BSDS

Jaime Enrique Cascante Vega
Universidad de los Andes
je.cascante10@uniandes.edu.co

Angela Castillo Aguirre
Universidad de los Andes
a.castillo13@uniandes.edu.co

Abstract

The segmentation is one of the open challenges in the Computer Vision area. The BSDS is a huge dataset from the University of Berkeley that is composed by the ground-truth data for the probabilities of boundaries and the segmentation in the regions. The aim of this study is to apply different techniques of clustering to segment the regions in the images, and then compare the methods with the recognized UCM method proposed by Pablo Arbeláez et al.

1. Introduction

Segmentation formulated a mathematical problem seeks to minimise the differences between the image and the groundtruth considering a relaxation on the edges (defined as the gradient of the images) and one can consider a regularization for posing the optimization problem in the given space. This can be in general written as:

$$\mathcal{F}(u, C) = \int_{\Omega} (u - u_0)^2 dx + \mu \int_{\Omega \setminus C} |\nabla(u)|^2 dx + \nu |C| \tag{1}$$

Where Ω is in general de image space, C the edges or boundaries of the segmentation regions, and ν a constant for relaxing the problem. The optimization problem is:

$$\min_{C} = \mathcal{F}(u, C)$$

D. Martin, et. al (2001) presents the BSDS database of segmentation produced by humans for images of a wide variety of natural scenes [1]. Before the introduction of this dataset almost qualitative evaluation were the methodology for discriminating algorithms. D. Martin, et al, (2004) propose the evaluation methodology for the boundary detection problem [2] and M. Maire, et. al (2008) propose that the perceptual grouping vision problem can be evaluated using the same methodology [3]. This evaluation propose that algorithms rather than just the pixel in the boundaries must provided the probability that a pixel is in a boundary, this probability might be gathered using the most common approach: hierarchical image segmentation. Considering the pixel probability of being part of a boundary authors can present precision (P) and recall (R) given a threshold in the range $[0, 1]$, then the average precision AP or F-measure 2 can be reported as the general performance of the algorithm.

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN}$$
$$F_{meas} = \frac{2PR}{P + R} \tag{2}$$

The BSDS500 dataset have ground-truths or annotations from different 5 humans, hence, the human performance is also measured in the dataset with a maximum $F_{score} = 0.79$. In figure above are shown 5 examples images of the dataset with their correspondent segmentation and probability of boundary ground-truth.

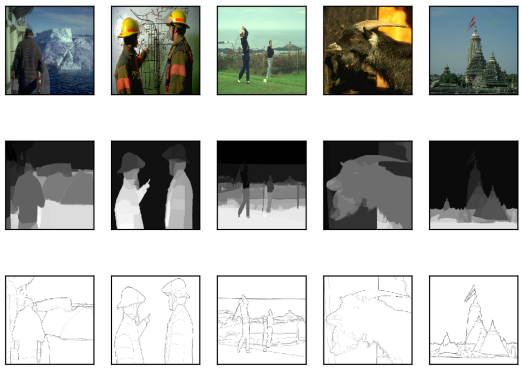


Figure 1: 5 random examples of the BSDS500 dataset. **Top:** Original images, **Middle:** Segmentation groundtruth, **Bottom:** Boundary probability groundtruth.

In this work we asses the general segmentation or perceptual grouping problem using just the pixel representation

in Lab and RGB colour space including spatial information pixel-wise. Hence, the representation space of feature space will be given by:

$$\mathbf{x}_{ij} = [I_1, I_2, I_3] \quad \mathbf{x}_{ij} = [I_1, I_2, I_3, i, j]$$

Where x_i is the pixel in position (i, j) of the image and I_i is the correspondent colour space channel. For addressing this problem we use common clustering algorithms for produce hierarchical segmentations, **K-means** and **Watersheds**. We selected these methods based principally on three criteria: *computational time*, *consistence of multiscale segmentation* and *number of hyper-parameters to tune*. This last one refers to the fidelity with the definition of multiscale segmentations, high scales segmentation must be joints, at least as possible, of low level segmentations. Additionally we consider the context in which the algorithms are used.

