Fast Superpixel-Based hierarchical approach to image segmentation

Publisher:

Springer Verlag

Published version:

DOI:10.1007/978-3-319-23231-7_33

Fast Superpixel-based Hierarchical Approach to Image Segmentation

Francesco Verdoja and Marco Grangetto

Università degli Studi di Torino, Computer Science department, Torino, Italy

Abstract. Image segmentation is one of the core task in image processing. Traditionally such operation is performed starting from single pixels requiring a significant amount of computations. It has been shown that superpixels can be used to improve segmentation performance. In this work we propose a novel superpixel-based hierarchical approach for image segmentation that works by iteratively merging nodes of a weighted undirected graph initialized with the superpixels regions. Proper metrics to drive the regions merging are proposed and experimentally validated using the standard Berkeley Dataset. Our analysis shows that the proposed algorithm runs faster than state of the art techniques while providing accurate segmentation results both in terms of visual and objective metrics.

Keywords: segmentation, superpixels, graph partitioning, hierarchical clustering, CIEDE2000, Mahalanobis distance, Bhattacharyya distance

1 Introduction

Region segmentation is a key low-level problem in image processing, as it is at the foundation of many high-level computer vision tasks, such as scene understanding [7] and object recognition [8]. Traditionally regions are found by starting from single pixels and then use different approaches to find clusters of pixels. Some examples of methods include region growing [14], histogram analysis [15] and pyamidal approaches [12]; another very commonly used class of algorithms treats the image as a graph. Graph-based techniques usually consider every pixel as a node in a weighted undirected graph and then they find regions in two possible ways: by partitioning the graph using some criterion, or by merging the nodes that are most similar according to a similarity measure. Methods of the first subclass are usually based on graph-cut and its variations [18] or spectral clustering [6]. For what concerns node merging techniques, one algorithm that has been widely used is the one by Felzenszwalb-Huttenlocher [4]. The criterion proposed in this latter work aims at clustering pixels such that the resulting segmentation is neither too coarse nor too fine. The graph is initialized considering every pixel as a node; the arcs between neighboring pixels are weighted with a proper dissimilarity measure (e.g. minimum color difference connecting two components). At every iteration the algorithm merges pair of nodes (components) that are connected by an edge characterized by a weight that is lower than the intra-component differences. As consequence, homogeneous components that are not separated by boundaries are progressively represented by the nodes of the graph.

A recent trend in segmentation, is to start the computation from superpixels instead of single pixels [16]. As shown in Fig. 1, superpixels are perceptually meaningful atomic regions which aim to replace rigid pixel grid. Examples of algorithms used to generate these kind of small regions are Turbopixel [9] and the widely used and very fast SLIC algorithm [1]. Over-segmenting an image using one of said techniques, and the performing actual region segmentation, can be interesting both in term of reducing the complexity of the problem (i.e. starting from superpixels instead of single pixels) and improving the quality of the final result, thanks to the intrinsic properties of superpixels [10].

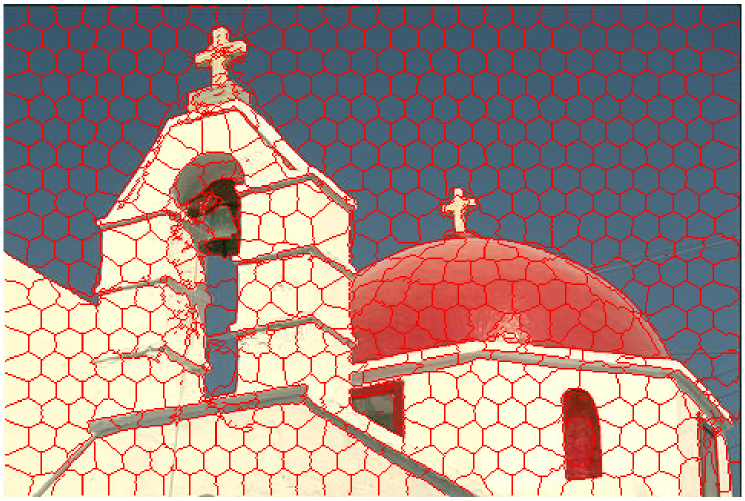

Fig. 1. An image divided into approximately 600 superpixels

In this study, we analyze the benefits of using a simple merging approach over a graph whose nodes are initialized with superpixels regions. The main contributions of the paper are:

- design of a local merging approach for the selection of the pair of superpixels that are likely to belong to the same image region;
- exploitation of CIELAB color space in the definition of the dissimilarity metric so as to better match human color perception;
- analysis of the performance and complexity trade-off with respect to the state of the art.

