Segmentation methods and their evaluation

Catalina Gómez
Universidad de los Andes
Cra 1 #18a-12, Bogotá, Colombia
c.gomez10@uniandes.edu.co

Diana Herrera
Universidad de los Andes
Cra 1 #18a-12, Bogotá, Colombia
ds.herrera10@uniandes.edu.co

Abstract

*Segmentation is one of the main problems studied within the field of Computer Vision. This problem aims to group pixels corresponding to the same object on an image, that are delimited by contours. In order to address this problem, each pixel is a candidate for assigning it a semantic label, which depends on its representation within a high-dimensional space. Combination between four clustering methods (k-means, GMM, hierarchical and watersheds) and six feature spaces (RGB, La*b*, HSV and these color spaces plus xy coordinates) were tried in four images to generate three clusters. Jaccard Index was calculated and the best results were obtained with the representation in La*b* (0.50) space and with the clustering method GMM (0.63).*

1. Introduction

Segmentation is one of the main problems studied within the field of Computer Vision. This problem aims to group pixels corresponding to the same object on an image, that are delimited by contours. In order to address this problem, each pixel is a candidate for assigning it a semantic label, which depends on its representation within a high-dimensional space.

Clustering algorithms have been used for addressing segmentation problems, since these algorithms require data, but not labels to group pixels together into clusters, or regions. The goal is to assign pixels to each cluster based on a measure of similarity, that must be greater within pixels of the same cluster, and lesser between pixels from different clusters. The input for these algorithms requires a representation of the pixels with some features that allow for an accurate discrimination. Once these features are computed, the grouping result depends on the method employed by the algorithm to cluster.

Perhaps the best known segmentation method is K-means algorithm, which belongs to a family of unsupervised learning techniques. In addition, it falls into the group of hard clustering, where each data point belongs to only one cluster. It works partitioning data into k clusters, parameter that must be known a *priori*. Every point is assigned into a cluster based on its distance to the center of the cluster, called centroid, in such a way that centroids and clusters are chosen to minimize the squared distances between points and centroids [1].

A similar unsupervised segmentation method is Gaussian Mixture Model (GMM), but unlike k-means this one performs a soft assignment, giving each point the probability to belong to one or more clusters, each one represented by a Gaussian distribution and without a unique stated limit. Hence, groups in this algorithm are assumed to be normally distributed and are represented by multivariate Gaussian distributions, configured to optimize the model through the Expectation Maximization algorithm (EM). To do so, the first step (E step) is to use random parameters for the Gaussian distributions and estimate the responsibilities of each point, called soft assignments. Afterwards, M step is performed in which parameters of the Gaussian distributions are estimated again until a minimum is found. An important limitation is that the minimum found can be local and hence the result obtained would not be the optimal result [2].

When the amount of clusters desired is not known a priori neither how the data might be grouped, it is useful to look for possible structures formed on different levels. This can be achieved through Hierarchical clustering, a method that groups data according to their similarity at different distance levels, giving a large amount of small clusters when the distance level is low, and a smaller amount of bigger clusters when distance level increases. This method is especially useful when various amounts of clusters want te be obtained as segmentation result. Thus, a multilevel hierarchy is built and usually expressed as a tree called dendogram, in which lower level clusters are joined in higher level clusters [3].

Another method that does not require to know the amount of clusters in the image is Watersheds, a clustering algorithm that might be easily understood through an

analogy with real world phenomena. The clusters are built by considering the image as a 3D graph in which there are holes in local minimum; the 3D surface is flooded through these holes forming lakes. Every limit where different lakes are merged is marked as a contour and the final segmentation is formed through the set of contours of the whole image. One of the main limitations of Watersheds method is oversegmentation, because textures and small details generate local minima; to overcome this problem, local imposed minima are used forcing the 3D graph to consider as minima only the places imposed by markers [4].

2. Materials and Methods

Definition of the function: The function to segmentate was developed in Matlab and the syntax used was:

```
function [segmentation] =
    segmentByClustering(rgbImage,
    featureSpace, clusteringMethod,
    numberOfClusters)
```

Where segmentation is a two dimensional output with the labels of the segmentation, rgbImage is the input image to be segmentated, featureSpace is the space in which features want to be extracted, the options available are rgb, hsv, lab or any of these color spaces together with xy coordinates from each pixel. ClusteringMethod used to segmentate can be chosen between k-means, Gaussian Mixture Model, Hierarchical or Watersheds, and the last parameter identifies the number of clusters that want to be obtained.

