464	73	9	1	0	0	1	2	0
1	g	3	kmeans	mode	911	309	42	3	0	1	0	1	2	0
1	g	3	kmeans	median	817	394	74	6	2	0	0	1	2	0
1	g	3	kmeans	gaussian	487	220	48	5	2	1	0	1	2	0
1	g	3	kmeans	LSV	349	184	41	4	2	0	0	1	2	0

Table 1 Sizes counts for Image 1, channel gray and k3 using k-means

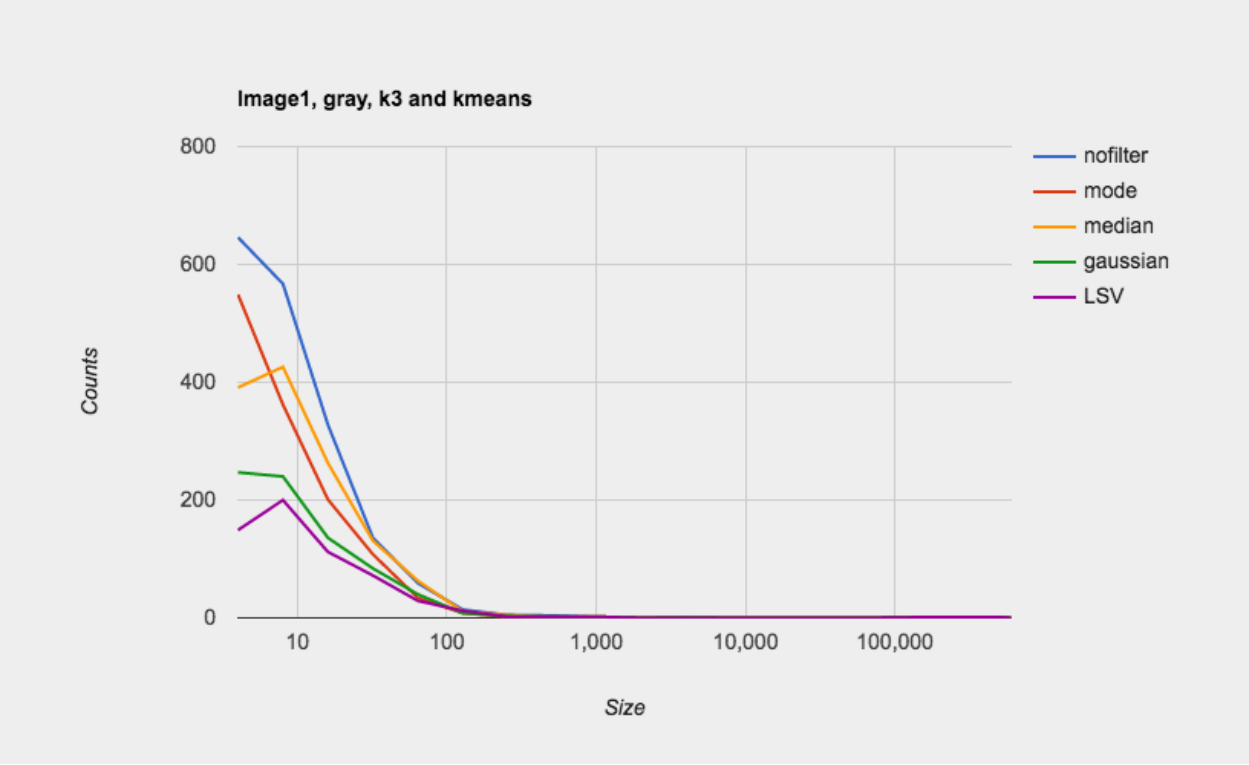

Figure 52 sizes counts chart for image 1, channel gray and k3 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
1	g	3	cmeans	noFilter	1246	359	56	4	2	1	0	1	2	0
1	g	3	cmeans	mode	838	225	30	7	1	1	0	1	2	0
1	g	3	cmeans	median	817	288	57	6	0	1	0	1	2	0
1	g	3	cmeans	gaussian	397	173	29	8	1	1	0	1	2	0
1	g	3	cmeans	LSV	254	157	41	7	0	1	0	1	2	0

Table 2 Sizes counts for Image 1, channel gray and k3 using c-means

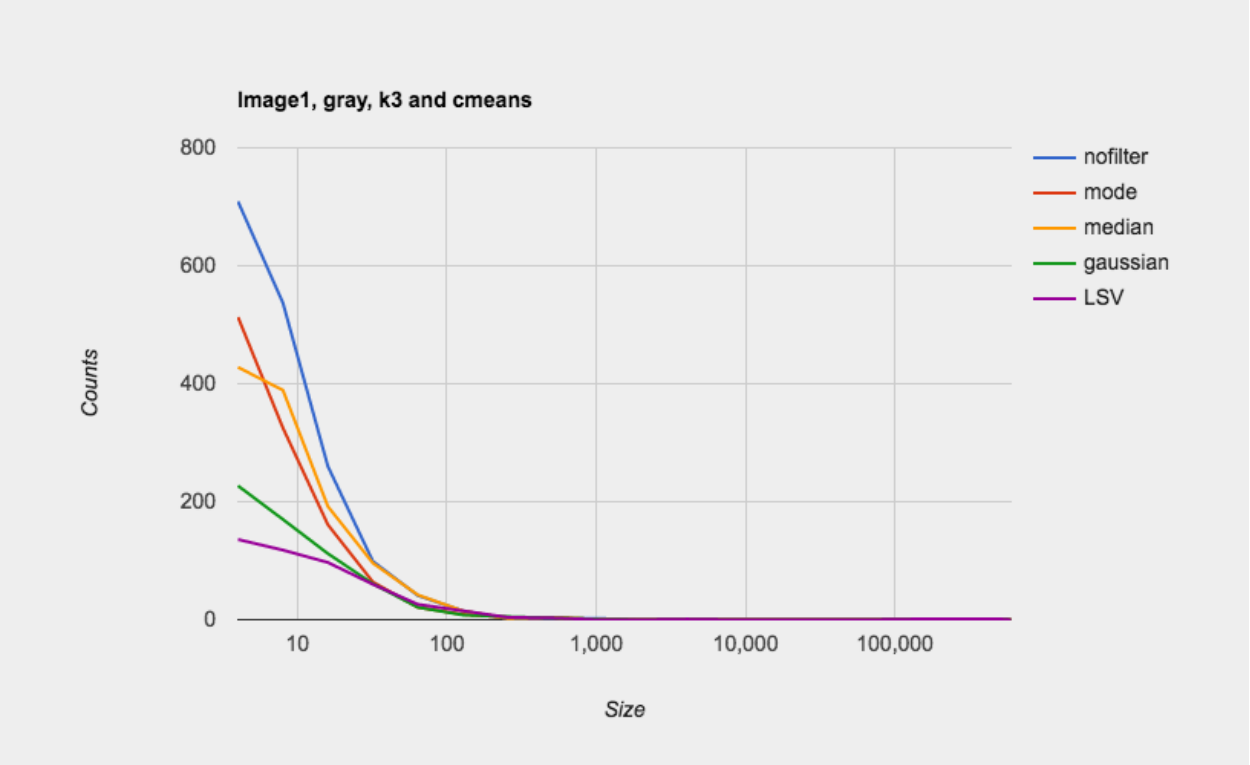

Figure 53 sizes counts chart for image 1, channel gray and k3 using c-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
1	g	5	kmeans	noFilter	123	41	9	1	0	0	0	1	2	0
1	g	5	kmeans	mode	158	61	25	1	0	0	0	1	2	0
1	g	5	kmeans	median	62	24	9	0	0	0	0	1	2	0
1	g	5	kmeans	gaussian	48	20	5	0	0	0	0	1	2	0
1	g	5	kmeans	LSV	25	12	5	0	0	0	0	1	2	0

Table 3 Sizes counts for Image 1, channel gray and k5 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
1	g	5	cmeans	noFilter	489	241	76	18	1	0	0	1	2	0
1	g	5	cmeans	mode	447	253	63	10	1	0	0	2	1	0
1	g	5	cmeans	median	342	202	75	17	0	0	0	1	2	0
1	g	5	cmeans	gaussian	289	191	64	13	0	0	0	2	1	0
1	g	5	cmeans	LSV	196	177	63	12	1	0	0	1	2	0

Table 4 Sizes counts for Image 1, channel gray and k5 using c-means

Tables 1 and 2 show size counts for the segmentation results of the gray channel for the first image using k-means and fuzzy c-means methods. The size counts for small sizes are significantly affected by filtering. There is remarkable reduction in those size counts, which indicate noise reduction. Table 1 (k-means) shows less noise then Table 2 (fuzzy c-means) for the no filter segmentation. For the filtered segmentation results, Table 2 (fuzzy c-means) shows less noise then Table 1 (k-means). It is interesting that, with the exception of the no filter, fuzzy c-means with filtering is less noisy then the k-means. Table 3 (k-means with centroids k=5) and Table 4 (fuzzy c-means with centroids k=5) show less noise then the centroids k=3 tables. Centroids k=5 is giving less noise the centroids k=3.

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
1	g&h	3	kmeans	noFilter	1100	437	80	8	1	0	0	1	2	0
1	g&h	3	kmeans	mode	841	296	48	4	0	1	0	1	2	0
1	g&h	3	kmeans	median	780	370	81	7	1	0	0	1	2	0
1	g&h	3	kmeans	gaussian	435	189	44	2	2	1	0	1	2	0
1	g&h	3	kmeans	LSV	289	176	37	3	2	1	0	1	2	0

Table 5 Sizes counts for Image 1, channel gray & hue and k3 using k-means

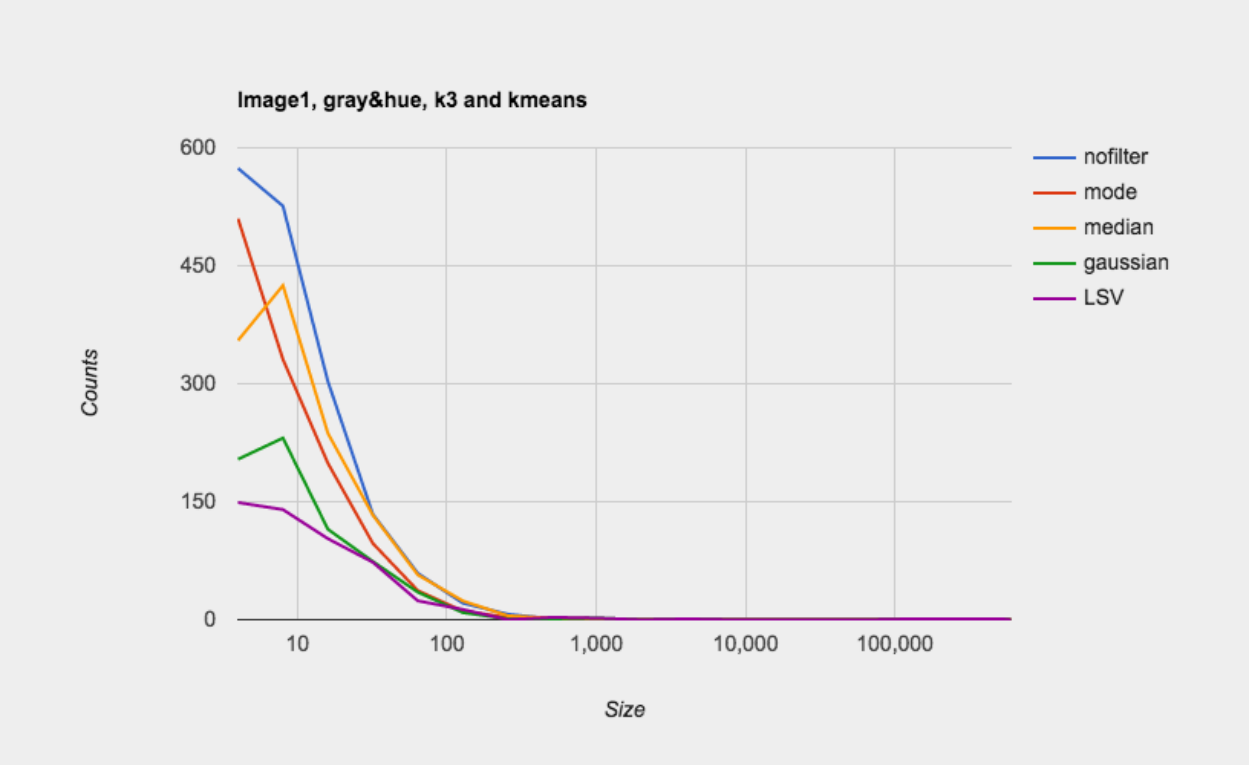

Figure 54 sizes counts chart for image 1, channel gray & hue and k3 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
1	g&h	3	cmeans	noFilter	947	261	49	20	2	2	1	1	2	0
1	g&h	3	cmeans	mode	587	147	49	9	1	1	0	1	2	0
1	g&h	3	cmeans	median	634	252	68	14	2	1	1	1	2	0
1	g&h	3	cmeans	gaussian	261	151	50	14	3	2	0	1	2	0
1	g&h	3	cmeans	LSV	211	141	55	10	4	2	0	1	2	0

