Lab 06 - Segmentation

Jhony A. Mejia
Universidad de Los Andes
Biomedical Engineering Department
ja.mejia12@uniandes.edu.co

Abstract

Segmentation through clustering is a useful tool. The parameters changed for obtaining different segmentations were number of clusters, feature space, method of clustering and evaluation method. The evaluation method was a variation of Jaccard's index focused on borders. The best result taking into account time and Jaccard's index was obtained with HSV channel, K-means, 10 clusters, default threshold for canny border detection and 10 pixels of distance between the predicted border and the ground truth. Better practices of pre-processing are suggested for obtaining better results.

1. Introduction

There are different methods for clustering data. One of the most used methods is k-means. K-means is a process of optimization of groups. The idea is to make that the data belonging to a cluster is more similar with data within the cluster than outside it. The most commonly distance used in k-means is the euclidean distance. The clusters are 'hard assignations' (which means, each cluster has a label). GMM is a variation of k-means, in which there are 'soft assignments' (instead of labels, there are possibilities of belonging to a cluster or another).

Hierarchical clustering is useful for seeing multi-scale relations between clusters. This method (the agglomerative one) begins with one point and starts joining together points that are close to each other. This type of clustering is usually represented in a tree or dendrogram.

Watersheds is a segmentation method that works through flooding. One can think of watersheds as a tub with holes in its base. AS one push the tub into the water, the tub starts getting full of water. Those holes are called minimums, and the barriers that diverges the water lakes are the watersheds. [1]

All of the previously mentioned methods can be used in segmentation through clustering. The features that are presented to the methods have a wide variety. One of the most common features are color (and its different color spaces representation) and space information (distance between points). For obtaining an optimal segmentation one must find the best combination of features, method and methodology of evaluation.

2. Materials and methods

The procedure can be divided in two parts: segmentation and evaluation. The first process was mostly developed with matlab built functions. The segmentation process took into account several factors as color space (RGB, Lab and HSV), method of clustering (kmeans, gmm, hierarchical agglomerative and watershed) and number of clusters. The second process was based on a variation of contour detection.

2.1. Segmentation

Three color spaces were taken into account (RGB, Lab and HSV). These were the descriptors of each image of the database. Additionally, each color space had a variation, in which the position of the pixels (X and Y) were added as descriptors. The change of color spaces was done by the functions 'rgb2lab.m' and 'rgb2hsv.m' from Matlab. The position descriptors were built manually, taking as (1,1) the upper left corner. After having the descriptors (in matrix representation) a change of dimensions was developed for passing from an image (mxnx3) to a set of descriptors (mnx3 or 5). Both of this procceses were developed in 'featChange.m'.

After having the descriptors, four different methods were used for clustering the mentioned descriptors. The methods used were k-means, gmm, hierarchical agglomerative clustering and watershed. It is important to say that the factor that defined the number of sub-segmentations was the number of clusters. In this occasion there were used 5, 10, 20 and 50 clusters.

For clustering by kmeans the process was pretty straight forward. The Matlab function 'kmeans' was used, using as inputs the descriptors and the number of clusters.

For clustering by gmm one didn't know how to deal with the probabilities given (soft assignments). These soft probabilities were obtained by using 'fitgmdist'. This function received as inputs the descriptor and the number of clusters. After obtaining the probabilities, 'hard assignments' were assigned to the data by using the Matlab function 'cluster'. Some problems of ill-conditions were produced, so a 'RegularizationValue' of 0.1 was defined for solving that issue.

For clustering by hierarchical agglomerative clustering, a resolution reduction was performed. The reduction was performed using 'impyramid' Matlab's function. The image was downsampled to half of its original dimensions (using Gaussian Pyramid one level below). After the downsampling, a hierarchical agglomerative tree was built using 'linkage' and the distance 'ward'. Finally, the tree was cut for obtaining n clusters using the Matlab function 'cluster'.