The function firstly checks whether the number of clusters given as an input is a whole number and is bigger than two, if it does not, then an error message is shown. If number of clusters stated is appropriate the function proceeds to resize the input RGB image to 80% the original size to facilitate computation (although resolution is lost), unless the chosen clustering method is Hierarchical, where the image is resized to 32% the original size because of limitations on computational power, in which a bigger matrix could not be processed with *pdist*.

Afterwards, image is transformed to the color space chosen and the features matrix is formed with the intensity from each of the three color channels and two additional columns if xy coordinates want to be used. It was imperative to normalize each of these chanels, as some of them, for example a and b channels in Lab color space, have values in a range from negative to positive data, but especially because limits for some features are different, hence results would have been affected when considering position features up to a 400 value while color features have maximum of 255 or even 1 depending on the color space chosen. In addition, the xy coordinates features were also normalized in order to avoid interferences due to differences in magnitude order.

Lastly, segmentation is applied according to the method chosen. In k-means method, the only inputs are the number of clusters and the features matrix, where each column is a feature and each row an instance. GMM model works with the same input parameters. In Hierarchical clustering, the first step is to calculate some similarity measure (in our case Euclidean distance), and then create the linkages between closer groups. The labels are assigned according to the maximum number of clusters specified in the function. The implementation of watersheds technique does not take into account the input number of clusters to determine the number of regions (but to impose minima to the image), nor the extraction of features. The gradient was computed for the color image, and then we implemented the minima imposition to avoid over-segmentation. Watersheds was applied to each channel and then averaged to obtain the segmentation result.

The hyperparameter *number of clusters* used was three. This number must be chosen according to the computational power available, because using a bigger k parameter increases significantly the number of calculations required and hence the computational power. The parameter must also be chosen depending on the detail of the segmentations that want to be obtained, therefore, if loads of clusters comprising small details want to be obtained, a higher k should be used; but if only big structures and no details want to be segmented, then a small k should be used to avoid over segmentation, as was the purpose of this paper.

The function described was applied in four images chosen from BSDS500 dataset, using three as the number of cluster parameter and for all the combinations between the six feature spaces and the four clustering methods, hence, 96 results were obtained. To compare them, multiple annotations were available but only one annotation per image was chosen, the one with the least amount of clusters, as our evaluation was aimed at the three biggest objects in each image. Another way to proceed would have been overlapping all the annotations and choosing the contours annotated in all or most of them. To evaluate the results Jaccard index was calculated for each one of the three biggest labeled objects between each segmentation and the corresponding chosen annotation according to their size automatically, without manual revision to check whether they corresponded to the same object. Average was calculated among the three objects evaluated to find the average Jaccard index for each one of the 96 segmentations. Afterwards, to compare the four clustering methods used, average for each image with each clustering method among the six feature spaces was calculated. The procedure was repeated but this time considering only the two biggest objects on each segmentation. Afterwards, average was calculated for each im-

age and each feature representation space, to compare the influence of the color space and the additional spatial information.

3. Results

The four images chosen from BSDS dataset are shown in the following figure.

Figure 1. Images chosen to be segmented.

Annotations chosen for each image are shown in Figure two.

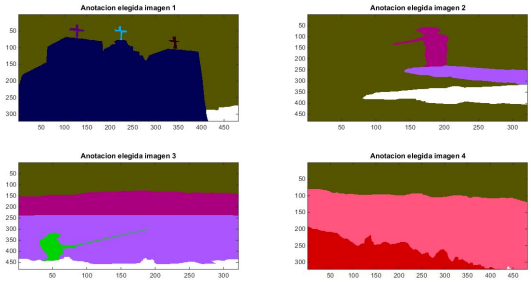

Figure 2. Annotations chosen for each of the four images evaluated.

Segmentation results for each one of the four chosen images are shown at the end of the document, where each row corresponds to the same feature space and the same column to each clustering method.

Additionally, an example of the three biggest objects compared in one image are shown in Fig. 3.