Table 6 Sizes counts for Image 1, channel gray & hue and k3 using c-means

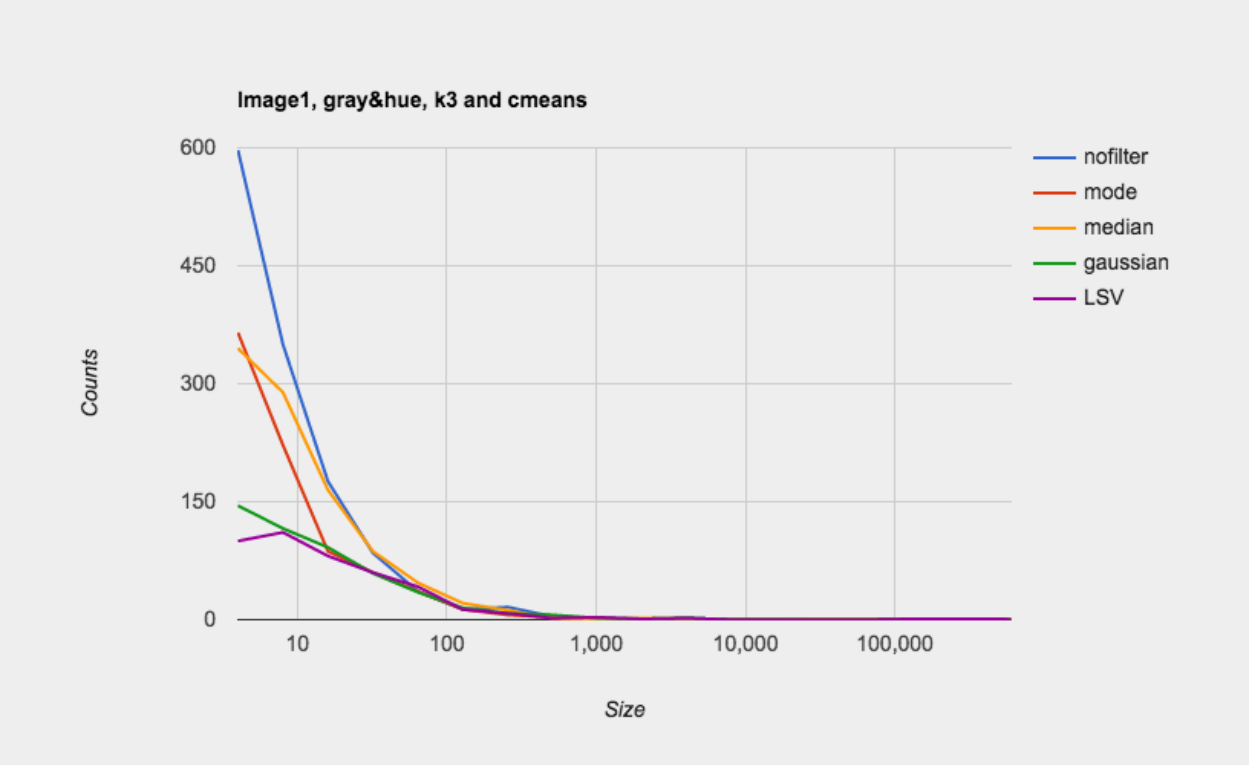

Figure 55 sizes counts chart for image 1, channel gray & hue and k3 using c-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
1	g&h	5	kmeans	noFilter	13	5	1	0	0	0	1	0	0	1
1	g&h	5	kmeans	mode	19	5	0	0	0	0	1	0	0	1
1	g&h	5	kmeans	median	13	2	2	0	0	0	1	0	0	1
1	g&h	5	kmeans	gaussian	19	11	2	0	0	0	1	0	0	1
1	g&h	5	kmeans	LSV	4	16	3	0	0	0	1	0	0	1

Table 7 Sizes counts for Image 1, channel gray & hue and k5 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
1	g&h	5	cmeans	noFilter	898	367	78	7	2	0	0	1	2	0
1	g&h	5	cmeans	mode	868	286	41	2	0	1	0	1	2	0
1	g&h	5	cmeans	median	777	279	52	3	2	1	0	1	2	0
1	g&h	5	cmeans	gaussian	294	176	51	10	0	0	0	2	1	0
1	g&h	5	cmeans	LSV	265	161	42	3	2	1	0	1	2	0

Table 8 Sizes counts for Image 1, channel gray & hue and k5 using c-means

Tables 5 and 6 show size counts for the segmentation results for the first image, but of gray with hue channels using k-means and fuzzy c-means methods. The sizes count for small sizes are significantly affected by filtering. There is remarkable reduction in those size counts which mean noise reduction. In these tables (gray with hue channels) Table 6 (fuzzy c-means) shows less noise then Table 5 (k-means). It is interesting that the gray with hue channels show fuzzy c-means is less noisy then the k-means. Table 7 (k-means with centroids k=5) only shows less noise then the centroids k=3 tables, while Table 8 (fuzzy c-means with centroids k=5) show close results to Table 5 and Table 6.

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
1	g&t	3	kmeans	noFilter	1252	462	63	9	2	0	0	1	2	0
1	g&t	3	kmeans	mode	926	292	39	5	0	1	0	1	2	0
1	g&t	3	kmeans	median	889	389	61	8	1	0	0	1	2	0
1	g&t	3	kmeans	gaussian	530	350	93	22	5	2	3	0	2	0
1	g&t	3	kmeans	LSV	349	300	81	20	6	1	4	0	2	0

Table 9 Sizes counts for Image 1, channel gray & variance textures and k3 using k-means

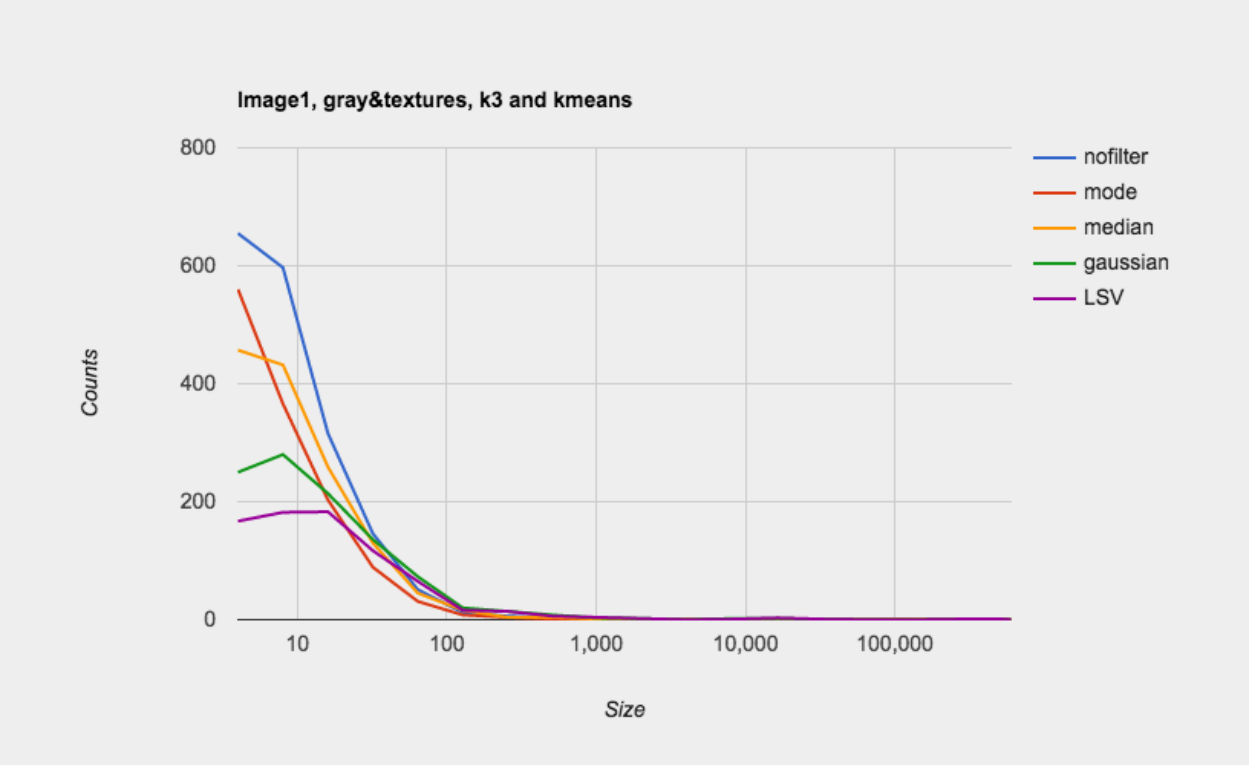

Figure 56 sizes counts chart for image 1, channel gray & variance textures and k3 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
1	g&t	3	cmeans	noFilter	1056	268	65	16	4	1	0	1	2	0
1	g&t	3	cmeans	mode	642	148	52	9	1	1	0	1	2	0
1	g&t	3	cmeans	median	651	253	73	17	1	2	0	1	2	0
1	g&t	3	cmeans	gaussian	282	168	65	20	3	3	0	1	2	0
1	g&t	3	cmeans	LSV	216	159	68	20	3	3	0	1	2	0

Table 10 Sizes counts for Image 1, channel gray & variance textures and k3 using c-means

