Computer Vision IBIO4490 - BSDS and Benchmarks Lab Report

Paula Florez Herrera (Cod. 201115316)

***Abstract*—The following report summarizes the process of development, calibration and testing of a segmentation by clustering function in MATLAB. The methods used were K-Means, Sum of Gaussians, Hierarchical Segmentation and Watershed. Also, the function was able to switch between feature spaces for the segmentation between RGB, L*ab, HSV, RGB+xy, L*ab+xy and HSV+xy. For calibration and testing, the Berkeley Segmentation Data Set (BSDS) and Benchmarks 500 from the Berkeley Computer Vision Group was used. The results show less than satisfactory image segmentations for a low quantity of clusters and a reasonably good recall for the segmentation method Watershed.**

I. Introduction

THE Berkeley Segmentation Data Set, available online for the public [1], provides a set of images and functions that allow anyone to compare the precision and recall of their own segmentation methods with the Berkeley algorithm. This availability provides a common ground for everyone who is currently working on image segmentation and boundary detection, in order to advance the development of the field and benefit from the work that has already been done in the community. The segmentation function about to be developed and tested will, evidently, not produce an efficient result when compared against the Berkeley algorithm, but the results will be shown and analyzed in order to identify what can be changed to improve its efficiency.

II. Methods in the Segmentation Function

The segmentation function developed has the following command:

$$mysegmentation = segmentbyclustering(rgbimage, featurespace, clusteringmethod, numberofclusters)$$

The output argument "mysegmentation" is a matrix containing the cluster number of every pixel in the original image. The input arguments are shown in Table I.

TABLE I
Input Arguments of Segmentation Function

Input Argument	Options
rgbimage	Image to process as a string '*.jpg'
featurespace	Color space:'rgb','hsv','lab','rgb+xy','hsv+xy','lab+xy'
clusteringmethod	Method:'kmeans','gmm','hierarchical','watershed'
numberofclusters	Number of clusters to obtain, type int.

The color spaces are switched using the predetermined functions from MATLAB and the function 'RGB2Lab' available also on the Berkeley Website [2].

The clustering methods used within the function are the following:

K-means: This method creates a given number of clusters in aleatory positions within the image, then, each point of the image is assigned to its nearest Euclidean mean. Afterwards, the positions of each point under a determined cluster number are used to calculate the new mean of the set and this is the location of the new centroid. The process is repeated until the Euclidean distance between each point and its centroid is the minimum possible. [3]

Gaussian Mixture Model (GMM): This method is a probabilistic model in which each point of the image is assigned a probability of being contained into the same cluster as the mean of a Gaussian function. There are as many Gaussians as there are clusters and the soft assignment is modified throughout the algorithm to find the most accurate mix of Gaussians. [3]

Hierarchical: In this method the whole set of points of the image is brought apart or together (depending on the approach) until the image is either completely grouped or completely separated (each pixel is a cluster). This occurs gradually, according to the similitude of the neighboring pixels, and is indexed to fit into a hierarchical tree. The final segmented image can be taken from the tree by selecting the amount of clusters wanted as a result. [3]

Watershed: This is a gray-scale model based on the topography of the image about to be segmented. It is referred to as the flooding model because its principle is that the segments which have the lowest intensity in their division (the ones that are less different between them than the others) are the first ones to be merged if the topographical model of the image were to be slowly submerged into water. [3]

The Figures 2, 3 and 4 are some examples of the results obtained using the function to segment a sample image.

III. Segmentation Function Testing

For the calibration and testing of the function, the BSDS database was also used. Within the test folder there were 200 images selected to test the algorithms and generate a .mat archive for each image. Later, the .mat archives were submitted to the test function in order to obtain a Precision vs. Recall graph that can be used to compare the efficiency of the clustering methods within the function and the function itself against the actual Berkeley data.