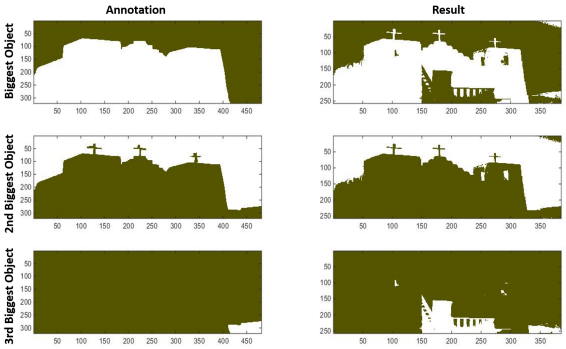

Figure 3. Comparison for each one of the three biggest objects between annotations and segmentation.

The Jaccard index results for each method on each image are shown in the tables below for three and two biggest objects, when automatic comparison between biggest objects in order was applied.

Table 1. Jaccard index results for three biggest objects, when automatic comparison according to bigger clusters was applied. Clustering method comparison.

Clustering Method	Image 1	Image 2	Image 3	Image 4	Method Average
K-means	0.554	0.121	0.205	0.692	0.393
GMM	0.590	0.189	0.503	0.871	0.538
Hierarchical	0.484	0.202	0.128	0.656	0.368
Watersheds	0.063	0.294	0.046	0.404	0.202

Table 2. Jaccard index results for two biggest objects, when automatic comparison according to bigger clusters was applied. Clustering method comparison.

Clustering Method	Image 1	Image 2	Image 3	Image 4	Method Average
K-means	0.801	0.180	0.295	0.742	0.505
GMM	0.836	0.259	0.544	0.881	0.630
Hierarchical	0.656	0.303	0.188	0.694	0.460
Watersheds	0.094	0.422	0.069	0.372	0.240

Comparing the Jaccard Index from the different clustering methods, using all the feature representations in each one, the best result was obtained with the Gaussian Mixture Model and the second best result with k-means, while the worst result obtained was using Watersheds.

In addition, we compared the results for the different features used in the representation space for each image, which are shown in table 3 and 4.

Table 3. Jaccard index results for three biggest objects comparing different feature representation.

Features	Image 1	Image 2	Image 3	Image 4	Feature Average
RGB	0.417	0.199	0.314	0.738	0.417
Lab	0.406	0.230	0.119	0.890	0.411
HSV	0.392	0.184	0.127	0.440	0.286
RGB+XY	0.465	0.250	0.353	0.548	0.404
Lab+XY	0.460	0.162	0.134	0.863	0.405
HSV+XY	0.397	0.184	0.278	0.455	0.382

Table 4. Jaccard index results for two biggest objects comparing different feature representation.

Features	Image 1	Image 2	Image 3	Image 4	Feature Average
RGB	0.602	0.284	0.365	0.679	0.482
Lab	0.565	0.345	0.174	0.921	0.501
HSV	0.565	0.271	0.179	0.496	0.378
RGB+XY	0.654	0.335	0.428	0.536	0.488
Lab+XY	0.631	0.243	0.201	0.903	0.495
HSV+XY	0.570	0.269	0.296	0.498	0.408

When comparing the feature spaces, the best result was obtained for the representation in the Lab color space, and the worst one in the HSV space. Adding the xy coordinates as a feature only increased slightly the Jaccard index for the color spaces RGB and HSV, but decreased in Lab space.

As it was mentioned before, the segmentation methods performed better when only comparing the two biggest objects that were labeled. In average, the GMM yielded better results for all the four images, but the poorest results were obtained for image 2 and 3, even when comparing the different feature representation, which could have reduced the average Jaccard Index. Watersheds technique for segmentation resulted in the lowest indexes. The fourth image was the one with better results for all methods.

4. Discussion

According to the results obtained when comparing the two and three biggest objects within our segmentations, the average Jaccard Index was better when only the two biggest objects were taken into account. As it is seen in Fig. 3, there is no correspondence in the third biggest object between the annotation and result, which decreases the average of Jaccard Index per image.

Using different color spaces to create features is useful to represent pixel information in a different way, because using only coordinates from primary colors RGB might not give the most discriminative features. A clear example is that in real life, luminosity is more significant than color itself, therefore color spaces such as Lab and HSV may give better results than RGB because the information they contain on each channel is sometimes more discriminative in real life images than using only colors.

