Segmentation:BSDS

Johana M. Ramirez Borda
Los Andes University
jm.ramirez11@uniandes.edu.co

Francisco A. Rozo Forero
Los Andes University
fa.rozo1843@uniandes.edu.co

Abstract

Contour Detection and Image Segmentation Resources is the name of the project developed by the computer vision group of the University of Berkeley in California. This group provided the BSDS500 data set that is used in this work. So, in order to segmentate this dataset, two methods were proposed: kmeans and gmm. Likewise, the results of the method ucm2 proposed by Arbelaez et al. were provided in order to compare those methods with the last. To do so, a benchmarking code was provided to assess these methods giving three types of results: maximal F-measured calculated with ODS (optimal dataset scale), the same calculated with OIS (optimal image scale) and the AP (average precision). In this sense, GMM and Kmeans had the same AP (0.32), but GMM had a maximal F-measure of 0.64 and an Kmeans had a maximal F-measure of 0.61. Between these two, GMM was the best method. However, the ucm method was far above these with an AP of 0.75 and a F-measure of 0.74.

1. Introduction

Kmeans segmentation

This method is based on the k-means algorithm which is an iterative, least-squares partitioning method that tries to minimize the sum, over all clusters, of distances from each pixel to the center of its cluster. The number of clusters is initially defined. In this sense, the main steps are the following:

- Initialize the number of clusters, K.
- Calculate the mean of each cluster, centroid.
- Compute the distance between each pixel and the centroid of each cluster.
- Assign each point to its nearest cluster.

The main idea is to iterate over these steps until the sum of the squared within group errors, the distance of the pixel to its centroid, cannot be lowered any more [9],[13].

In this project, this method was selected because it is one of the most used clustering algorithms, because of its simplicity and fast computation [9]. Moreover, this method allows the variation of many parameters such as the distance metric, including squared Euclidean, cityblock, cosine, correlation, hamming, etc. More important, the kmeans matlab function allows to use the replicate option, that sets the number of times to repeat clustering using new centroid positions. It has been found that it is a good practice to use this option because it searches for lower, local minima and the function output returns the solution of the replicate with the lowest total sum of distances [12]. In this sense, one of the main disadvantages of this method, that is the results of k-means are highly dependent on the initial and subjective selection of centroids, was a little reduced. On the other hand, the method of hierarchical segmentation, which also groups by distance, had a similar performance than k-means. However, the hierarchical method requires a large amount of time to process a single image because it starts with an over-segmentation, it means segmenting almost each pixel as a different own region [18]. Hence, due to the similar performance, the way found to improve the k-means disadvantage and the difference in time spent, the k-means algorithm was selected.

Finally, it was also found that both k-means and hierarchical algorithms had one of the best results when using the hsv+xy colorspace. This was noticed because the segmentation separated successfully different objects and/or regions, without separating many details within them. The opposite case happened with the watershed algorithm which, even though the minima were imposed with the gradient of the image, the resulted image in all spaces was over segmented.

The selection of the hsv+xy colorspace was firstly based on the experimental results obtained previusly and second, on the related literature. It is well known that hsv is one of the most frequently colorspace used when segmentating [7], for example, in [19] Bora et al. extracted features from this colorspace and demonstrated that the segmentation gave better identification of objects compared to that obtained when using RGB colorspace. Moreover, [11] and [16] proposed two different segmentation methods in which the first step was to convert the image from RGB to HSV. The first one then used Otsu's multi-thresholding on the *V* channel in order to get the best threshold and then used k-means. They reported that their results (values of MSE) were quite satisfactory [11].

GMM segmentation

The second method implemented was the Gaussian mixture model (GMM). This method is an example of Model-based clustering and its main idea is to represent each cluster with a Gaussian, or parametric, distribution. In this sense, each cluster is represented by the probability density function of the observed variables as a mixture of multivariate normal densties [1]. So, the main goal of this method is to find the parameters of a mixture of Gaussians that fit the model and explain the data. To do so, this method uses an algorithm called: *Expectation Maximization*, which in the first step, estimates the responsabilities (soft assigment) and then estimates the parameters given those responsabilites[1].

We decided to use this method, firstly because its performance followed hierarchical segmentation, and second, because it allows distance to adapt to the data distribution and assigns it with different weights or probabilities (soft assignment) contrary to k means (hard assignment). Moreover, it is a method that has been applied before, for example, [15] adopted the GMM to represent the probability distribution of their features and performed clustering using this model.