For clustering by watershed, the gradients of each channel was obtained using a sobel filter. After that, each channel gradient was added up and saved in a new total gradient. Next, the 'watershed' function was used with the total gradient as input, and a changing number of h-minimums. De h-minimums started in 1, and started to grow into 10 until obtaining the number of clusters desired. The process of growing h-minimums and comparing the number of clusters was done iteratively.

All of the clustering methods processes can be seen in detail in 'useCluster.m'. Both codes ('featChange.m' and 'useCluster.m') where put together in 'segmentByClustering.m' for obtaining a segmentation based on the number of clusters.

2.2. Evaluation

The evaluation method was based on boundaries. The boundaries for evaluation were obtained using Canny's edge detector on the segmented image given by 'segmentByClustering.m'. Canny's edges were generated using the Matlab function 'edge'. This function has an optional parameter for thresholding the border detection. Three threshold levels were taken into account for obtaining the borders: 'Default', 0.1, 0.5 and 0.9. This process was done in 'boundOfSeg.m'.

Each image had multiple ground truth for boundaries and segmentations. In this case, a border was considered a border if at least 2 people marked it as a border. In other words, a multiple intersection was constructed for obtaining a unique ground truth annotation. This process was only done for the boundaries (as that is our evaluation metric) in 'unifyGT.m'.

For evaluating the similarity between boundaries obtained and its corresponding ground truth, the index of Jaccard was used. One is aware that Jaccard is used for segmentation problems, and that for boundary detection one should use RP curves, but the index of Jaccard seemed much easier to implement. However, the index of Jaccard isn't the traditional one. What one did was considering a threshold of distance (in pixels) for saying if the border detected corresponded to the one of the ground truth. This distance was determined by dilating the ground truth with a disk of 'd' radius. After dilating, an intersection (the denominator of Jaccard's index) was developed between the borders obtained and the ground truth dilated. The union (numerator of Jaccard's index) was defined as the 'sum' of the original ground truth annotation and the original borders obtained.

The radius 'd' for considering if two borders correspond, was a parameter varied between 1, 5 and 10 pixels. Most of the evaluation process can be seen in 'evalData.m'.

Finally, all the process was put together in 'runAll.m' (for segmenting the full database with given parameters). An optimized version of 'runAll.m' can be seen in 'runO.m' (this was the code used for obtaining all the test data). An easier to visualize and understand script is shown in 'Demo.m'.

3. Results

There was no special scale procedure for any of the channels. One considered that the method would work with raw information. The only rescale process was performed in the hierarchical method, and it had to be done because of lack of memory. The rest of the methods were done with the original image's resolution. One didn't want to downscale the images because a lot of border information can be lost in this process. However, downsampling could be useful for removing textures, and therefore, reducing false contours.

Additionally, the hierarchical method had to have an special parameter for producing decent results. The default distance (single) produced undesirable segmentations 1. That is why this distance had to be changed. The distances median and centroid had problems of non-monocromic clustering. THe most used distance is 'ward', and this distance was the one used for the hierarchical method.

The multiple ground truth problem was solved by considering that if one border was selected by more than one person, it must be a real border. Although this is a very strong affirmation, it can be useful for reducing computer cost (instead of comparing each annotation with each result). However, one knows that the most appropriate way for dealing with multiple annotations is developing a RP curve, and determining its AP or F-value.

The evaluation method was selected because of its ease of implementation. One knows that dealing with borders problems can be easier than dealing with segmentation problems. The main factor for the decision for using borders instead of segmentations was the problem of indexing between annotations and data. The main trouble was that the indexes will probably not match, so an additional process of connected components extraction had to be performed. By choosing borders one reduces the possibilities to a binary problem (is or isn't a boundary).

The determination of the correspondence of borders was also chosen because of its ease of implementation. The modified index of Jaccard provided a metric of how well the segmentation process was developed. Anyway, one doesn't know how valid it is to evaluate a border detection problem as a segmentation problem.

Surprisingly, the fastest method was watershed. However, watershed had the problem of being unable to directly take into account spatial (location) information. This means that the results obtained in RGB, Lab or HSV and RGB+XY, LAB+XY and HSV+XY are respectively the same.