On average, best color spaces were RGB and Lab and these color spaces plus xy coordinates information. The best clustering methods were GMM and k-means. Lab is an uniform distributed color space, and has a luminosity channel, therefore distances among pixels are representative and thus lead to a better clustering results in methods that use euclidean distance in multidimensional space such as GMM and k-means. RGB is useful when the image to be segmented has clusters with different colors, that is why results in the first and third images have high values unlike the other two images. Adding to these colors spatial information may lead to better results if pixels belonging to the same region are close by, but may be an error source if they are not. On the other hand, k-means and GMM are simple but yet powerful clustering methods because the hyperparameters needed to configure them are few and straight forward. In this case, we knew we wanted three clusters and that was enough information for them to cluster the most similar three regions. With hierarchical and watersheds more parameters are needed because on the first one, we need to know at which distance level of the dendrogram there are three clusters and in the second one, although we imposed three minima that does not guarantee to obtain only three homogeneous regions and is extremely susceptible to textures.

Furthermore, there are some limitations of the method that must be highlighted. The need to resize the images to decrease computational power required to segment, decreases the resolution of the image and may affect results. Additionally, textures are an important error source because clustering methods tend to segment small regions generated by textures that actually belong to the same object. A clear example can be seen on the results of the second image, and in most results obtained by using watersheds method. To avoid this, perhaps filtering would be a good strategy as long as big regions with no details want to be clustered, or to add information about texture. Another strategy could be using a Gaussian Pyramid to make more homogeneous big regions of an image in small scales of the pyramid, making easier their segmentation and removing textures. In addition, different objects that humans can easily segment semantically, may have similar color or features information (such as the boy and the wood logs in the second image), and clustering methods can not possibly differ between them. Besides, sometimes objects that are not among the biggest ones may be the most interesting ones in an image, and therefore automated evaluation through labeled object size would not be useful. An example can be seen on image three in which the fisher is assumed to be part of the sky or of the trees on the background.

Depending on the intensities (and their differences among pixels) of the objects within an image, there could be channels that are more discriminative. For instance, the

channels that contain information about luminosity provide a good discrimination since they reflect changes in intensity. On the other hand, if there are significant differences between objects hue, then H would be discriminative, and if the intensities of the objects are made up of **pure** colors they could be easily distinguished from complex backgrounds within the Saturation channel.

One of the drawbacks of these segmentation methods is that the labels for the pixels that belong to the same cluster/object are changed every time the algorithm is run, so it can not be guaranteed that an specific object is labeled with certain tag. In addition, it depends on the number of clusters or objects that want to be distinguished, the bigger the number of clusters, there would be more partitions in regions for the image. The feature representation just took into account intensities and coordinates at every single pixel, but there are some descriptors such as texture or shape, that must be evaluated within a patch, and help to discriminate objects. In addition, differences in intensities does not imply different objects, and these clustering algorithms tend to group similar instances within the representation space. The poor results with watersheds algorithm are a reflex of the number of partitions made within the image, even though we apply minima imposition to reduce over-segmentation, and our evaluation methodology just took into account the biggest objects. Another limitation of these clustering algorithms is that they are sensible to outliers, in this case for intensities induced by noise or artifacts, and these values are included within the clusters, therefore affecting the distribution of the clusters.

As it was mentioned before, we compared the segmentations for objects based on the number of pixels labeled with the same tag. However, the label with higher frequency in the segmentation result and the one in the respective annotation could not correspond to the same object, so in this case we end up comparing different structures, which explains low values for Jaccard Index. This procedure was performed automatically by comparing the three biggest objects within each image for each segmentation method. Another problem with this metric, is that if the size of the object of interest is not big enough compared to other structures, then it would not be taken into account for the evaluation, such as in the case of the image of the boy standing in front of the sea, where the boy is smaller than the other structures but the results of the segmentation assigned labels to the boy. In addition, in some cases the results of the segmentations contained the same label for different structures that were grouped together by the clustering method. An alternative to improve the evaluation strategy would be to look manually for the labels corresponding to the same object, and in this way there is a guarantee that we are comparing the same object. To deal with the problem of different structures with the same label, one could limit the region where the object of interest is located and ignore the other labels that are far away from this region and not connected to the main object.