Finally, it was found that the best result within this method (GMM) was obtained when using the lab+xy colorspace. This may be due to the fact that it has been found in literature that, for color clustering, it is desirable that the selected color features define a uniform color space [6]. In this sense, CIELab is a perceptually uniform color model that has showed to be appropiate for color images with various types of noises [6] and it closely matches with the human perception and gives better results [17].

In both cases it was used the (x,y) coordinates of each pixel so the clustering was made based on (hsv+xy) for KNN and (lab+xy) for GMM. The main reason was that using these coordinates the results were better than when using just hsv or lab. This may happen because by using them, the spatial coherence is enforced and the clustering is made with not only similar, but near-by pixels [1]. In addition, both methods have as hyper-parameter the number of clusters, which has to be introduced manually in the algorithm.

In this project, the Berkeley Segmentation Dataset (BSDS500) was used in order to test two segmentation methods such as K-means and GMM. Then, the benchmarking code provided was used to compare them with the, also provided, results of the UCM2 method [4].

2. Materials and methods

For this lab, the software implemented was MATLAB R2016b.

2.1. Dataset

The computer vision group of the University of Berkeley in California developed a project called ”Contour Detection and Image Segmentation Resources”. The main objective of this project was to provide an empirical basis for research on image segmentation and boundary detection, promoting scientific progress in the study of visual grouping [14],[20].

The dataset BSDS500 consists of 500 natural images that contain landscapes, animals alone, in groups, camouflaged with the landscape or with humans, also humans alone or in groups and camouflaged by animals or the landscape. Some examples are shown in Figure 1.

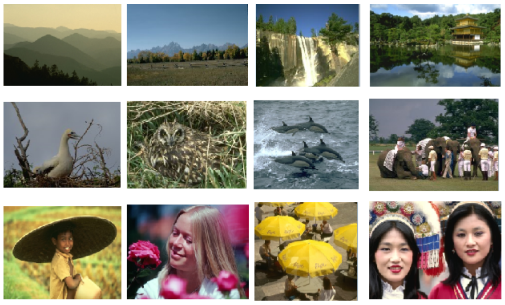


Figure 1. Examples of images found in BSDS500 database.

Moreover, there is one *.mat file* with 5 human annotations serving as ground-truth, for each image. Each human annotation contains one binary matrix representing the boundaries and other matrix containing the colored segmented image. One example is shown in Figure 2.

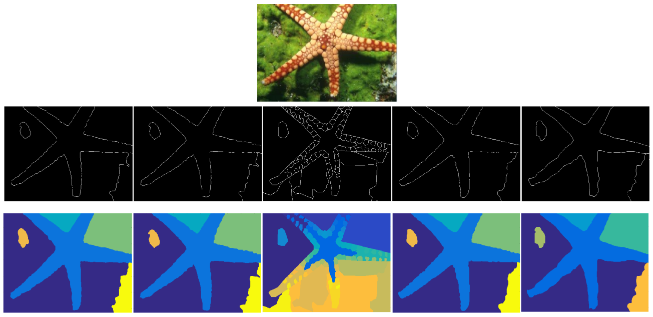


Figure 2. Example of 5 human annotations for one image: boundaries and segmentetation. This annotations were found for all images.

Like these, there are 500 images (481x321 pixels each)and they are separated into train (200), test (200) and validation (100) [14][20].

Besides the groundtruth annotations, this dataset contains a benchmark that is provided in order to compare different segmentation and boundary detection algorithms. This was provided in the form of a *benchmarking code* [4] that is going to be used in this project to assess quantitatively two segmentation methods: K-means and GMM, and then compare them with UCM2. The results for the K-means and GMM were calculated by us and the results for UCM2, a methodology proposed in [3] by Arbelaez et al., are already given within the benchmarks. In this sense, the benchmarking code provides scores and curves such as the precision-recall curve, which is used to describe the performance of the algorithm. However, even though this curve is a rich descriptor of performance, it is used the maximal F-measure value too, as a descriptor that assesses the performance into a single number [14],[20].

In short, kmeans and gmm algorithms were used to segmentate just once each set: train, val and test. Both kmeans and gmm were calculated using 16 values of *k*, from 2 to 17 because we assumed that no image will have more than 17 objects to segmentate. Therefore, 16 matrices were saved in a *.mat* file for each image. This process was repeated for all images within each set separately (train, val and test).