2. Methodology

2.1. Clustering algorithms

K-means provide high consistency of multi-scale segmentation, as the parameter K increase segmentations that were initially nested (cluster of pixels) break up in different groups, however has a high computational cost that increase propotional to $\mathcal{O}(nK)$ where n is the number of pixels to cluster and K the number of clusters. We select **K-means** over **GMM** due to their definition, **GMM** as a parametric clustering algorithm assume that the different observations in each feature space follows a normal independent distribution. Although the first assumptions is strong the second **independence** really fuck up the algorithm, images have are really dependant feature space or input representation, furthermore when considering localization, (X, Y) features, by definition we assume that the position is dependent of the other features: brightness and colour. K-means only need to tune one hyperparameter: the number of cluster K, and using multi-scale representation we are optimizing that parameter in both training and validation set.

Watersheds is one of the most used algorithms for large multi scale representation, furthermore by definition watershed consider edges, segmentations boundaries, in the change of the gradient of the feature space. Hence, authors proposed watershed variants in which they consider not only the gradient but also the orientation of the detected edges, and matches globally the orientations of adjacent object boundaries [4]. Additionally, as we consider only the gradient and the regional minima to impose, for considering multiscale representation, the computational cost is reduced. Additionally the catchment basins of the regional minima considered are produced by over-segmentations of watersheds unconstrained, hence these minima correspond to objects in different scales. This definition is consistent with the nested multi-scale representations. Watersheds have no hyper-parameters to tune, however one can consider the regional minima for determining the current scale representation as a hyper-parameter not direct of watersheds but in general of the methodology proposed.

Even agglomerative clustering have in general good performance, the computational cost is really large and therefore we do not consider it.

2.2. Preprocessing and Enhancement

2.2.1 Preprocessing

The input space have in general different representation scale, for example in *Lab* $L \in [0, 100]$ and $a \in [-a, a]$, $b \in [-b, b]$ and when considering the localization of the pixel $x \in [0, w]$ and $y \in [0, h]$ where w is the width of the images and h the heigh of the image. Hence we use mean normalization in each input channel given by:

$$z_i = \frac{x_i - \mu_i}{\sigma_i}$$

Where x_i is $x_i = [x_1^1, x_1^2, ..., x_1^n]^T$ the $i - th$ input feature with n observations (pixels), μ_i the mean of x_i and σ_i the standard deviation of x_i. Therefore the resultant input vector z_i have zero mean and unit variance ($\mathbb{E}(z_i) = 0$, $Var(z_i) = 1$).

2.2.2 Enhancement

A well known fact in psychology and computer vision is that intensity information is more semantically rich that colour information. This idea follows from the idea that intensity information is the most common representations of the visual system in mammals. Authors state that colour evolved for alternated task such as mating, feeding and water search [5]. Following this idea we enhance the intensity channel L in the Lab and $Lab + xy$ input space. We take gain values on L channel of $G_L = [1.2, 1.5, 2]$. Remark that $G = 1$ has no effect on the input space.

2.3. Evaluation

For evaluating quantitatively the results obtained we use the metrics proposed in [3]. Basically from a multiscale segmentation one can construct precision recall curves based on hierarchical thresholding of the segmentations and computing the correspondent $F_{measure}$. Originally this evaluation was proposed for algorithms that output the probability of a pixel for being in a boundary, however as we are consider different levels of clustering (K-means) and regional minima (Watersheds) the optimal threshold

is the optimal number of cluster of the algorithm in the dataset [4]. We consider cluster and regional minima correspondent to K or minima level in the sequence $\{2, 4, 6, 8, 10, 12, 14, 15, 16, 18, 20\}$.

Precision recall curves gives information not only of the general performance of the algorithm (F_{measure}) but also of the performance of the algorithm over specifically tasks. High precision and small recall indicate that the algorithm output contours that highly match in continuity with the contours in the annotation. Similarly high Recall and small precision indicates that the algorithm output a lot of contours but a lot of these outputs does not correspond to the annotations. Hence, the best results correspond to high precision and high recall $F_{\text{measure}} = 1$.

Meas / F.S	*ODS*	*OIS*	*ODS*	*OIS*
RGB+xy	0.33	0.41	0.35	0.38
Lab	0.33	0.39	–	–
Lab+xy	0.36	0.43	0.35	0.37
Lab+xy $G_L = 1.2$	0.36	0.43	–	–
Lab+xy $G_L = 1.5$	0.36	0.43	–	–
Lab+xy $G_L = 2$	0.36	0.43	–	–

Table 1: Best results at optimal threshold, i.e maximum F_{meas}. Two first colums correspond to test results with **K-means**, two last to test results with **Watersheds**.