Finally, as we mentioned before, filtering or using Gaussian pyramid low levels would be useful to avoid over segmentation caused by textures or small details, which is especially useful when trying to cluster few big regions. Additionally, using some semantic knowledge or annotations of the objects that want to be segmented would certainly improve the results obtained. For Watershed method, marking manually the minimals to be imposed would change drastically the results. To improve Jaccard indexes, a way to match the labels belonging to the same objects would be useful to avoid comparing regions from different structures; similarly, annotations with a given number of clusters would certainly improve the results.

5. Conclusions

- Different clustering methods can be used to address segmentation tasks; each one is more useful on specific data and requires different number of calculations. Hence, computational power available and the difficulty and amount of data used must be taken into account when choosing one method for an specific problem.
- Watersheds is more susceptible to cause errors caused by textures, even though we used minima imposition to avoid over-segmentation.
- Although k-means and GMM are similar clustering methods, k-means gives a hard label to each object while GMM gives a soft label, considering the possibility of a pixel to belong to each cluster. That is why results obtained are slightly better with GMM in all the images evaluated.
- Evaluation for segmentation tasks remains one of the biggest problems to address. First of all, each human annotation is different and to evaluate either a unique annotation must be chosen or the overlap of them must be calculated. Second, in many clustering methods the label assigned to each region changes on each iteration, hence automatic procedures such as considering objects according to their size (used here) or perhaps to their shape must be used to match the regions that are evaluated through Jaccard index, and this provides an important error source if regions compared through Jaccard do not belong to the same object. This could be avoiding through checking manually which label belongs to each object but the task would be extremely time consuming and inefficient.
- Color spaces provide different features and results in segmentation tasks. The best choice depends on the

characteristics of the image to be segmented. Using additional xy coordinates information does not improve the results but does increase computational complexity in all the segmentation methods used.

References

[1] MathWorks. Machine Learning Matlab. Applying Unsupervised Learning.

[2] Scherrer, B. Gaussian Mixture Model Classifiers. 2007. [online] Available at: http://www.medialab.bme.hu/medialabAdmin/uploads/VITMM225/GMMScherrer07.pdf

[3] Hierarchical Clustering. s.f. Mathworks [online] Available at: https://www.mathworks.com/help/stats/hierarchical-clustering.html

[4] Beucher, S. The Watershed transformation. 2010. [online] Available at: http://cmm.ensmp.fr/~beucher/wtshed.html

6. Annexes

Segmentation results for each image with each method and feature space with number of clusters equals three.