2.2. Evaluation

After segmentating just once each set of images with both classifiers and saving the results, the benchmarking code was used to evaluate performance by comparing the 17 segmented images of our methods with the 5 grountruth images contained in the benchmarks. This benchmark matchs the machine-detected results with the human annotations to define true positives (TP), falses positives (FP) and misses (MS). This process is repeated for each image and for all the human segmentations. Then, it aggregates TP, FP and MS across images and defines the point corresponding to the overall precision and recall. This process is repeated for multiple thresholds and the precision-recall curve is produced. Then, the algorithms are compared using the maximal F-measure [1]. In the precision-recall curve, the precision is the probability that a predicted response is a true positive, also, is a measure of how much noise is in the output; recall is the probability that a true positive of the groundtruth is detected. Moreover, the precision-recall curve tells us about the trade-off between misses and false positives for different threshold. On the other hand, an ”ideal” result can be defined in different ways. First, it could be defined as a curve which maximum F-measure is 1, which is usually unlikely. Second, it could be defined as the curve which maximum F-measure is the one produced by human [14],[20].

3. Results and discussion

The benchmark code was successfully runned three times, one for each set of images (train, val and test), for both methods. Each time, the code compared our results with the annotations as explained before.

Maximal F-measure calculated with ODS and OIS and Average Precision

In Table 1, 2 and 3 are shown the results obtained when evaluating the GMM, k-means and UCM2 methods respectively:

Table 1. Results for GMM segmentation on test and val sets.

GMM			
TEST		**VAL**	
F with ODS	0.58	**F with ODS**	0.56
F with OIS	0.64	**F with OIS**	0.62
Area	0.32	**Area**	0.30

Table 2. Results for Kmeans segmentation on test and val sets.

KMEANS			
TEST		**VAL**	
F with ODS	0.56	**F with ODS**	0.55
F with OIS	0.61	**F with OIS**	0.59
Area	0.32	**Area**	0.31

Table 3. Results for ucm2 segmentation on test and val sets.

UCM2			
TEST		**VAL**	
F with ODS	0.74	**F with ODS**	0.73
F with OIS	0.77	**F with OIS**	0.76
Area	0.75	**Area**	0.74

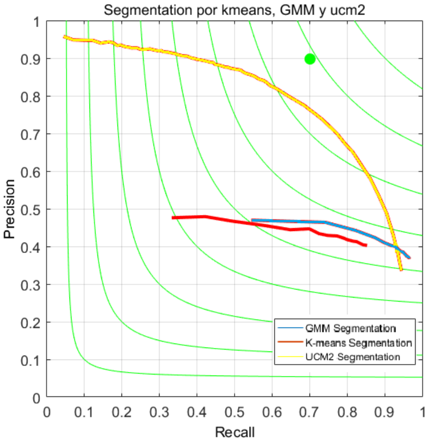
Based on these tables, it is possible to compare the three methods using the AP (average precision) and the maximal F-measure calculated with ODS and OIS:

First, as observed in Table 1, 2 and 3, the maximal F-measure calculated with OIS is always greater than the one calculated with ODS. This may be because when calculating the F-measure with ODS it is being used the *optimal dataset scale*, it means that just one scale (k) es being chosen for all the images in the dataset. On the contrary, when calculating the F-measure with OIS, it is being used the *optimal image scale*, it means that is being used the scale (k) that best segmentates each image. With this in mind, it is understandable that if each image is being segmented by its best k, the results would be better than those obtained with a generic one. Arbelaez et al. [4] found this similar pattern and affirmed that with OIS, one naturally obtains better segmentations.

Second, in these tables is reported the AP (average precision), which represents the area under the precision-recall curve. As observed in Table 1 and Table 2, the methods of gmm and kmeans obtained a very similar AP. Moreover, these two areas were far below the area of the ucm2 method. This means that both kmeans and gmm present low precision (high rate of false positives) and low recall (high false negative rate). On the contrary, ucm2 presents high scores for both, meaning that is returning more accurate results.

Precision-recall curve

In the Figure 3. can be observed the precision-recall curve that compares the three methods.



,

Figure 3. Precision-recall curve of the three methods.

In the first place, lookin at Figure 3. and Tables 1 and 2, although both knn and gmm had basically the same AP measure, the values of ODS and OIS differed more. As observed in Table 1 and Table 2, both ODS and OIS were greater in the gmm method than those of kmeans. The maximal F-measure gives information about both precision and recall. For that reason, it is possible to say that, even though the difference is not very large, the gmm method gave better segmentations than kmeans. In the first place, it is well known that k-means is only appicable if the mean is defined, moreover it needs the determination of the value of k and it is very sensible to outliers. It also only join clusters with spheric forms. On the ther hand, GMM can be seen as the generalization of kmeans. This method forms clusters by representing the probability function of of a pixel to belong o a certain combination of non-parametric distributions. It is said that gmm method helps to express the uncertainty that kmeans doesn't allow to [5],[2].