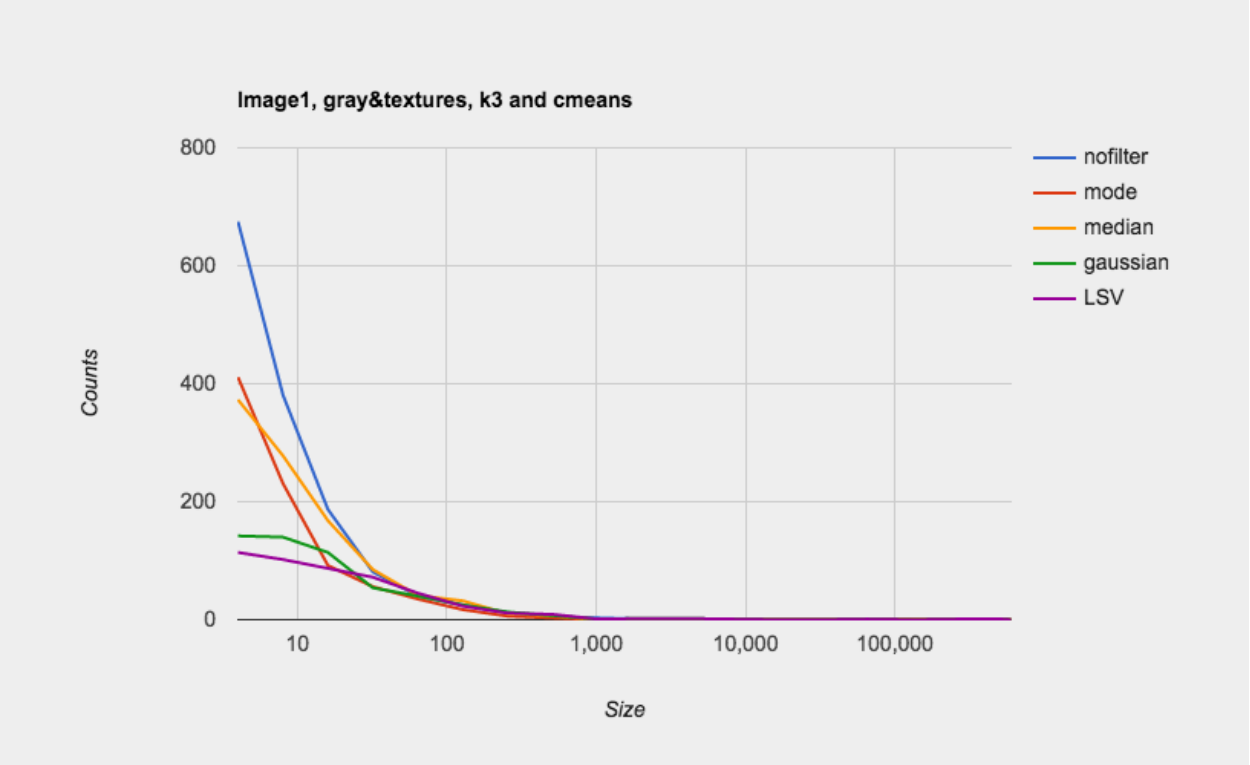

Figure 57 sizes counts chart for image 1, channel gray & variance textures and k3 using c-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
1	g&t	5	kmeans	noFilter	182	66	20	3	0	0	0	1	2	0
1	g&t	5	kmeans	mode	241	109	40	6	1	0	0	1	1	0
1	g&t	5	kmeans	median	836	364	61	7	0	0	0	1	2	0
1	g&t	5	kmeans	gaussian	40	12	4	0	0	0	0	1	2	0
1	g&t	5	kmeans	LSV	272	231	95	9	0	0	0	2	1	0

Table 11 Sizes counts for Image 1, channel gray & variance textures and k5 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
1	g&t	5	cmeans	noFilter	1250	467	68	7	1	0	0	1	2	0
1	g&t	5	cmeans	mode	912	360	56	6	0	1	0	1	2	0
1	g&t	5	cmeans	median	842	273	48	11	0	1	0	1	2	0
1	g&t	5	cmeans	gaussian	379	149	37	12	2	1	0	1	2	0
1	g&t	5	cmeans	LSV	266	122	34	14	3	1	0	1	2	0

Table 12 Sizes counts for Image 1, channel gray & variance textures and k5 using c-means

Tables 9 and 10 show size counts for the segmentation results for the first image, but of gray with variance texture channels using k-means and fuzzy c-means methods. The size count for small sizes is significantly affected by filtering. There is remarkable reduction in those size counts which means noise reduction. In these tables (gray with variance texture channels) Table 10 (fuzzy c-means) also shows less noise then Table 9 (k-means). It is interesting that the gray with hue channels show fuzzy c-means is less noise then the k-means. Table 7 (k-means with centroids k=5) and Table 8 (fuzzy c-means with centroids k=5) show close results to Table 9 and Table 10.

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
2	g	3	kmeans	noFilter	1001	423	102	22	1	0	0	1	2	0
2	g	3	kmeans	mode	756	289	88	15	1	0	0	1	2	0
2	g	3	kmeans	median	668	346	107	24	1	0	0	1	2	0
2	g	3	kmeans	gaussian	395	252	84	23	0	0	0	1	2	0
2	g	3	kmeans	LSV	298	240	83	27	0	0	0	1	2	0

Table 13 Sizes counts for Image 2, channel gray and k3 using k-means

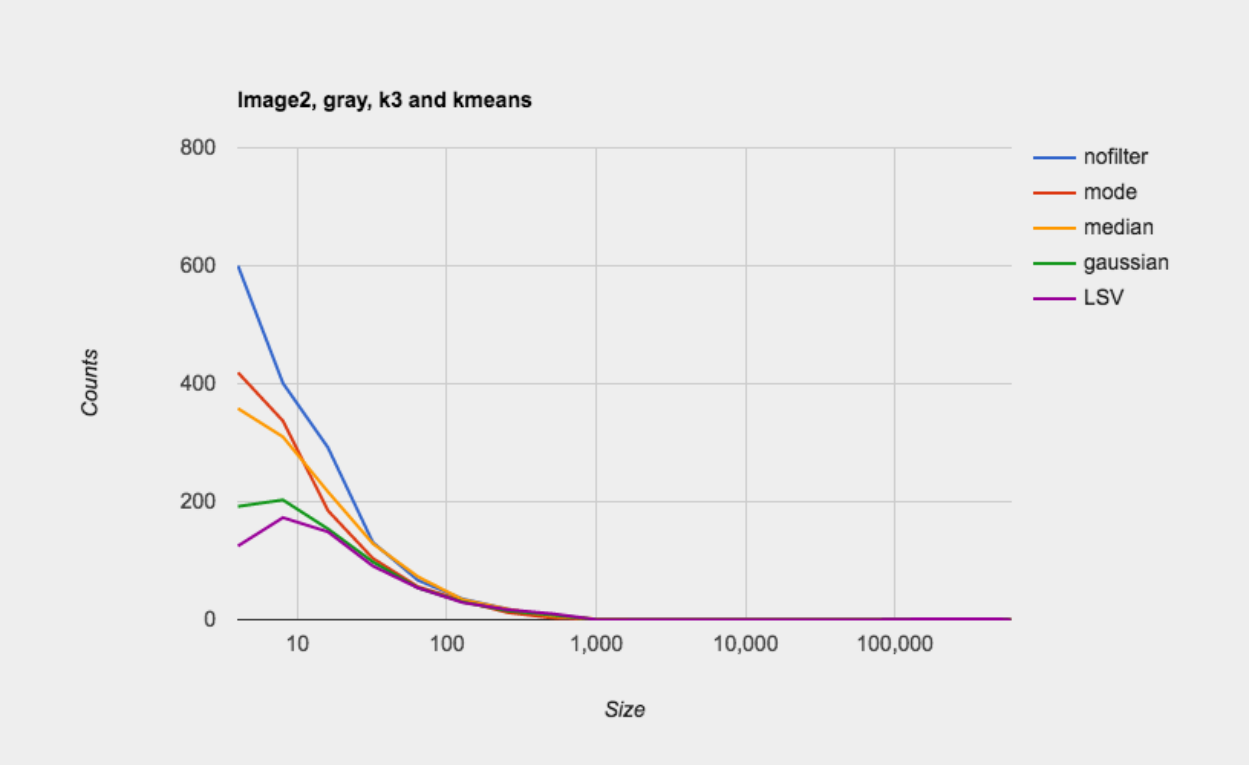

Figure 58 sizes counts chart for image 2, channel gray and k3 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
2	g	3	cmeans	noFilter	1137	405	110	25	0	0	0	1	2	0
2	g	3	cmeans	mode	787	299	85	13	1	0	0	1	2	0
2	g	3	cmeans	median	732	363	124	26	2	0	1	1	2	0
2	g	3	cmeans	gaussian	395	250	86	27	1	0	0	1	2	0
2	g	3	cmeans	LSV	296	220	91	28	1	0	0	1	2	0

Table 14 Sizes counts for Image 2, channel gray and k3 using c-means

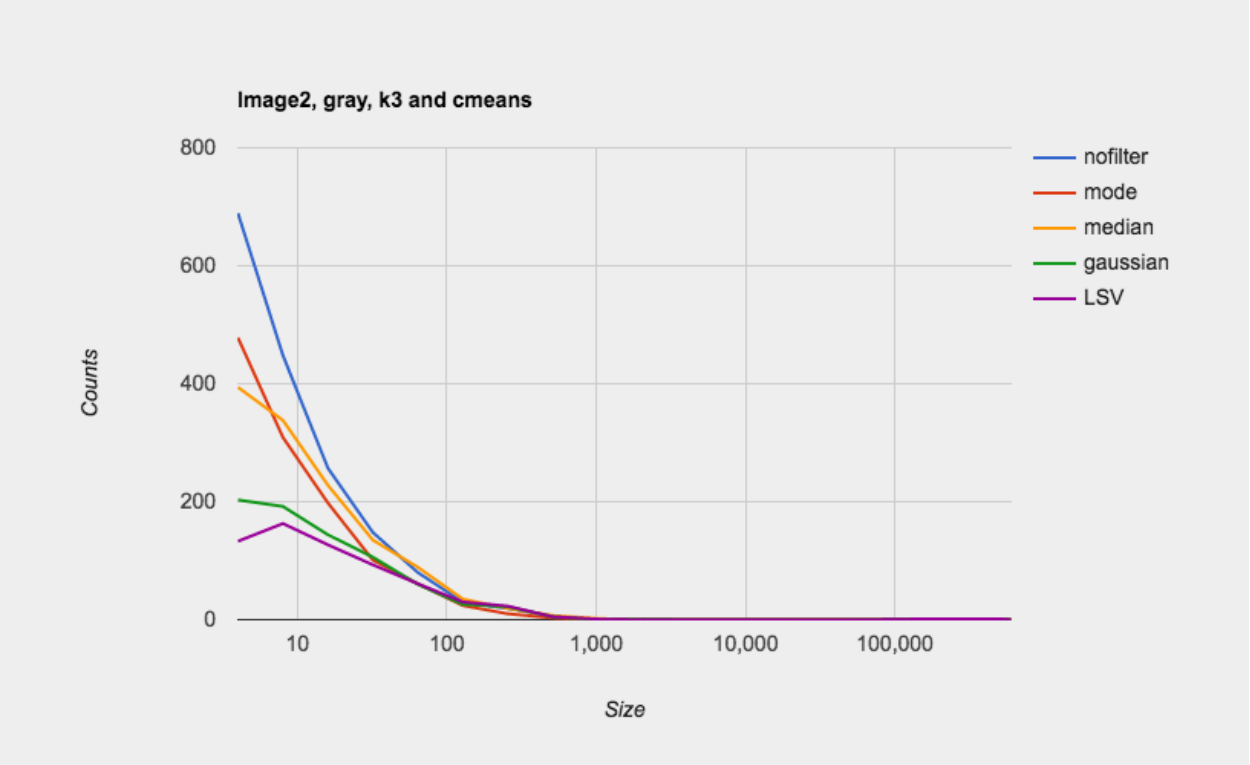

Figure 59 sizes counts chart for image 2, channel gray and k3 using c-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
2	g	5	kmeans	noFilter	404	158	44	5	0	0	0	2	1	0
2	g	5	kmeans	mode	306	159	20	9	0	0	0	1	1	0
2	g	5	kmeans	median	238	143	34	5	0	0	0	2	1	0
2	g	5	kmeans	gaussian	211	139	57	6	0	0	0	2	1	0
2	g	5	kmeans	LSV	178	145	58	10	1	0	0	2	1	0

Table 15 Sizes counts for Image 2, channel gray and k5 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
2	g	5	cmeans	noFilter	698	278	75	14	1	0	0	2	1	0
2	g	5	cmeans	mode	635	279	80	14	1	0	0	2	1	0
2	g	5	cmeans	median	460	263	80	14	2	0	0	2	1	0
2	g	5	cmeans	gaussian	330	227	69	12	1	0	0	2	1	0
2	g	5	cmeans	LSV	254	198	74	18	1	0	0	2	1	0

Table 16 Sizes counts for Image 2, channel gray and k5 using c-means

Tables 13 and 14 show size counts for the segmentation results for gray channel of the second image using k-means and fuzzy c-means methods. The size count for small sizes is significantly affected by filtering. There is remarkable reduction in those size counts, which means noise reduction. Table 13 (k-means) shows less noise then Table 14 (fuzzy c-means) for the no filter segmentation. For the filtered segmentation results, Table 14 (fuzzy c-means) shows close results to Table 13 (k-means). Table 15 (k-means with centroids k=5) and Table 16 (fuzzy c-means with centroids k=5) show less noise then the centroids k=3 tables. Centroids k=5 is giving less noise the centroids k=3 here as well.