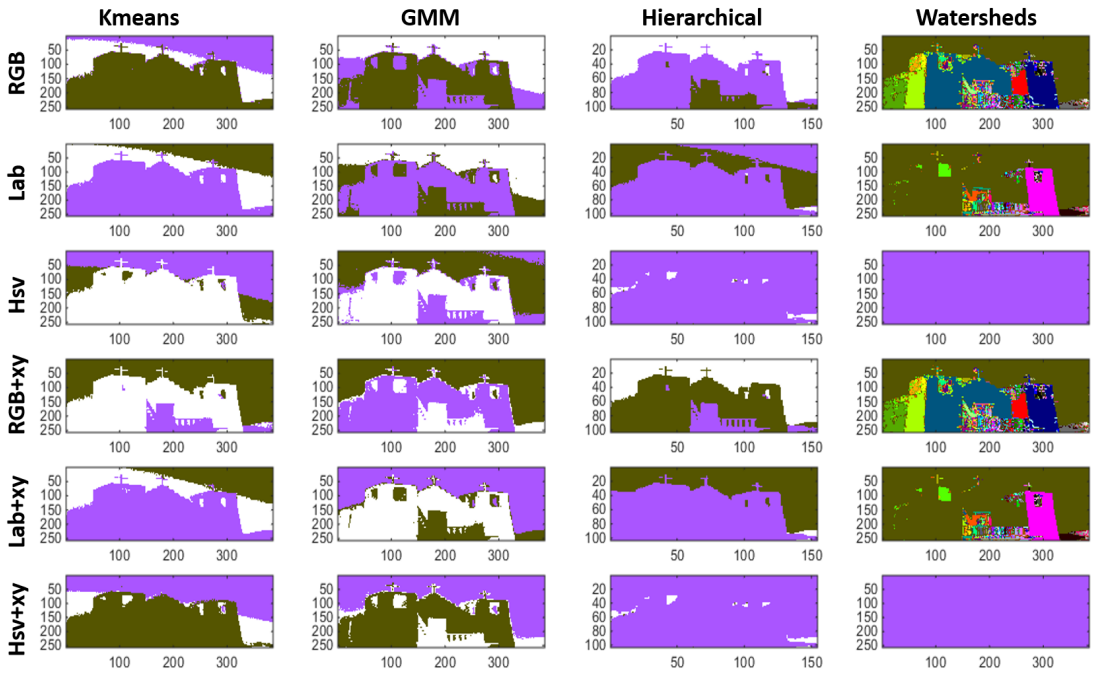

Figure 4. Resultados de segmentación para la imagen 1

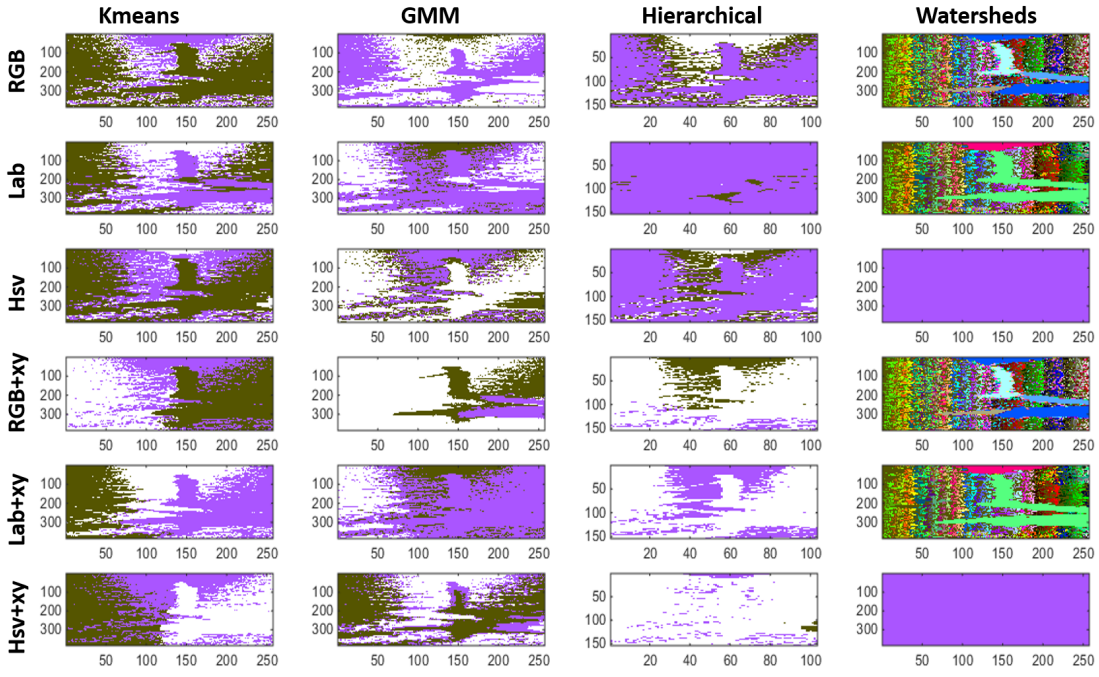

Figure 5. Resultados de segmentación para la imagen 2