Our conclusions are that superpixels can efficiently boost merging based segmentation techniques by reducing the computational cost without impacting on the segmentation performance. In particular we show that such result can be achieved even without resorting to global graph partitioning such as graph-cut [20] or spectral clustering [10].

It's important to note that altough other superpixel-based hierarical approaches have been proposed in the past, the most notable among them by Jain et al. [5], none of them have been intended as a general-use segmentation technique. The work by Jain et al., for example, has been tested only on human brain images, and its validity on standard datasets is not known. The performance of the proposed algorithm, which is intended to work on any type of image, are going to be instead objectively evaluated on a well known standard dataset for image segmentation.

The paper is organized as follows. In Sect. 2 the proposed segmentation technique is presented, whereas in Sect. 3 and Sect. 4 complexity and segmentation results are discussed, respectively.

2 The Proposed Technique

Let's start by defining an n regions segmentation of an image $I = \{x_i\}_{i=1}^{N}$ with N pixels as a partition $L = \{l_i\}_{i=1}^{n}$ of the pixels of I; more precisely, the segmented regions must satisfy the following constraints:

$$\begin{array}{l} \forall x \in I, \exists l \in L \mid x \in l \ ; \\ \forall l \in L, \nexists l' \in L - \{l\} \mid l \cap l' \neq \emptyset \ . \end{array} \tag{1}$$

Please note that in the rest of the paper the terms *region*, *label*, and *segment* are going to be used interchangeably to refer to one of the parts of the segmented image, i.e. one of the set of pixels l.

In this paper we propose to initialize the segmentation algorithm with an over-segmented partition L^m. This first segmentation can be obtained with any superpixel algorithm. Since the quality of the starting superpixels is not going to be checked by the proposed technique, the segmentation accuracy of the chosen algorithm for finding superpixels is of crucial importance in this context. In this work SLIC has been used given its known computational efficiency and segmentation accuracy [1].

Starting from an image I and a partition L^m composed of m regions, the proposed algorithm aims at merging at each iteration the pair of labels representing the most similar regions between the ones determined in the previous step. In particular at the k-th iteration the two most similar between the k segments of L^k are merged to obtain a new set L^{k-1} composed of $k-1$ segments. This process can be iterated for $k = m, m-1, \ldots, 2$; when $k = 2$ a binary segmentation L^2 is obtained, where only foreground and background are discriminated.

The proposed iterative merging algorithm generates a full dendrogram, that carries information about the hierarchy of the labels in terms of regions similarity. We can represent the merging process using a weighted graph. When the algorithm starts, an undirected weighted graph $G^m = \{L^m, W^m\}$ is constructed over the superpixel set L^m, where

$$W^m = \left\{w_{ij}^m\right\}, \ \forall i \neq j \mid l_i^m, l_j^m \in L^m \wedge A\left(l_i^m, l_j^m\right) = 1 \tag{2}$$

for some adjacency function A. Since G^m is an undirected graph we have that $w_{ij}^m = w_{ji}^m$; the weights represent the distance (or dissimilarity measure) between pair of regions $w_{ij}^m = \delta\left(l_i^m, l_j^m\right)$. The possible functions that can be used to compute the distance δ are going to be discussed in detail in Sect. 2.1.

At each iteration, the algorithm picks the pair of labels $l_p^k, l_q^k \in L^k$ having $w_{pq}^k = \min\left\{W^k\right\}$ and merges them; i.e. it generate a new partition $L^{k-1} = L^k - \left\{l_q^k\right\}$ having all the pixels $x \in l_p^k \cup l_q^k$ assigned to the label l_p^{k-1}. L^{k-1} contains now just $k-1$ segments. After that, edges and corresponding weights needs to be updated as well. W^{k-1} is generated according to the following rule:

$$w_{ij}^{k-1} = \begin{cases} \delta\left(l_p^{k-1}, l_j^{k-1}\right) & \text{if } i = p \vee i = q \ , \\ w_{ij}^k & \text{otherwise} \ . \end{cases} \tag{3}$$

Please note that w_{pq}^k is not going to be included in W^{k-1} since it doesn't exist anymore.