Fig. 1. Original Image used for segmentation

Fig. 2. Segmentation using K-Means Method

Fig. 3. Segmentation using GMM Method

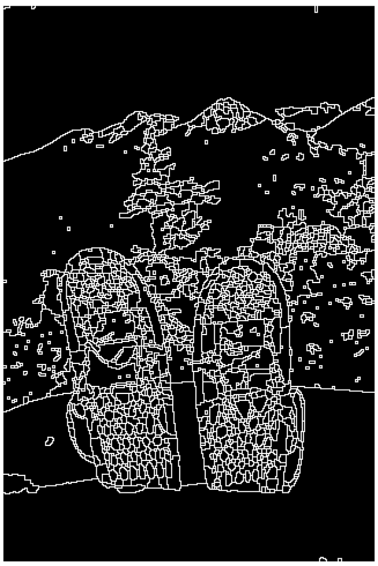

Fig. 4. Segmentation using Watershed Method

IV. GRAPHS AND RESULTS

The chosen color spaces and clustering methods selected were the following:

First, using K-Means and the color space HSV. Second, using Watershed and the color space L*ab and third, using Watershed and the color space RGB. However, due to an unexpected error while creating one of the .mat files with k-means, the testing function didn't allow the progress of the algorithm beyond the 76th image and the first test was unable to be concluded.

The second and third tests were successful, but the results obtained were far from the desired effect of an efficient segmentation model, the results are shown in the Figure 5.

The BSDS testing methodology is based on Precision and

Recall, these concepts refer to the amount of True Positives (meaning the segments that were found and also were tagged as segments in the ground truth) over the total segments obtained, and the amount of True Positives over the amount of actual segments in the ground truth, respectively.

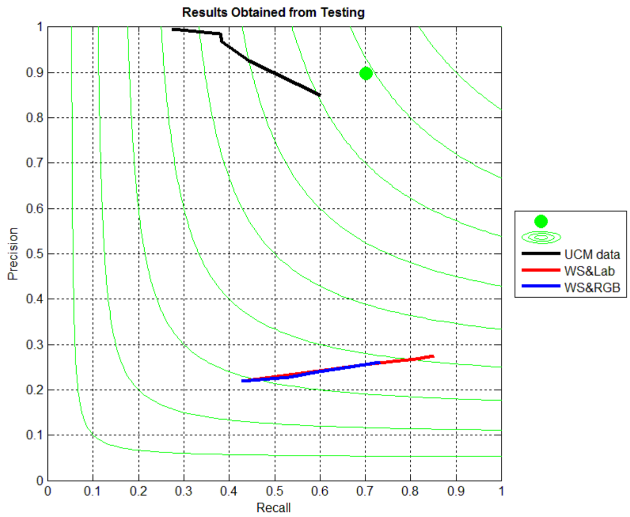

Fig. 5. Final Precision vs. Recall graph containing the results of testing the function

The curves on Figure 5 intend to show the relation between the ability of the method to find majorly real segments on the image and its ability to find the the majority of the segments previously set as the ground truth.

The function developed during this laboratory, as it can be seen in the graph, had a good recall, but sacrificed a lot of precision, proving not to be very efficient. This means that the method tested was able to find most of the segments on the ground truth for every image, but failed when testing the total of segments found, because most were not on the ground truth. This can translate into an over-segmentation, and observing the result on Figure 4 it is evident that Watershed requires a set of maximum clusters otherwise defined in order to produce useful image segmentations.

The areas of the curves obtained by the function were 0.11 and 0.07, while the UCM one is around three times bigger. This is a clear indication that the function needs to be improved.

V. Conclusion

As a possibility to improve the clustering methods' programmed on the function, a better calibration and an actual learning process might be the best solution, given the amount of computational time required for most of these, the time spent on calibration was little when compared to the time required for actual testing.

Given that the main inconvenience when testing was the time spent on correcting (or trying to correct) the resulting .mat files from the k-means first test and still not being able to draw results form it, another important improvement to the function would be to fix the k-means method in order for it not to produce empty clusters (or to correct itself when it does) in order to avoid creating a .mat file of a dimension not comparable to the rest of the tested images.

Finally, the laboratory was useful as a way of understanding how hard it is to obtain satisfactory results in a field as complex as Computer Vision, and also to visualize and compare how far advanced the ongoing investigation of image segmentation and boundary detection actually is, and how invested one must be on its different methods to succeed on contributing significantly to this field of study.

References

[1] Eecs.berkeley.edu,. 'The Berkeley Segmentation Dataset And Benchmark'. N.p., 2015. Web. 20 Mar. 2015.
[2] Eecs.berkeley.edu,. 'Berkeley University'. N.p., 2015. Web. 20 Mar. 2015.
[3] Szeliski, Richard. Computer Vision. London: Springer, 2011. Print.