3. Results and Discussion

In table 1 are summarized the best results for the methods proposed. A can be seen contrary to the intuition enhancing the brightness channel does not show any significant improvement. However, as can be seen in figure **??** the increasing the brightness gain smooth the clusters obtained. Additionally we might be wrong with the values of the gain, we select values in a small range, hence no significant results were obtained. From table 1 and figure **??** we see that using the same number of clusters in *K-means* is not equivalent to the number of regional minima. This, is because watersheds consider gradient information, hence, regions are produced based on intensity change, but in $K - means$ the input space is sub-sampled for producing sets of points, hence for obtaining a more detailed precision recall for watershed the number of regional minima imposed must be increased.

In figure 4 are shown the **baseline** results. Watersheds have really poor performance in both precision and recall, however, as discussed above increasing the number of regional minima imposed might increase the characterization of the algorithm. Additionally, it can be seen that for the same number of clsuter of K-means and regional minima on watershed different regions of the precision recall curve are covered, watersheds have approximately constant precision but seem to increase its recall and alternatively K-means have large values of recall put poor performance in precision. This might be an expected result as with larger values of K the segmentations has no semantic content, hence is over-segmentated based on small changes of the input space.

Qualitative results are shown in figures 5-7. In figure 5 is shown the poor performance of watershed with the best threshold, contrary to $K - means$ that despite the over-segmentation look of the image we can see that it preserves important semantic information, body of sea-star, bird, etc. However it can also be seen that the x, y feature input bias the segmentation of similar texture like objects into spatial segmentation. This is in general an undesired property, however, we can see in other objects where texture is not that homogeneous with the background that the clustering methods preserve more smoothly the object regions.

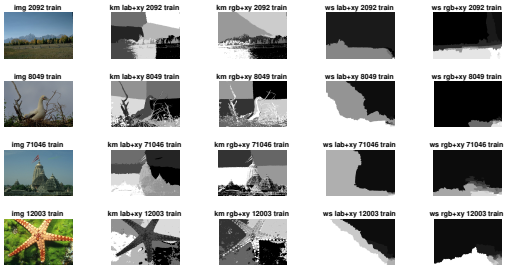
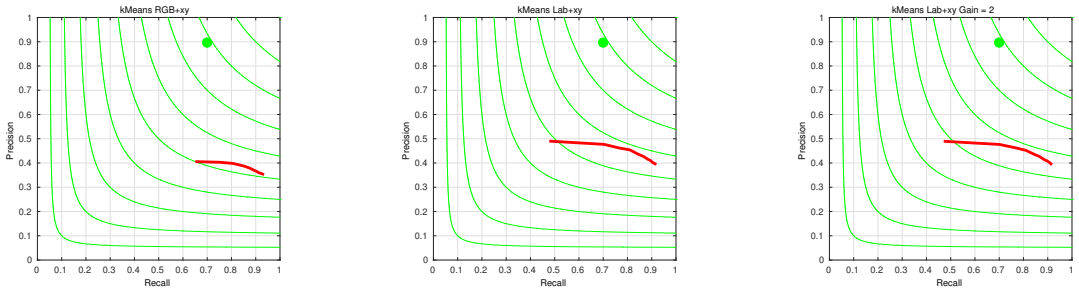
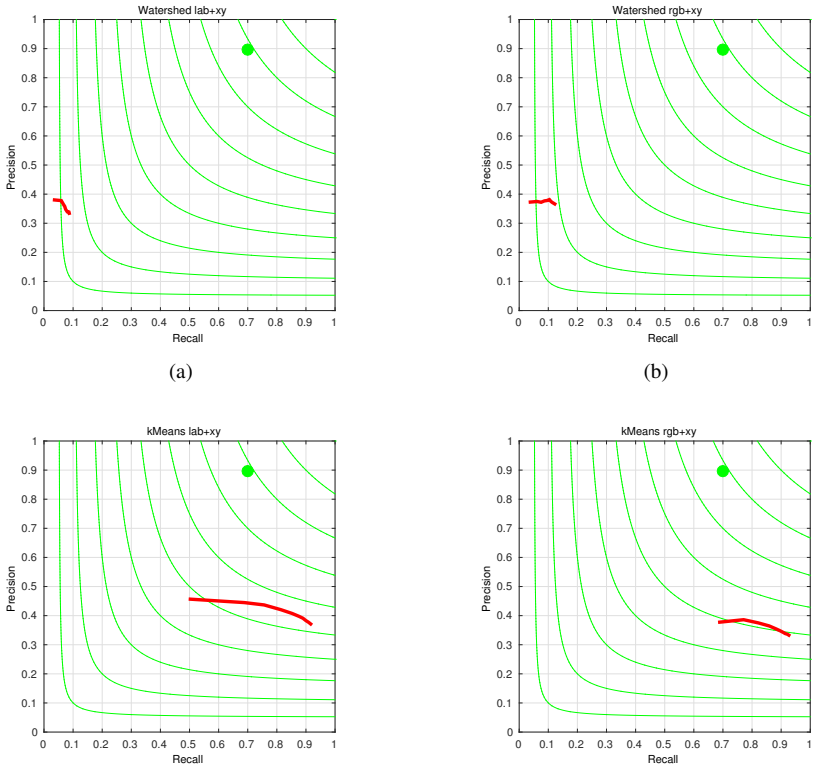