The second fastest method was k-means. However, this method was the one that showed more number of fails of convergence. It is important to say that the maximum of iterations was left in 100, in order to keep the computational cost low.

The third fastest method was hierarchical agglomerative clustering. Despite of being of a lower resolution (half of the rest), this method was almost the slowest. This fact is understandable as the whole tree of linkages must be built before defining a number of clusters. Taking this into account, hierarchical clustering can be useful for considering different number of clusters, but the cost of building the tree is high (compared with the other methods). It is important to say that this method showed warnings of possible misconstruction of the tree (for example in Lab color space). The causes of this warning is unknown. Possibly a rescale of this channel could solve the issue.

The slowest method was gmm. It is important to say that this method didn't run until the 'RegularizationValue' was defined. One doesn't know what that means, but it works for preventing errors. However, it is important to say that HSV channel was surprisingly fast, and it showed none failures of convergence. This suggests that a better rescale of the data can be useful for improving this method performance. The hypothesis is that this works well because HSV can be thought as a cone, and gmm basis is the supposition of gaussian distribution of the data. Roughly, a gaussian distribution can be merely described by a cone upside down. In the other cases the method takes very long and tends to fail for convergence because the data isn't similar to a gaussian distribution.

The best results are shown in Tables shown in Figs. 2 and 3. As one can see in the table, the best averages of Jaccard Indexes are shown in gmm. However, this method was extremely slow (except for HSV). In our consideration, the excess of time is not worth it. That is why, the two best results are obtained in the HSV channel with K-Means, 10 clusters, default canny edge detector and 10 pixels of distance between ground truth and data (0.3384); and in the RGB+XY channel with 20 clusters, default canny edge detector and 10 pixels of distance. The full data obtained for each image can be found in .mats in the folder 'Results'.

As one could have expected, the best results were obtained in the maximum distance between the predicted border and its annotation (10 pixels). However, 10 pixels is somehow moderate and can be accepted as a measure of similarity between ground truth and predictions.

An interesting thing is that the worse performance is alway shown in hierarchical clustering. This can be the result of two things. The first one is the downsampling that needed to be done. The second one is the lack of data treatment (scale). Maybe with a pre-processing state this method's performance can be improved.

Another interesting thing is that the number of clusters do not tend to follow a rule. The estimation of the best number of clusters seems to be empirical (trough try and failure). It is fair to note that an increase in the number of clusters does not guarantee an improvement in the performance of a given method. In some cases, an excessive number of clusters can worsen the results obtained.

Watershed seem to have a better performance with an increased number of clusters. An explanation of this is the over-segmentation that this method tend to generate. AS there are many regions, an increase of the number of clusters can diminish the loss of regions and improve the method's performance.

The best two results segmentation and boundaries can be seen in Figs. 4, 5, 6 and 7 respectively. As one can see, in the best result there are a lot of false positives. However, the perception of segmentation is pretty good in the best result. In the second results (RGB+XY) there are less false positives, disregarding a higher number of clusters. The causes of this results are unknown. However, in RGB+XY one can see smaller segmentations. The previous can be explained because the algorithm tends to put close objects together.

This algorithm can be improved by making a preprocessing phase, for scaling the data. Another improvement can be to put the (0,0)in the XY coordinates in the center of the image, not in the top left corner. This based on that the most important information is located in the center of the image, not in the top left corner. However, we chose the top left corner because of its ease of implementation.

4. Conclusions

Segmentation problems can be solved through different parameters, as number of clusters, method of clustering and method of evaluation. There is no manual for the perfect number of clusters. The stimation of the best number of clusters have to be done empirically.

Different results can be obtained through diverse representations of an image. Distance and color spaces are common features and are easy to use for segmentation purposes.

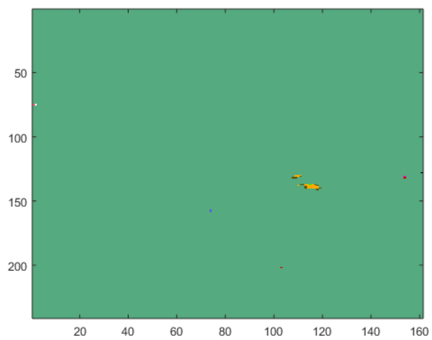

Figure 1. Hierarchical clustering with single distance.