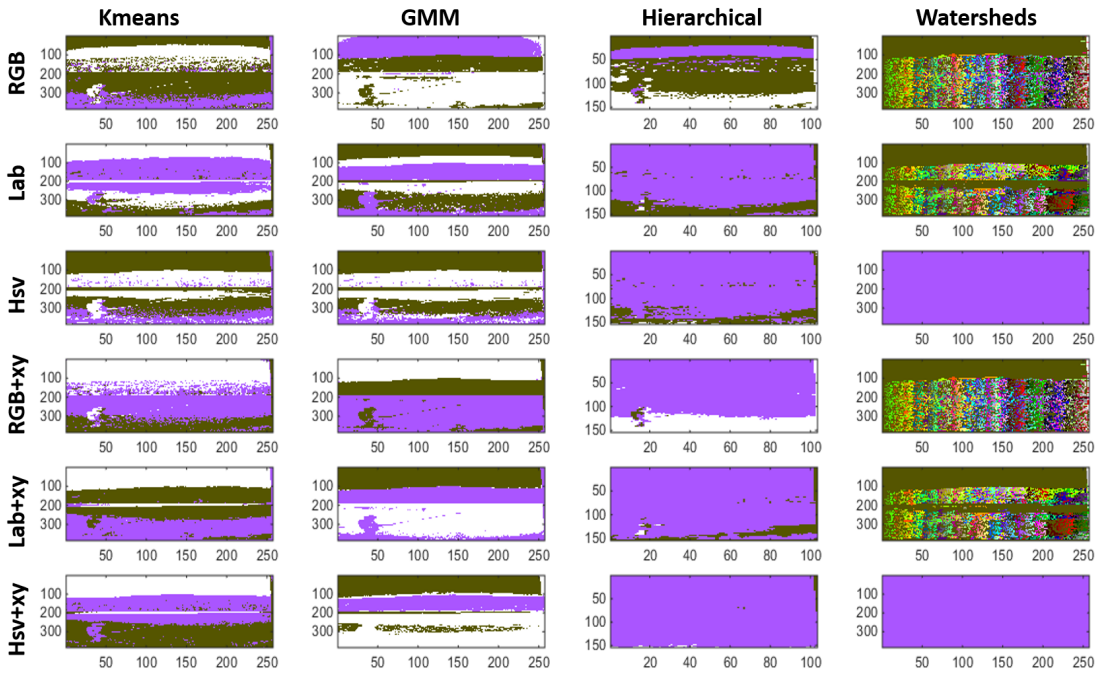

Figure 6. Resultados de segmentación para la imagen 3

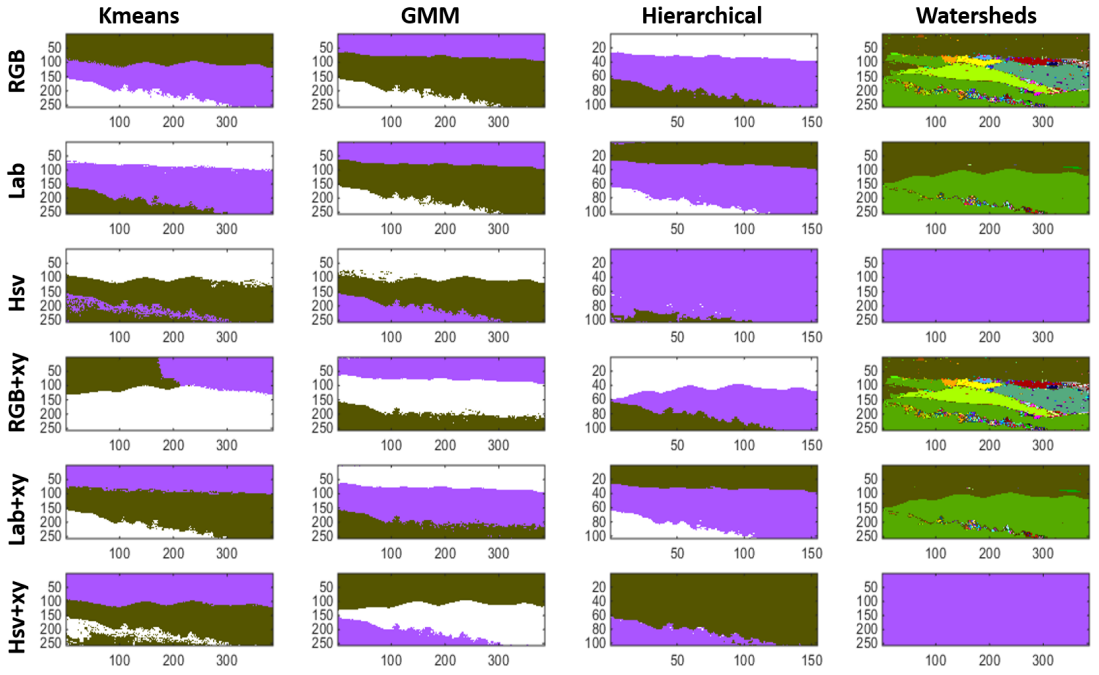

Figure 7. Resultados de segmentación para la imagen 4