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
2	g&h	3	kmeans	noFilter	860	372	97	21	0	0	0	2	1	0
2	g&h	3	kmeans	mode	893	322	92	13	0	0	0	1	2	0
2	g&h	3	kmeans	median	616	332	100	29	0	0	0	1	2	0
2	g&h	3	kmeans	gaussian	405	235	91	25	0	0	0	1	2	0
2	g&h	3	kmeans	LSV	145	133	38	6	6	6	1	3	1	0

Table 17 Sizes counts for Image 2, channel gray & hue and k3 using k-means

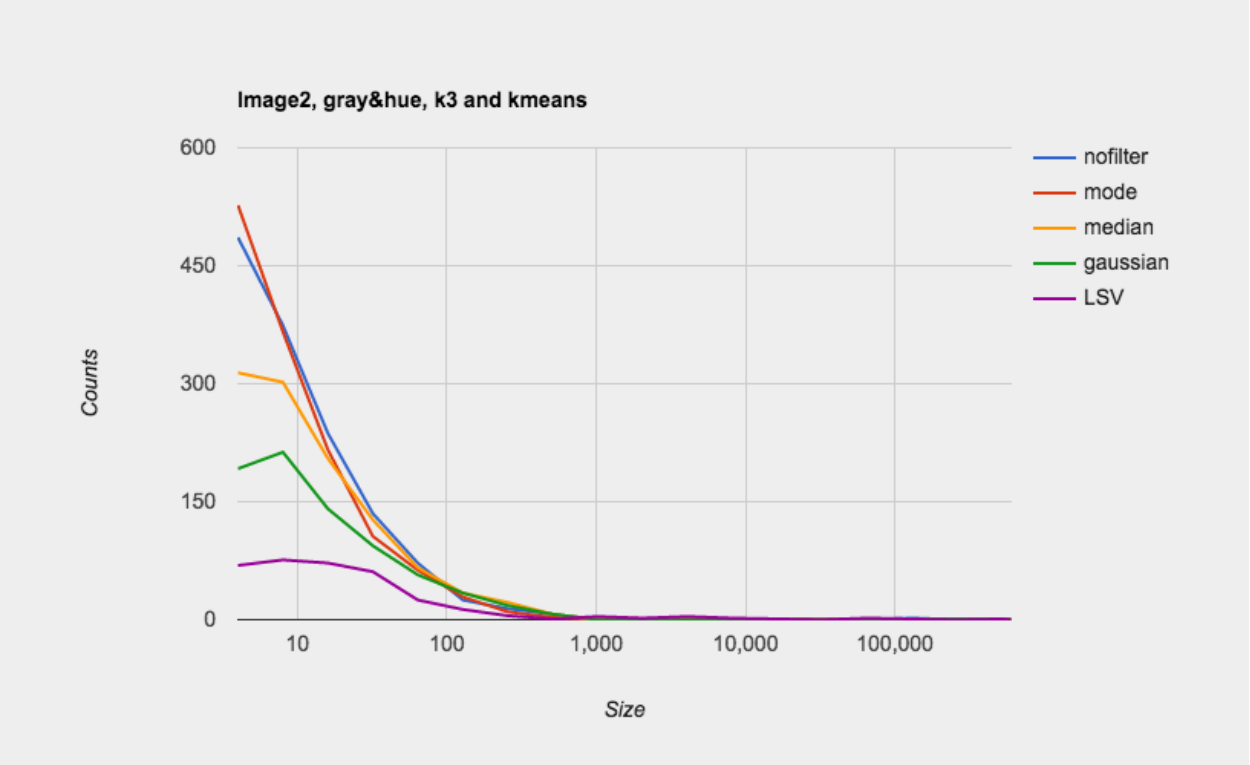

Figure 60 sizes counts chart for image 2, channel gray & hue and k3 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
2	g&h	3	cmeans	noFilter	1069	402	116	26	1	0	1	1	2	0
2	g&h	3	cmeans	mode	463	169	38	7	4	2	1	1	1	0
2	g&h	3	cmeans	median	724	393	109	29	1	0	1	1	2	0
2	g&h	3	cmeans	gaussian	233	179	67	27	7	7	1	2	1	0
2	g&h	3	cmeans	LSV	169	162	64	21	13	12	1	5	1	0

Table 18 Sizes counts for Image 2, channel gray & hue and k3 using c-means

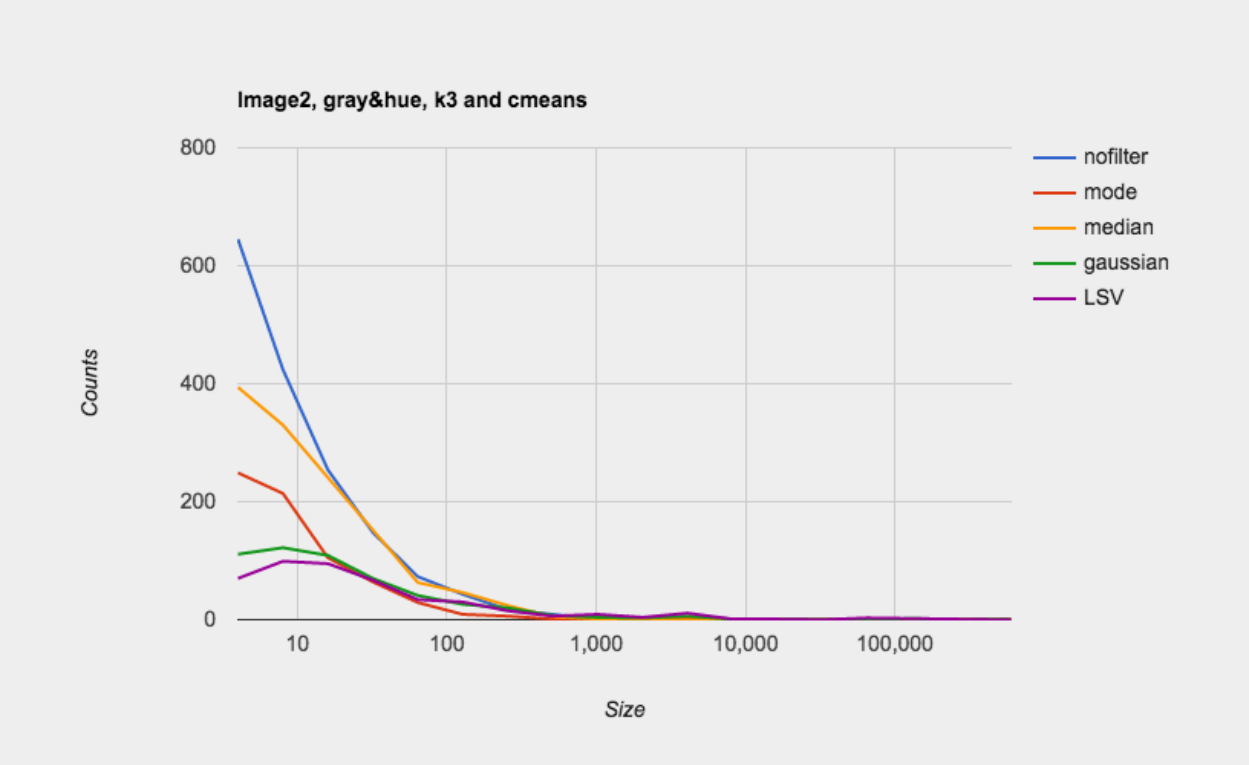

Figure 61 sizes counts chart for image 2, channel gray & hue and k3 using c-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
2	g&h	5	kmeans	noFilter	731	358	98	19	1	0	0	2	1	0
2	g&h	5	kmeans	mode	823	329	78	20	0	0	0	1	2	0
2	g&h	5	kmeans	median	500	271	78	18	1	0	0	2	1	0
2	g&h	5	kmeans	gaussian	386	259	77	20	1	0	0	2	1	0
2	g&h	5	kmeans	LSV	316	239	70	22	1	0	0	2	1	0

Table 19 Sizes counts for Image 2, channel gray & hue and k5 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
2	g&h	5	cmeans	noFilter	412	138	25	1	0	0	0	1	1	0
2	g&h	5	cmeans	mode	309	130	34	5	2	0	0	1	1	0
2	g&h	5	cmeans	median	203	131	45	11	3	3	1	1	1	0
2	g&h	5	cmeans	gaussian	149	124	38	15	4	5	1	1	1	0
2	g&h	5	cmeans	LSV	65	18	4	0	0	0	0	1	0	1

Table 20 Sizes counts for Image 2, channel gray & hue and k5 using c-means

Tables 17 and 18 show size counts for the segmentation results for the second image, but of gray with hue channels using k-means and fuzzy c-means methods. The size count for small sizes is significantly affected by filtering. There is remarkable reduction in those size counts which means noise reduction. In these tables (gray with hue channels) Table 17 (k-means) shows less noise then Table 18 (fuzzy c-means). The opposite for Table 20 (fuzzy c-means with centroids k=5) shows less noise then 19 (k-means with centroids k=5).