Second, neither of the two methods implemented didn't beat the Pablo's UCM method and were far behind its performance. This can be seen in Figure 3. (and Table 1,2 and 3) because the curve of the ucm method (yellow) was far above the other two (red and blue). This implies a bigger area under the curve and a bigger maximal F-measure, which, as explained before, means higher precision an higher recall. Kmeans and GMM were of the first basic methods implemented for segmentation and they group pixels based only on their similarity. On the other side, the ucm method finds boundaries in the image based on their color, brightness and texture. Additionally, it builds a graph based on the lines present in the image and their different magnitude and angle and then uses normalized cuts based on the eigenvectors of the graph; which is a greater processing to the image than just reorganizing pixels based on distances. For that reason, looking at the results and the background of each method, the one recomended is ucm2.

With the aim to illustrate the performance of the three methods, in Figures 4 and 5 there are the groundtruth, gmm segmentation, kmeans segmentation and ucm2 result provided.

,

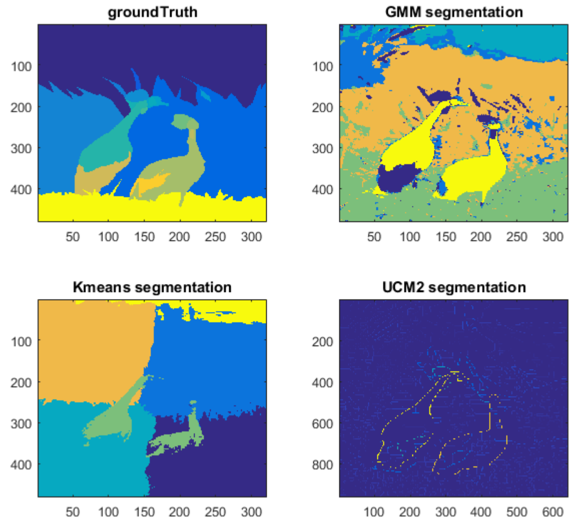


Figure 4. Example of segmented image by gmm, kmeans and ucm2, and its groundtruth. The *k* used was 6.

In this figure, it can be seen what was expected based on the Tables and the Figure 3. That is, the performance of the kmeans alhorithm is lower than gmm regarding to the groundtruth. Also, it is possible to see what we mention about the lack of spatial information in this method: even though there are regions that belong to the same category or object (floor, background, etc), kmeans clusters them as different. On the contrary, gmm segmentated, at least, these regions as similars. However, it is also possible to see that in gmm result there is a little over-segmentation regarding the groundtruth. Finally, ucm2 result borders almost exactly each animal and region.

,

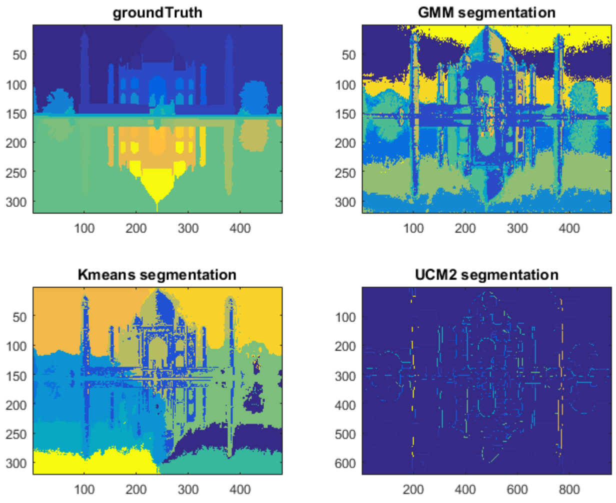


Figure 5. Example of segmented image by gmm, kmeans and ucm2, and its groundtruth. The *k* used was 11.

As seen in Figure 4, Figure 5. illustrates the lack of spatial information in the kmeans segmentation. However, now not only kmeans but gmm present a little over-segmentation regarding to the groundtruth.

3.1. Limitations

Kmeans algorithm is fast, processing 2.76 images per minute, while gmm spent 1.2 minutes for two images with a processor of 8GB RAM. Therefore, both of them are memory intensive [1]. Also, both methods rely only on the pixel (intensity) information regarding position, colorspace values, distance measurement method and σ for kmeans and gmm respectively. Pixel information ignores boundaries and edges that separate objects. Moreover, it ignores information given by textures, boundaries, lines orientations,local minimum or maximums. Others limitations about hyper-parameters were explained before.