Each clustering method has its own vantages and disadvantages. One must know how each one of the methods works in order to prepare the data for obtaining better results. K-means should have good initial seeds and easy to separate regions. Gmm should have a gaussian-like distribution of data. Hierarchical clustering's preferences is not known by this author. Watershed data should have clear gradients and ideally should have markers as minimums.

Hierarchical performance can be improved by building a divisive algorithm instead of a agglomerative one. The divisive alternative has the possibility of having access to all the data at the same time, while the agglomerative method only sees part of it. However, divisive algorithms are way slower, so that is why it wasn't used in this article.

Segmentation through clustering is a relatively fast method, and easy to implement. Probably the most challenging part is to optimize the parameters for obtaining better results. Also, it is important to say that this methods are highly sensitive to the initial seeds (k-means and gmm).

References

[1] Arbelaez, P. *Lecture 5: Clustering*. Universidad de Los Andes. 2018.

5. Images and Tables

Color space	Method	Clusters	distT	cannyT	Best Avg
RGB	kmeans	5	10	0.1	0.3317
	gmm	5	10	D	0.345
	hierarchical	20	10	0.9	0.0134
	watershed	50	10	0.5	0.1364
Lab	kmeans	5	10	D	0.336
	gmm	5	10	D	0.3529
	hierarchical	10	10	0.9	0.0118
	watershed	50	10	0.5	0.1997
HSV	kmeans	10	10	D	0.3384
	gmm	5	10	0.1	0.3189
	hierarchical	5	10	0.9	0.0095
	watershed	-	-	-	-

Figure 2. Best results for each color space (RGB, Lab and HSV) with the given parameters.

Color space	Method	Clusters	distT	cannyT	Best Avg
RGB+XY	kmeans	20	10	D	0.3389
	gmm	10	10	0.1	0.3644
	hierarchical	5	10	0.5	0.0101
	watershed	*	*	*	*
Lab+XY	kmeans	50	10	D	0.2898
	gmm	20	10	0.1	0.3516
	hierarchical	10	10	0.9	0.0104
	watershed	*	*	*	*
HSV+XY	kmeans	50	10	D	0.2007
	gmm	50	10	D	0.2653
	hierarchical	20	10	D	0.0073
	watershed	*	*	*	*

Figure 3. Best results for each color space (RGB+XY, Lab+XY and HSV+XY) with the given parameters.

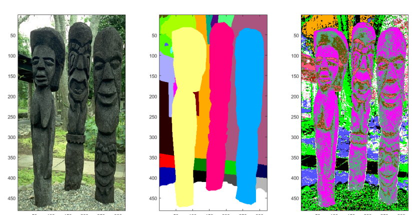

Figure 4. Segmentation comparison of the best result.

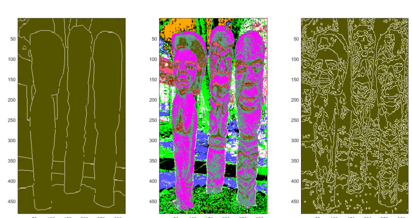

Figure 5. Boundary comparison of the best result.

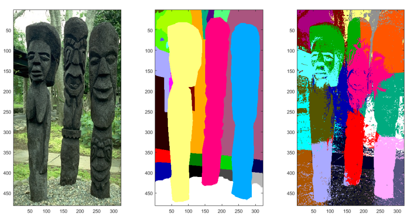

Figure 6. Segmentation comparison of the second best result.

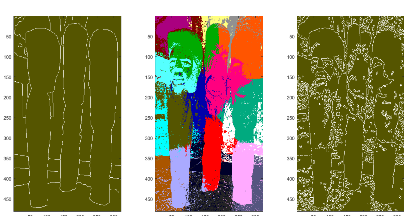

Figure 7. Boundary comparison of the second best result.