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
2	g&t	3	kmeans	noFilter	1093	432	145	25	1	0	1	1	2	0
2	g&t	3	kmeans	mode	1091	407	89	15	2	1	2	1	2	0
2	g&t	3	kmeans	median	590	339	107	24	1	0	2	1	2	0
2	g&t	3	kmeans	gaussian	420	301	123	27	1	0	1	1	2	0
2	g&t	3	kmeans	LSV	294	275	114	26	1	0	1	1	2	0

Table 21 Sizes counts for Image 2, channel gray & variance textures and k3 using k-means

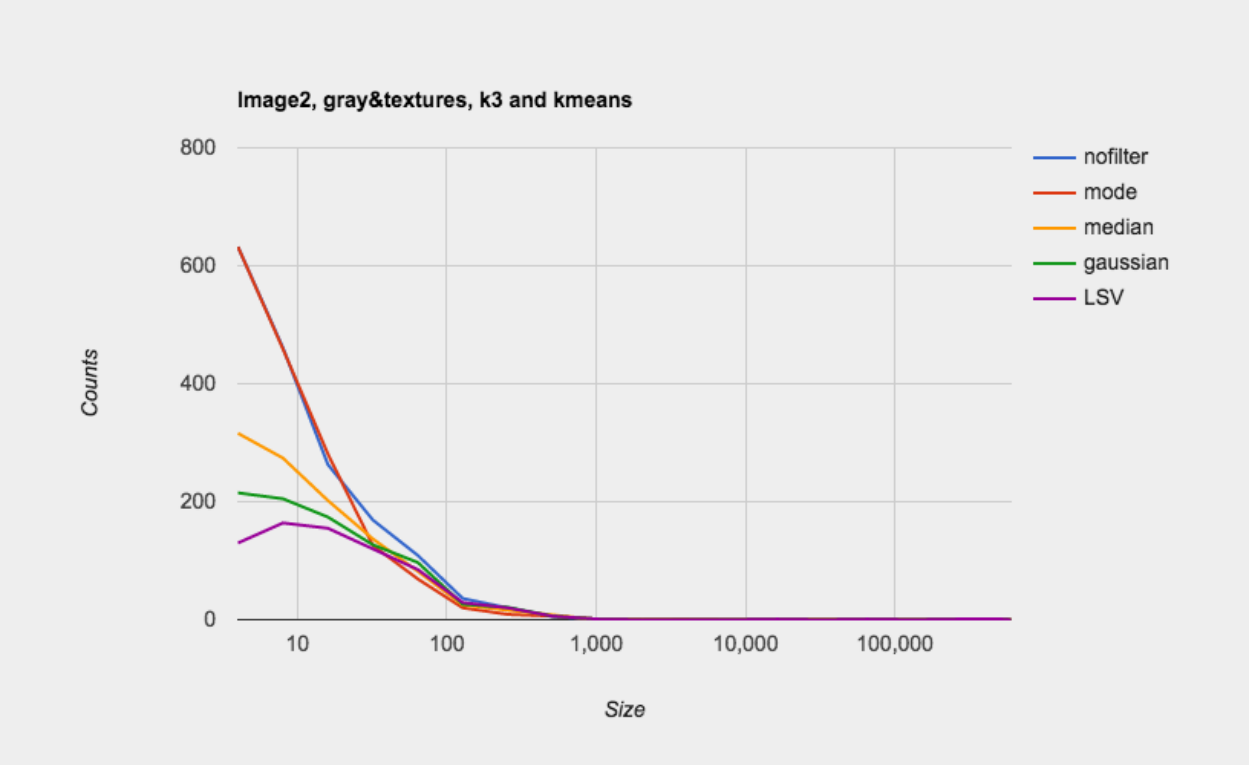

Figure 62 sizes counts chart for image 2, channel gray & variance textures and k3 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
2	g&t	3	cmeans	noFilter	1005	427	66	6	5	1	1	1	2	0
2	g&t	3	cmeans	mode	832	283	52	7	4	1	1	1	2	0
2	g&t	3	cmeans	median	672	414	82	9	4	1	1	1	2	0
2	g&t	3	cmeans	gaussian	425	243	33	10	4	1	2	1	2	0
2	g&t	3	cmeans	LSV	336	186	32	14	4	1	1	1	2	0

Table 22 Sizes counts for Image 2, channel gray & variance textures and k3 using c-means

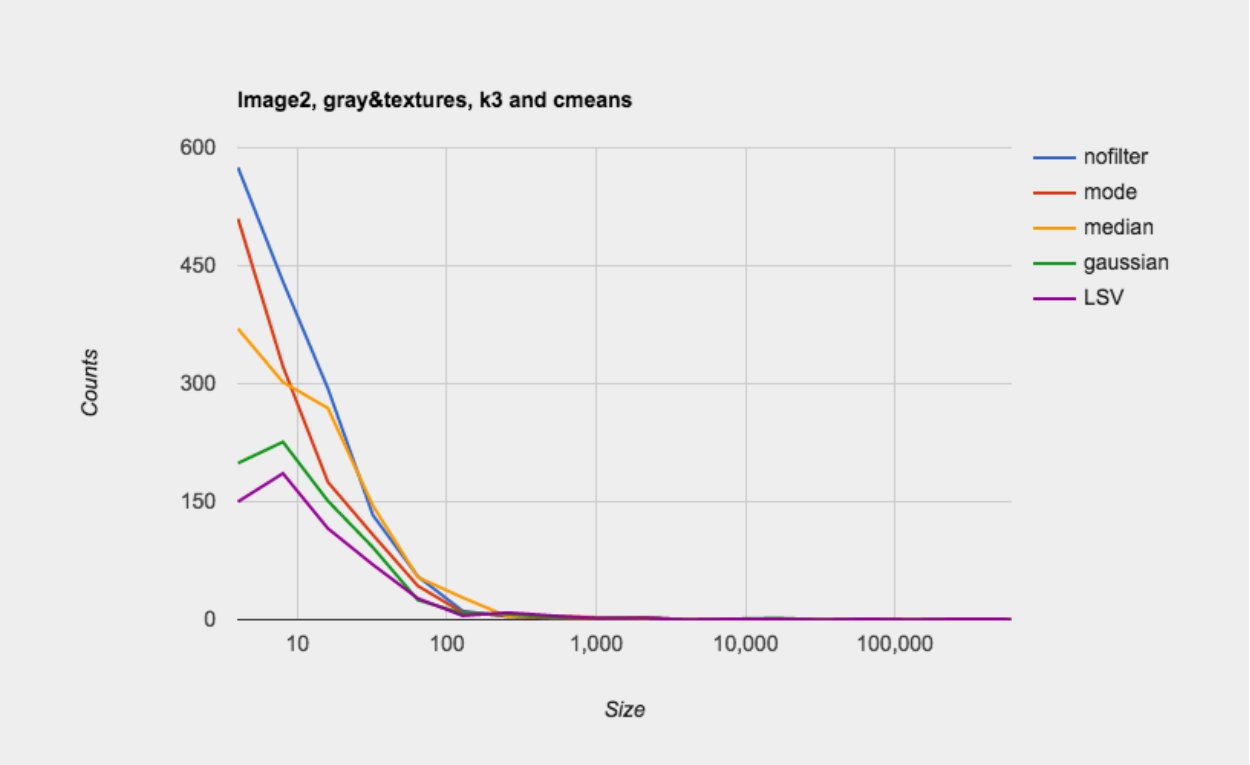

Figure 63 sizes counts chart for image 2, channel gray & variance textures and k3 using c-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
2	g&t	5	kmeans	noFilter	724	286	66	13	2	0	0	1	1	0
2	g&t	5	kmeans	mode	735	271	77	24	1	0	0	2	1	0
2	g&t	5	kmeans	median	523	251	71	14	1	0	0	1	1	0
2	g&t	5	kmeans	gaussian	307	173	69	15	1	0	0	1	1	0
2	g&t	5	kmeans	LSV	246	150	71	15	2	0	0	1	1	0

Table 23 Sizes counts for Image 2, channel gray & variance textures and k5 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
2	g&t	5	cmeans	noFilter	1057	441	104	7	1	1	0	1	2	0
2	g&t	5	cmeans	mode	872	312	92	7	2	0	0	1	2	0
2	g&t	5	cmeans	median	652	344	99	12	2	1	0	1	2	0
2	g&t	5	cmeans	gaussian	369	240	79	10	1	1	0	1	2	0
2	g&t	5	cmeans	LSV	284	214	86	8	2	0	0	0	2	0

Table 24 Sizes counts for Image 2, channel gray & variance textures and k5 using c-means

Tables 21 and 22 show size counts for the segmentation results for the second image, but of gray with variance texture channels using k-means and fuzzy c-means methods. The size count for small sizes is significantly affected by filtering. There is remarkable reduction in those sizes counts, which means noise reduction. In these tables (gray with variance texture channels) Table 22 (fuzzy c-means) shows close results to Table 21 (k-means). For the centroids k=5, Table 23 (k-means) shows less noise then Table 24 (fuzzy c-means) only in the no filter segmentation, but slightly the same with the filtering segmentation.

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
3	g	3	kmeans	noFilter	195	40	8	2	0	0	0	0	2	0
3	g	3	kmeans	mode	247	75	20	1	0	0	0	0	2	0
3	g	3	kmeans	median	77	25	4	2	0	0	0	0	2	0
3	g	3	kmeans	gaussian	41	19	4	2	0	0	0	0	2	0
3	g	3	kmeans	LSV	15	19	2	3	0	0	0	1	1	0

Table 25 Sizes counts for Image 3, channel gray and k3 using k-means

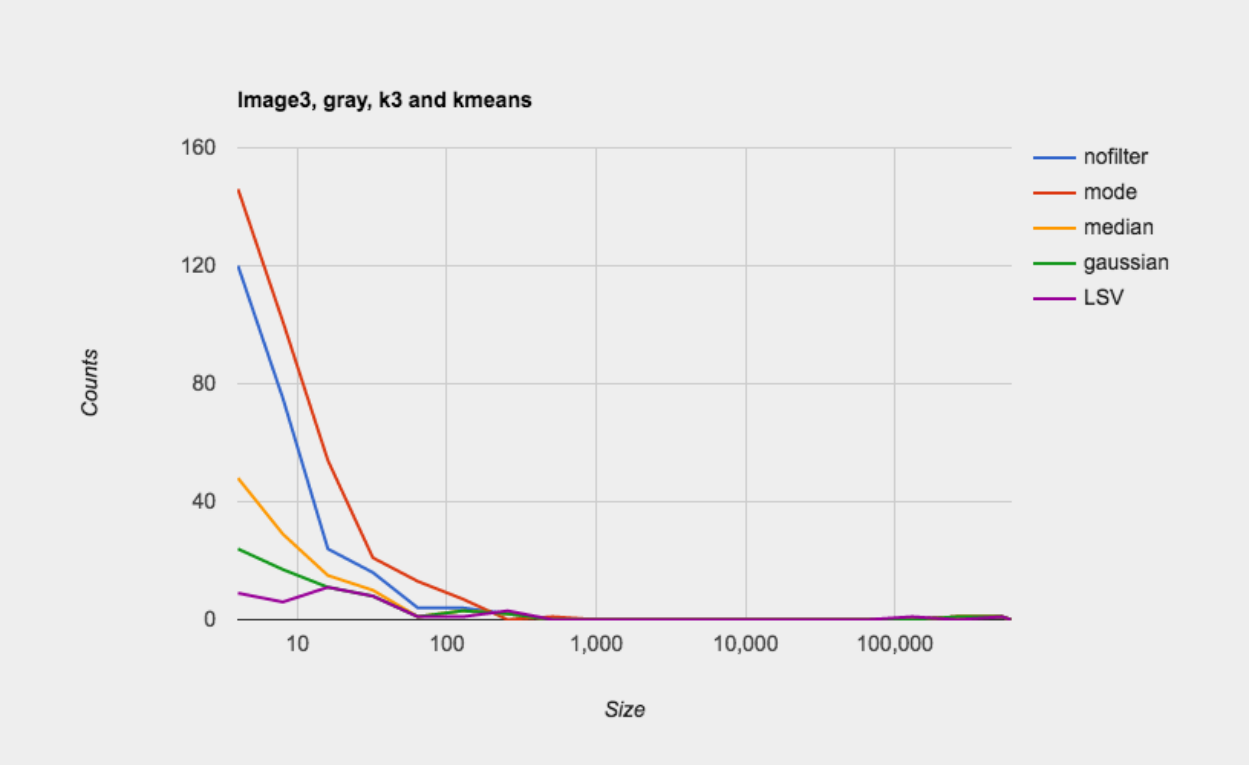

Figure 64 sizes counts chart for image 3, channel gray and k3 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
3	g	3	cmeans	noFilter	223	45	10	2	0	0	0	0	2	0
3	g	3	cmeans	mode	247	75	20	1	0	0	0	0	2	0
3	g	3	cmeans	median	100	39	10	2	0	0	0	0	2	0
3	g	3	cmeans	gaussian	48	18	5	2	0	0	0	0	2	0
3	g	3	cmeans	LSV	30	15	4	3	0	0	0	0	2	0