On the other hand, when the pixels characteristics are not normalized, the relevance of the information with low values is meaningless next to those with high values (i.e. L channel of Lab vs xy position). Also, not every image responds the same way to the different colorspaces. For example, shadows and reflections in images tend to become segmentated objects in the image.

3.2. Future work

- One way to improve the segmentation is to increase the information that is extracted from the pixels. This can be done by combining the information provided by more than one colorspace as well as the position. For example, [10] and [8] realized that the usual is to adopt one space (hsv,rgb, etc). However, they changed that idea and selected more than one colorspace, finding that it helps in better differentiate between pixels.
- Combine methods, such as adding information of watersheds to the one provided by kmeans or even adding mean shift. For example, [15] added mean shift to determine automatically the number of mixture components to use in gmm.
- Detect border or magnitud of images and then cluster pixels boundred by them.

4. Conclusions

It was possible to segmentate the BSDS500 dataset using the kmeans and gmm segmentaton methods within hsv+xy and lab+xy colorsapces. Moreover, we successfully ran the benchmarking code in order to compare these two methods with the provided results of the ucm2 method.

Firstly, although Kmeans presented one of the best segmentations when testing it with a small set of images (same as herarchical), the results in the whole BSDS500 database showed that GMM was the best method from the chosen ones. However, while GMM had a better performance than Kmeans, both of them were far below ucm2 which presented the highest AP (0.75) and maximal F-measure (0.74). We consider that this can be explained by their low processing, high memory processing and the type of information they extract from the images (just intensity). So, in order to improve the results, one method is not enough to achieve a good performance and the combination of many methods, as well as many colorspaces, would make the difference.

References

[1] P. Arbelaez. Lecture 5: Clustering. In *Computer Vision*.

[2] P. Arbelaez. Lecture 5: Clustering. In *Computer Vision IBIO4680*.

[3] P. Arbelaez. Boundary extraction in natural images using ultrametric contour maps. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 182–182. IEEE, 2006.

[4] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011.

[5] U. Baid, S. Talbar, and S. Talbar. Comparative study of k-means, gaussian mixture model, fuzzy c-means algorithms for brain tumor segmentation. 2017.

[6] S. Bansal and D. Aggarwal. Color image segmentation using cielab color space using ant colony optimization. *International Journal of Computer Applications*, 29(9):28–34, 2011.

[7] D. J. Bora, A. K. Gupta, and F. A. Khan. Comparing the performance of l* a* b* and hsv color spaces with respect to color image segmentation. *arXiv preprint arXiv:1506.01472*, 2015.

[8] D. J. Bora, A. K. Gupta, and F. A. Khan. An efficient approach toward color image segmentation with combined effort of soft clustering and region based techniques using l channel of lab color space. *International Journal of Computer Applications*, 128(12), 2015.

[9] N. Dhanachandra and Y. J. Chanu. A new approach of image segmentation method using k-means and kernel based subtractive clustering methods. *International Journal of Applied Engineering Research*, 12(20):10458–10464, 2017.

[10] D. Girish, V. Singh, and A. Ralescu. Extended pixel representation for image segmentation. 2016.

[11] V. Jumb, M. Sohani, and A. Shrivas. Color image segmentation using k-means clustering and otsus adaptive thresholding. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 3(9):72–76, 2014.

[12] Mathworks. k-means clustering, jun.

[13] S. of Computing: University of UTAH. Segmentation using k-means algorithm, jun 2002.

[14] C. F. Pablo Arbelaez and D. Martin. The berkeley segmentation dataset and benchmark, jun 2007.

[15] J. H. Park, G. S. Lee, and S. Y. Park. Color image segmentation using adaptive mean shift and statistical model-based methods. *Computers & Mathematics with Applications*, 57(6):970–980, 2009.

[16] A. K. Randhawa and R. Mahajan. An improved approach towards image segmentation using mean shift and felicm. *International Journal of Advanced Research in Computer Science*, 2014.

[17] S. Sigurdsson, L. K. Hansen, and K. T. Drzewiecki. Color segmentation of skin lesions with the generalizable gaussian mixture model. Technical report, IMM Technical report 2003-22, 2003.

[18] S. software. Hierarchical segmentation, jun 2010.

[19] S. Sural, G. Qian, and S. Pramanik. Segmentation and histogram generation using the hsv color space for image retrieval. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 2, pages II–II. IEEE, 2002.

[20] C. vision group. Contour detection and image segmentation resources, jun.

All the codes created and used are attached.