Figure 5: 4 random examples of the train segmentation results. **First Column:** Original images, **Second Column:** Segmentation using K-Means and Lab+xy, **Third Column:** Segmentation using K-Means and RGB+xy, **Fourth Column:** Segmentation using Watershed and Lab+xy, **Last Column:** Segmentation using Watershed and RGB+xy.



(a) (b) (c)

Figure 2: Test results for best methods. From **left** to **right**: K-means on $RGB + xy$, K-means on $Lab + xy$ and K-means on $Lab + xy$ with $G_L = 2$.



(a) (b) (c) (d)

Figure 3: Test precision-recall curves for nominal condition, i.e no gain in L channel.

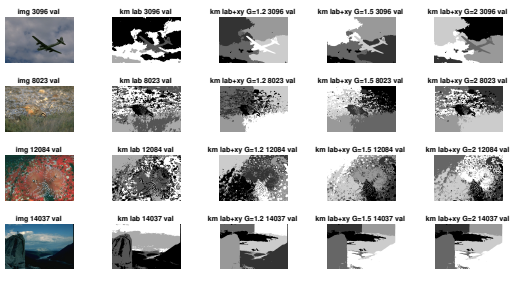


Figure 6: 4 random examples of the validation segmentation results. **First Column:** Original images, **Second Column:** Segmentation using K-Means and Lab, **Third Column:** Segmentation using K-Means and Lab+xy with $G = 1.2$, **Fourth Column:** Segmentation using K-Means

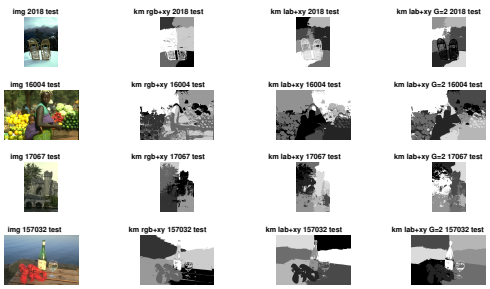


Figure 7: 4 random examples of the validation segmentation results. **First Column:** Original images, **Second Column:** Segmentation using K-Means and RGB+xy, **Third Column:** Segmentation using K-Means and Lab+xy,