Table 26 Sizes counts for Image 3, channel gray and k3 using c-means

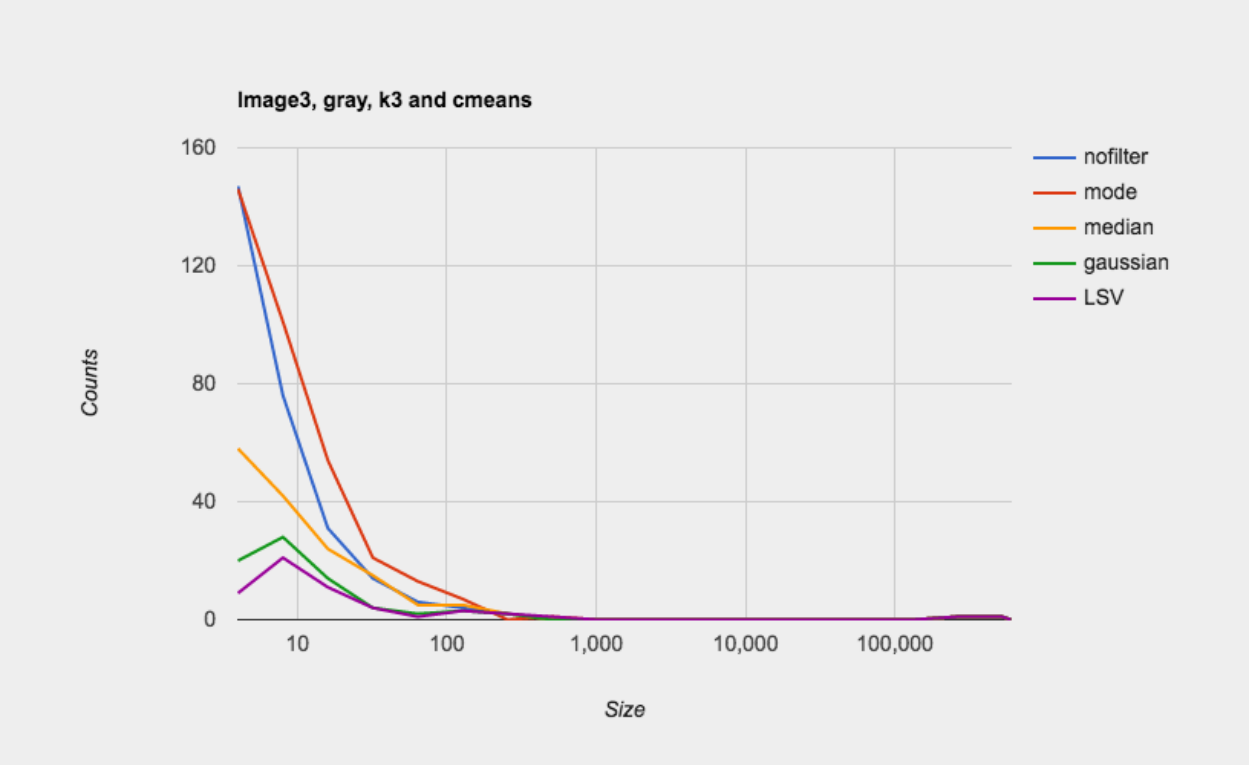

Figure 65 sizes counts chart for image 3, channel gray and k3 using c-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
3	g	5	kmeans	noFilter	89	15	7	4	0	0	0	0	2	0
3	g	5	kmeans	mode	122	40	10	5	0	0	0	0	2	0
3	g	5	kmeans	median	26	14	7	4	0	0	0	1	1	0
3	g	5	kmeans	gaussian	23	10	5	4	0	0	0	1	1	0
3	g	5	kmeans	LSV	14	9	5	4	0	0	0	1	1	0

Table 27 Sizes counts for Image 3, channel gray and k5 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
3	g	5	cmeans	noFilter	93	16	6	3	0	0	0	0	2	0
3	g	5	cmeans	mode	246	56	3	0	0	0	0	1	1	0
3	g	5	cmeans	median	44	16	7	4	0	0	0	1	1	0
3	g	5	cmeans	gaussian	20	14	6	4	0	0	0	1	1	0
3	g	5	cmeans	LSV	11	13	5	4	0	0	0	1	1	0

Table 28 Sizes counts for Image 3, channel gray and k5 using c-means

Tables 25 and 26 show size counts for the segmentation results for gray channel of the third image using k-means and fuzzy c-means methods. The size count for small sizes is significantly affected by filtering. There is remarkable reduction in those size counts, which means noise reduction. Table 25 (k-means) shows less noise then Table 26 (fuzzy c-means) in all segmentation results. Table 27 (k-means with centroids k=5) and Table 28 (fuzzy c-means with centroids k=5) show less noise then the centroids k=3 tables. Centroids k=5 here is giving less noise the centroids k=3 here as well.

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
3	g&h	3	kmeans	noFilter	41	28	7	3	1	0	0	1	1	0
3	g&h	3	kmeans	mode	29	18	4	3	1	0	0	1	1	0
3	g&h	3	kmeans	median	28	16	6	4	0	0	0	1	1	0
3	g&h	3	kmeans	gaussian	15	14	4	3	1	0	0	1	1	0
3	g&h	3	kmeans	LSV	13	13	4	3	1	0	0	1	1	0

Table 29 Sizes counts for Image 3, channel gray & hue and k3 using k-means

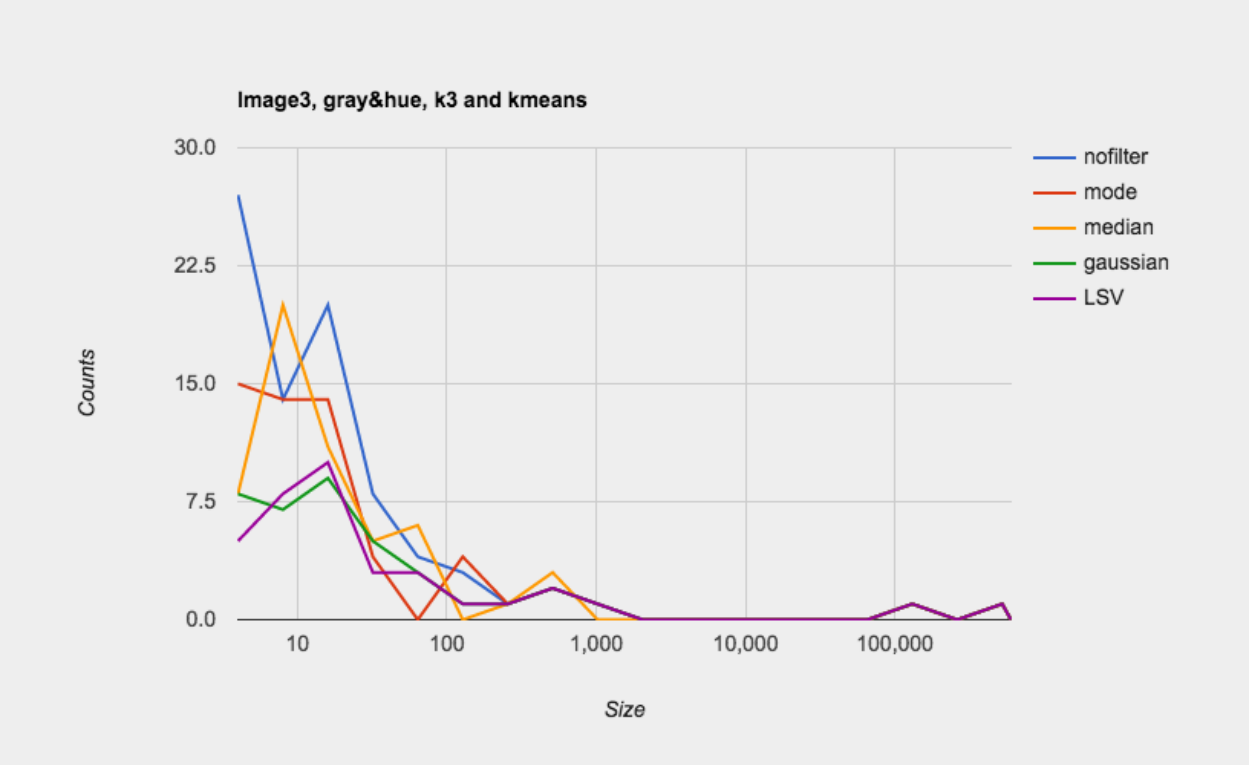

Figure 66 sizes counts chart for image 3, channel gray & hue and k3 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
3	g&h	3	cmeans	noFilter	55	19	0	3	0	0	0	1	1	0
3	g&h	3	cmeans	mode	81	15	4	3	0	0	0	0	2	0
3	g&h	3	cmeans	median	36	16	4	4	0	0	0	1	1	0
3	g&h	3	cmeans	gaussian	14	9	1	4	0	0	0	1	1	0
3	g&h	3	cmeans	LSV	14	11	2	4	0	0	0	1	1	0

Table 30 Sizes counts for Image 3, channel gray & hue and k3 using c-means

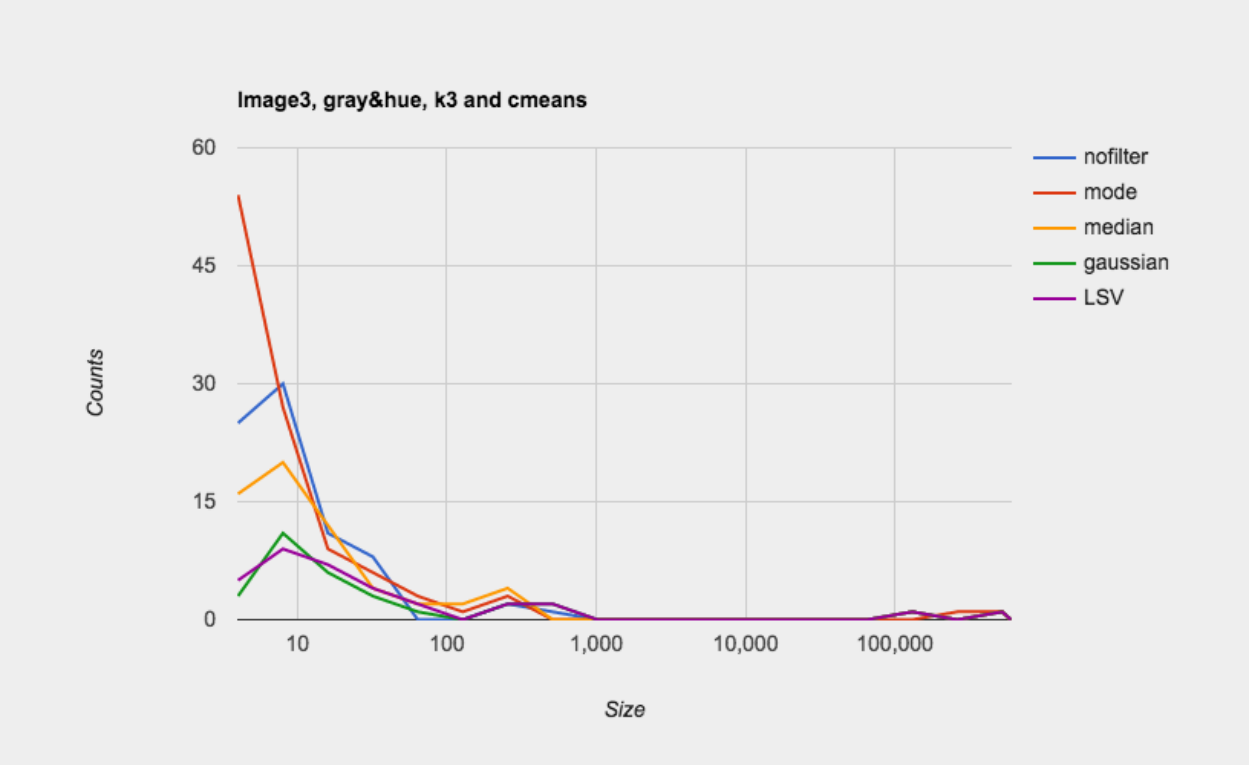

Figure 67 sizes counts chart for image 3, channel gray & hue and k3 using c-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
3	g&h	5	kmeans	noFilter	62	40	12	3	2	0	0	1	1	0
3	g&h	5	kmeans	mode	49	29	6	4	1	0	0	1	1	0
3	g&h	5	kmeans	median	46	33	10	3	2	0	0	1	1	0
3	g&h	5	kmeans	gaussian	44	25	7	4	2	0	0	1	1	0
3	g&h	5	kmeans	LSV	40	28	6	4	2	0	0	1	1	0

Table 31 Sizes counts for Image 3, channel gray & hue and k5 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
3	g&h	5	cmeans	noFilter	33	19	5	2	1	0	0	1	1	0
3	g&h	5	cmeans	mode	39	14	5	2	1	0	0	1	1	0
3	g&h	5	cmeans	median	21	23	3	4	0	0	0	1	1	0
3	g&h	5	cmeans	gaussian	18	9	3	3	1	0	0	1	1	0
3	g&h	5	cmeans	LSV	14	14	1	4	0	0	0	1	1	0

Table 32 Sizes counts for Image 3, channel gray & hue and k5 using c-means

Tables 29 and 30 show sizes counts for the segmentation results for the third image, but of gray with hue channels using k-means and fuzzy c-means methods. The size count in those tables are dispersed as the results are irregular. But, in Table 32 (fuzzy c-means with centroids k=5) and Table 33 (k-means with centroids k=5) they are more organized.

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
3	g&t	3	kmeans	noFilter	125	42	3	5	0	0	0	0	2	0
3	g&t	3	kmeans	mode	119	36	7	5	0	0	0	0	2	0
3	g&t	3	kmeans	median	54	21	2	4	0	0	0	1	1	0
3	g&t	3	kmeans	gaussian	36	14	1	4	0	0	0	1	1	0
3	g&t	3	kmeans	LSV	21	10	3	4	0	0	0	1	1	0

Table 33 Sizes counts for Image 3, channel gray & variance textures and k3 using k-means

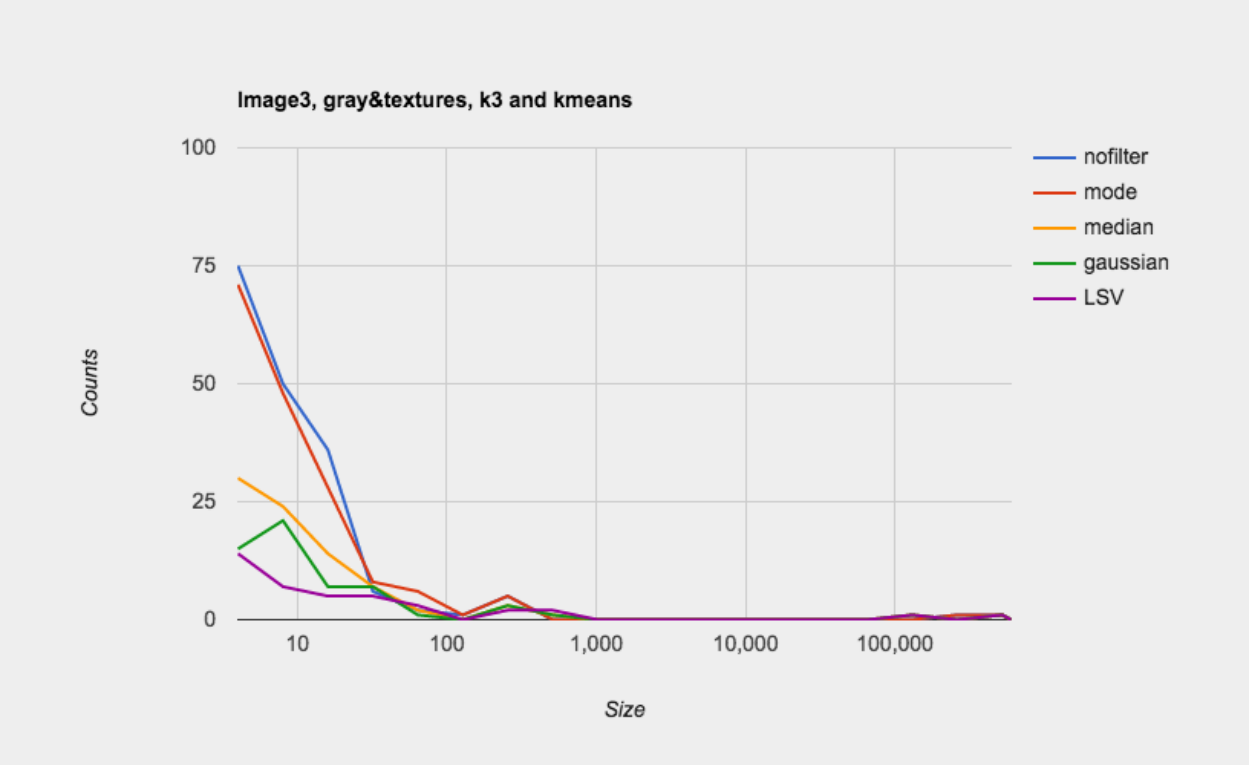

Figure 68 sizes counts chart for image 3, channel gray & variance textures and k3 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
3	g&t	3	cmeans	noFilter	306	94	13	5	0	0	0	0	2	0
3	g&t	3	cmeans	mode	277	80	21	1	0	0	0	0	2	0
3	g&t	3	cmeans	median	145	43	9	2	0	0	0	0	2	0
3	g&t	3	cmeans	gaussian	61	19	8	2	0	0	0	0	2	0
3	g&t	3	cmeans	LSV	33	12	6	3	0	0	0	0	2	0

Table 34 Sizes counts for Image 3, channel gray & variance textures and k3 using c-means

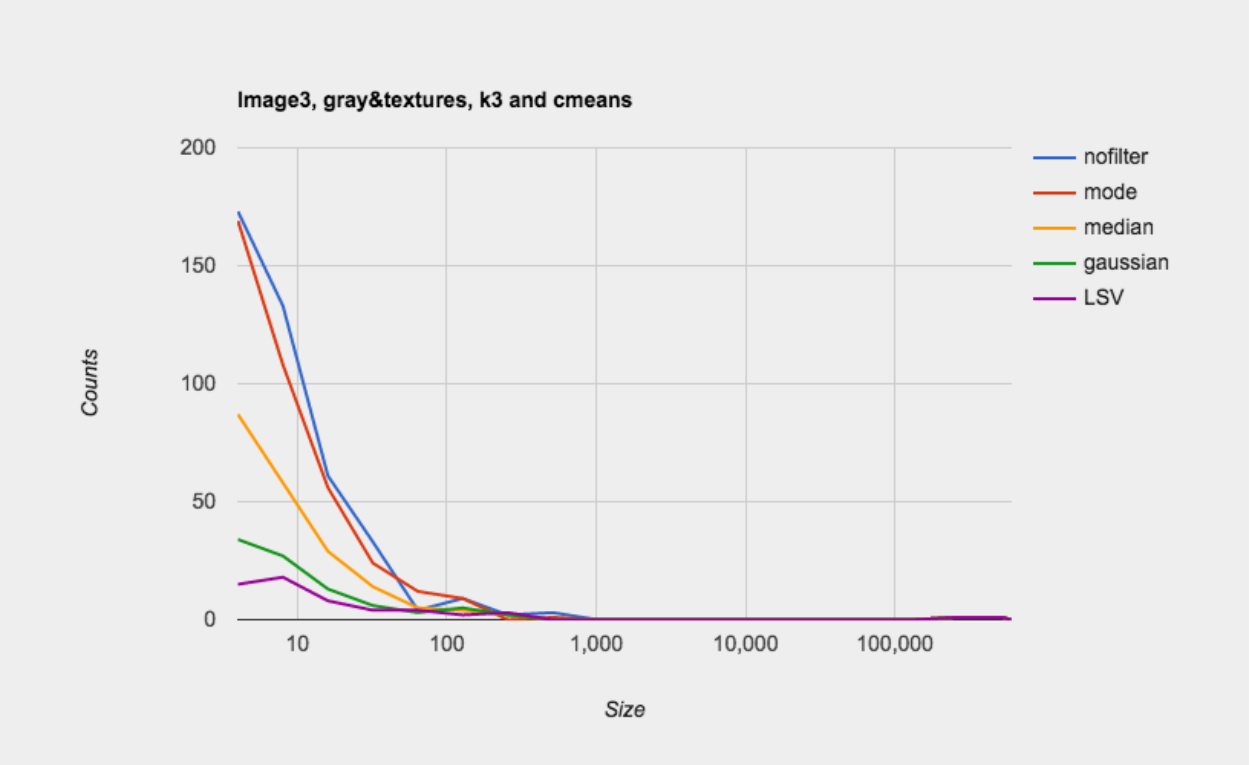

Figure 69 sizes counts chart for image 3, channel gray & variance textures and k3 using c-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
3	g&t	5	kmeans	noFilter	382	97	13	0	0	0	0	1	1	0
3	g&t	5	kmeans	mode	325	86	10	2	0	0	0	1	1	0
3	g&t	5	kmeans	median	19	16	4	4	0	0	0	1	1	0
3	g&t	5	kmeans	gaussian	9	9	2	4	0	0	0	1	1	0
3	g&t	5	kmeans	LSV	7	9	4	4	0	0	0	1	1	0

Table 35 Sizes counts for Image 3, channel gray & variance textures and k5 using k-means

					Count \ Sizes (less than)									
Image	Channel	k	Method	Filter	8	32	128	512	2048	8192	32768	131072	524288	600000
3	g&t	5	cmeans	noFilter	115	36	6	4	0	0	0	0	2	0
3	g&t	5	cmeans	mode	296	43	3	0	0	0	0	1	1	0
3	g&t	5	cmeans	median	78	31	4	5	0	0	0	0	2	0
3	g&t	5	cmeans	gaussian	28	17	2	4	0	0	0	1	1	0
3	g&t	5	cmeans	LSV	25	13	3	4	0	0	0	1	1	0

Table 36 Sizes counts for Image 3, channel gray & variance textures and k5 using c-means

Tables 33 and 34 show size counts for the segmentation results for the third image, but of gray with variance texture channels using k-means and fuzzy c-means methods. The size count for small sizes are significantly affected by filtering. There is remarkable reduction in those size counts, which mean noise reduction. In these tables (gray with variance texture channels) Table 34 (fuzzy c-means) shows more noise than Table 33 (k-means). For the centroids k=5, Table 35 (k-means) shows less noise then Table 36 (fuzzy c-means) only in the filtered segmentation, but more noise with the no filter segmentation.