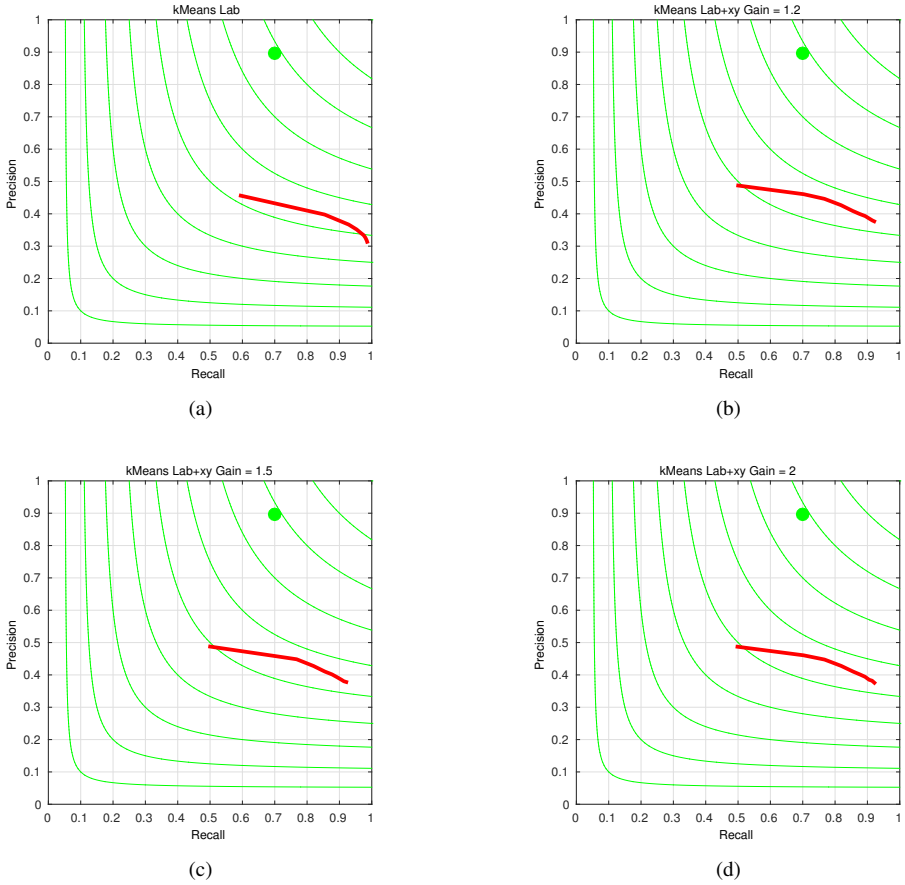


Figure 4: Validation set precision-recall curves for optimizing intensity gain G_l. **Top Left:** Lab as Input space, no brightness gain. **Top Right:** $Lab + xy$ as Input space, brightness gain $G_L = 1.2$. **Bottom Left:** $Lab + xy$ as Input space, brightness gain $G_L = 1.5$. , **Bottom Right:** $Lab + xy$ as Input space, brightness gain $G_L = 2$.

In figure above are showed our best results against the result baseline results proposed in [4] against our best methods.

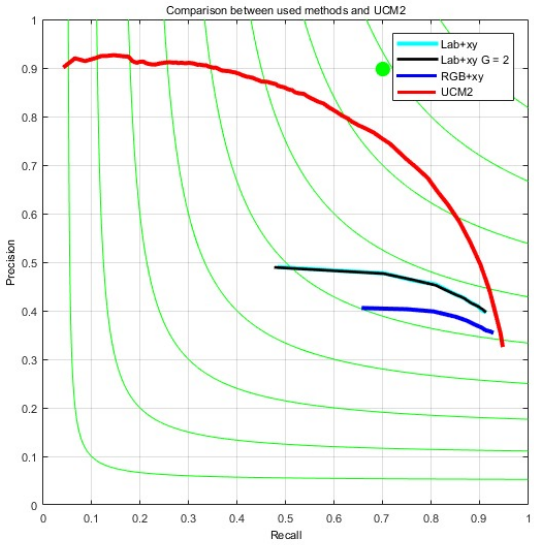


Figure 8: Performance of UCM vs our best methods.

Arbealez. et.al report a maximal $F_{\text{measure}} = 0.71$, from above image, as seen none of our methods produce better results. This is due to much reasons however in [4] is discussed that globalization of the input space is in general a good idea for producing better results. The basic idea lies in supposing that edges are not only defined for local information like gradient, but also for global information of the image and semantic information of the object. Our implementations not even consider pixel-neighbourhood information, and therefore the performance is reduced to local information gathering.

4. Conclusions

We can conclude that the best algorithm for the segmentation task in this context is *K-means* with input space as $Lab + xy$ and intensity gain $G_L = 2$. It is an expected result, must of the research methods, before machine learning and deep learning became mainstream in the computer vision field, that most use $K - means$ at some level for assignment of histogram or for more compact representation [4, 6].

About the evaluation methodology the BSDS500 benchmark provide a general framework for comparing algorithm, form last laboratory we select the methods mostly based on their qualitative performance. Hence, the performance tested on this work is quantitative measurable and then algorithm performance can be tested.

Most of the limitations of the algorithm used in this work is the lack of globalization, as we said before the input space only consider pixel-wise information not even considering patches or neighbourhoods. Additionally from image 6 we see that the regions obtained have artefact due to the x, y channels, furthermore, when no spatial information is considered we see that the algorithm produce over-segmentations, i.e on small changes of intensities the algorithm identifies different clusters.

Finally we think that using more *intelligent* hand-crafted features will increase the performance. We suggest to consider for example local cues descriptors as, colour, texture, gradient brightness.

References

[1] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*.

[2] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 530–549, 2004.

[3] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik, "Using contours to detect and localize junctions in natural images," *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

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

[5] J. K. Bowmaker, "Evolution of colour vision in vertebrates," *Eye*, vol. 12, no. 3, pp. 541–547, 1998.

[6] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1265–1278, May 2005.