4.6 Analysis - Task 2

To analyze task 2 segmentation results (4.3.2), I listed tables of water shield size over time in pixels. For Bahr al-milh Lake, there are three images, which means there are three sizes: sizes 1, 2 and 3. The first, second and third images were taken in 1995, 2003 and 2013. The change column shows the absolute changes: size 3 minus size 1. The percentage column shows the absolute changes (change column) divided by first size. The bottom right of tables there are two extra rows, average and standard deviation. The average and the standard deviation are for the percentage column. For Shasta Lake, there are two images, which means there are two sizes: sizes 1 and 2. The first and second images were taken in 2011 and 2014. The change column shows the absolute changes, which are size 2 minus size 1. The percentage column shows the absolute changes (change column) divided by size 1. The average and the standard deviation are for the percentage column.

Method	Channel	K	Filter	Size 1	Size 2	Size 3	Change	Percentage
kmeans	gray	3	noFilter	87625	50642	15021	-72604	-0.829
kmeans	gray	3	mode	91293	52504	15390	-75903	-0.831
kmeans	gray	3	gaussian	87252	49828	15680	-71572	-0.82
kmeans	gray	3	median	87518	50429	14676	-72842	-0.832
kmeans	gray	3	LSV	85338	47638	15551	-69787	-0.818
							Average	-0.826
							Standard dev.	0.0065

Table 37 Changes over time on Bahr Al-milh lake using the k-means segmentation

Method	Channel	K	Filter	Size 1	Size 2	Size 3	Change	Percentage
cmeans	gray	3	noFilter	87971	51102	15514	-72457	-0.824

cmeans	gray	3	mode	91362	52638	15863	-75499	-0.826
cmeans	gray	3	gaussian	87744	51160	15899	-71845	-0.819
cmeans	gray	3	median	87792	50899	15415	-72377	-0.824
cmeans	gray	3	LSV	86081	49539	15868	-70213	-0.816
							Average	-0.8218
							Standard dev.	0.0041

Table 38 Changes over time on Bahr Al-milh lake using the c-means segmentation

Tables 37 (k-means) and Table 38 (fuzzy c-means) show the size changes for Bahr al-milh Lake. Although the change measurements are close to each other, there are slightly more differences in the k-means results (Table 37). The percentage column shows that the statistical mode is giving the largest, while the LSV is giving the smallest. The percentage column, also, shows statistical mode and median filters were closes to the no filter results. The Gaussian and the LSV show the biggest difference from the no filter. The fuzzy c-means (Table 38) show more stable results by having fewer differences between results. The statistical mode again is slightly larger, while the LSV shows smaller sizes. For changes, the percentage column show statistical mode and median filters were closest to the no filter results. Gaussian and the LSV show smaller differences than the no filter as well. In general, both tables 37 and 38 measurements are similar to each other. Both tables averages are around 0.82 and booth standard deviation are quite small (0.0065 and 0.0041). For Bahr Al-milh, both segmentation method are giving close results and the filters are not giving big differences between themselves.

Method	Channel	K	Filter	Size 1	Size 2	Change	Percentage
kmeans	gray	3	noFilter	27057	13333	-13724	-0.507
kmeans	gray	3	mode	32710	17642	-15068	-0.461
kmeans	gray	3	gaussian	27420	12618	-14802	-0.54
kmeans	gray	3	median	26469	12277	-14192	-0.536
kmeans	gray	3	LSV	22692	9231	-13461	-0.593
						Average	-0.5274
						Standard dev.	0.0484

Table 39 Changes over time on Shasta lake using the k-means segmentation

Method	Channel	K	Filter	Size 1	Size 2	Change	Percentage
cmeans	gray	3	noFilter	28244	19610	-8634	-0.306
cmeans	gray	3	mode	33031	18987	-14044	-0.425
cmeans	gray	3	gaussian	29216	24149	-5067	-0.173
cmeans	gray	3	median	27243	16807	-10436	-0.383
cmeans	gray	3	LSV	26108	16969	-9139	-0.35
						Average	-0.3274
						Standard dev.	0.0967

Table 40 Changes over time on Shasta lake using the c-means segmentation

Table 39 (k-means) and Table 40 (fuzzy c-means) show the size changes for Shasta Lake. Especially for the fuzzy c-means results (Table 40), there are significant differences between the filters results. The change columns in both tables are vary. And the percentage column as well is showing quite diverse results. The standard deviation shows high error in the filters change percentages in both tables (0.0484 in 39 and 0.0967 in 40). Also, the average is quite different between tables (-0.32 and -0.51) which show how the segmentation methods are different. Shista Lake is an ongoing problem that are quite sensitive both filtering and segmentation methods, and

this is due to the length of the shore (water touching sand) areas and the line like narrow water areas.

5. Conclusion

I have examined aerial image segmentation aspects, starting from channel selection, multiple filtering and two segmentation methods. The examination is demonstrated in two methods: qualitative and quantitative. The qualitative approach addressed the noise problem under multiple scenarios, and the quantitative addressed the same scenario's affects on a climate change (water shield shrinking) measurement. There are multiple points to present based on this thesis project.

A first point is related to the effect of channel selection on noise reduction. Task 1 showed how hue and variance texture channels involved with gray channel reduced the noises in multiple results. These two channels are slightly smoother then the gray one with respect to the image structures and objects as well. As channel selection shows effects on image segmentation, I would like to encourage it to be considered in segmentation tasks. Channels might be selected based on the segmentation requirements to reach a more optimized solution.

The second point of this thesis is to confirm spatial filtering effects on aerial image segmentation. For task 1, all filters used (statistical mode, median Gaussian and the new filter LSV) showed a significant noise reduction. Almost all segmentations showed the filter's noise reduction strength in the order (strongest to least strong) of LSV, Gaussian, median and statistical mode. The largest singular value spatial filter was quite beneficial for noise reduction compared to the other filters examined. The approximation ability of the norm 2 is quite interesting. I would highly recommend more study on such a phenomenon. On the other hand,

filtering showed a loss of small details in the images. All other filters showed a loss of the water shield borders. For a sensitive image, like Shasta Lake, filtering show significant effects on the surface measurement. I would highly encourage careful use of filtering in cases of sensitive, minutely detailed images.

Shasta lake is a special case image. The length of its shores and the line like water shapes make the lake quite sensitive. This lake shows big differences between filtering results and between segmentation methods (k-means and c-means). This confirm a need to check images visually before start segmentation and choose segmentation methods and filtering carefully.

Finally, except Shasta lake, the segmentation method (k-means and fuzzy c-means) results are mostly close to each other. It is clear in several results that the fuzzy c-means is more sensitive. Also, I found that it is more sensitive to the filtering results as well. The fuzzy c-means sensitivity shows more response to filtering in some results, which may be studied further in future.

In general, I would recommend using spatial filters especially the new filter (LSV) in cases where small details are not wanted. In other cases where small details are needed, more focus on the fuzzy c-means are recommended, and there might be a chance to use spatial filtering carefully as well. Channel selection also is a quite interesting step, and I would recommend analyzing possible channels of an image visually to select the most channels that highlight the segmentation requirement.

Bibliography

A.E. Serebryannikov, E. C. (2016, 01). Multifrequency spatial filtering: A general property of two-dimensional photonic crystals. *Photonics and Nanostructures*.

Baumann, P. R. (2009). *History of Remote Sensing, Satellite Imagery, Part II*. Retrieved 04 11, 2016, from College at Oneonta: http://www.oneonta.edu/faculty/baumanpr/geosat2/RS%20History%20II/RS-History-Part-2.html

Bernice Ellen Rogowitz, T. N. (2007). *Human Vision and Electronic Imaging XII* . SPIE.

Bourne, R. (2010). *Fundamentals of Digital Imaging in Medicine.* New York: Springer.

Castleman, K. R. (1979). *Digital image processing.* Englewood Cliffs, NJ: Prentice-Hall.

Coates, A. &. (2012). Learning Feature Representations with K-means. *G. Montavon, G. B. Orr, K.-R. M¨uller (Eds.), Neural Networks: Tricks of the Trade, 2nd edn, Springer*.

Farag, A. A. (1992). Edge-based image segmentation. *Remote sensing reviews*.

Fricke, T. (2000). *Hoshen Kopelman algorithm*. Retrieved 05 11, 2016, from https://www.ocf.berkeley.edu/~fricke/projects/hoshenkopelman/hoshenkopelman.html

Gonzalez, R. C. (1987). *Digital image processing.* Reading, MA: Addison-Wesley.

James C. Bezdek, R. E. (1984). FCM: The Fuzzy C-means Clustering Algorithm. *Computers & Geosciences*.

Knudsen, J. B. (1999). *Java 2D Graphics.* O'Reilly.

Kopelman, J. H. (1976). Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm. *American Physical Society*.

Krein, I. C. (1069). *Introduction to the Theory of Linear Nonselfadjoint Operators in Hilbert Space.* American Mathematical Society.

Lee C., H. S. (1998). Unsupervised Connectivity-Based Thresholding Segmentation of Midsagittal Brain MR Images. *Computers in Biology and Medicine*.

Liu, J. &. (2015). An Improved K-means Algorithm for Brain MRI Image Segmentation. *Proceedings of the 3rd International Conference on Mechatronics, Robotics and Automation*.

M.SEETHA, B. &. (2008). Artificial Neural Networks and Other Methods of Image Classification . *Journal of Theoretical and Applied Information Technology* .

Opencv. (2016). *Opencv*. Retrieved 04 11, 2016, from http://opencv.org/

Pavan, K. R. (2011). Robust seed selection algorithm for k-means type algorithms. *International Journal of Computer Science and Information Technology IJCSIT*.

Pratt, W. K. (1978). *Digital image processing.* New York, NY: Wiley.

Radha, M. R. (2011). Edge Detection Techniques For Image Segmentation. *International Journal of Computer Science and Information Technology IJCSIT*.

Ramesh, K. (2000). *Digital Photoelasticity: Advanced Techniques and Applications.* New York.

Rosin, F. F. (1994). Filtering remote sensing data in the spatial and feature domains. *Image and Signal Processing for Remote Sensing*.

Shi, W. (2012). *Advances in Spatial Data Handling and GIS: 14th International Symposium on Spatial Data Handling.* Ney York: Springer Verlag.

USGS. (2015, 11 17). *United States Geological Survey*. Retrieved from http://remotesensing.usgs.gov/gallery/

Vandevenne, L. (2004). *Flood fill*. Retrieved 2016, from Lode Vandevenne: http://lodev.org/cgtutor/floodfill.html

Watkins, D. S. (2002). *Fundamentals of Matrix Computations.* Ney York: Wiley-Interscience.

Yang, Y. (2007). Image Segmentation by Fuzzy C-means Clustering Algorithm With a Novel Penalty Term. *Computing and Informatics*.

Yansun Xu, J. B. (1994). Wavelet Transform Domain Filters: A Spatially Selective Noise Filtration Technique. *IEEE TRANSACTIONS ON IMAGE PROCESSING*.

Yu Qian, K. Z. (2004, 12). Spatial contextual noise removal for post classification smoothing of remotely sensed images. *Proceedings of the 2005 ACM Symposium on Applied Computing (SAC)*.

Yud-Ren Chen, G. E.-I. (2005). *Optical Sensors and Sensing Systems for Natural Resources and Food Safety and Quality* . SPIE.

Zadeh, L. A. (1965). Fuzzy Set. *Information and Control* .

Zaixin, Z. L. (2014). Neighbourhood weighted fuzzy c-means clustering algorithm for image segmentation. *IET Image Processing*.

Zhouping Wei, J. W. (2012). A median-Gaussian filtering framework for Moiré pattern noise removal from X-ray microscopy image. *Micron*.