When $k = 2$, the algorithm stops and returns the full dendrogram $D = \left\{L^m, \ldots, L^2\right\}$ that can be cut at will to obtain the desired number of regions. An example of different cuts of the dendrogram can be seen in Fig. 2.

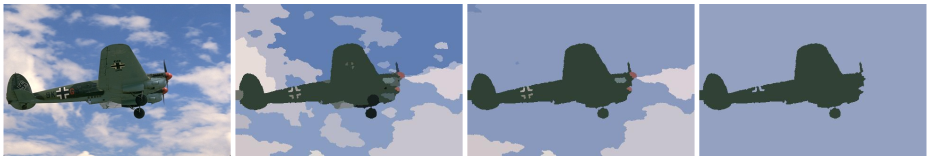

Fig. 2. A sample image and hierarchy of 3 segmentations obtained with $k = 50, 15, 2$ and δ_C metric.

2.1 Region Distance Metrics

The approach proposed here can be used in conjunction with several distance metrics capable to capture the dissimilarity between a pair of segmented regions. In the following we discuss a few alternatives that will be used in our experiments.

The first and simplest one that we have explored is color difference between the two regions. To better match human color perception, CIELAB color space and the standard CIEDE2000 color difference have been chosen [17]. Given two regions l_1 and l_2, we compute the mean values of the L*a*b* components $M_1 = (\mu_{L^*,1}, \mu_{a^*,1}, \mu_{b^*,1})$ and $M_2 = (\mu_{L^*,2}, \mu_{a^*,2}, \mu_{b^*,2})$, and we define the distance between the two labels as

$$\delta_C\left(l_i, l_j\right) = \Delta E_{00}\left(M_i, M_j\right) \tag{4}$$

where ΔE_{00} is the CIEDE2000 color difference [17].

Another possibility it to exploit the Mahalanobis distance [11] given its ability to capture statistical differences between two distributions of color component. Given a set of n_1 pixels $l_1 = \{x_i = (x_{L^*,i}, x_{a^*,i}, x_{b^*,i})\}_{i=1}^{n_1}$, we can estimate their mean $M_1 = (\mu_{L^*}, \mu_{a^*}, \mu_{b^*})$ and covariance as

$$C_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} (x_i - M_1)(x_i - M_1)^T \quad . \tag{5}$$

Then we compute the Mahalanobis distance of any other set of n_2 pixels $l_2 = \{y_i = (y_{L^*,i}, y_{a^*,i}, y_{b^*,i})\}_{i=1}^{n_2}$ from the estimated distribution of l_1 as

$$\Delta M(l_1, l_2) = \frac{1}{n_2} \sum_{i=1}^{n_2} (y_i - M_1)^T C_1^{-1} (y_i - M_1) \quad . \tag{6}$$

Since ΔM is non symmetric, i.e. $\Delta M(l_1, l_2) \neq \Delta M(l_2, l_1)$, we compute the distance between two labels as the minimum of their relative Mahalanobis distances obtaining the following symmetric metric:

$$\delta_M(l_i, l_j) = \min\{\Delta M(l_i, l_j), \Delta M(l_j, l_i)\} \quad . \tag{7}$$

Since during the iterative merging process is important to merge homogeneous regions, in particular without crossing object boundaries, we also investigate a local Mahalanobis metric that aims at detecting image segment whose adjacent borders look very different. This border variation consists in evaluating the Mahalanobis distance just for the pixels near the border between the two regions. More precisely, let us define b_{ij} the portion of common border between two adjacent image segments. Then we can define a subset of pixels whose location is across the two adjacent regions $c_{ij} = \{x \in I \mid r_1 < d(x, b_{ij}) < r_2\}$, where d is the Euclidean spatial distance and r_1 and r_2 are proper ranges. Now we can introduce function $B(l_i, l_j)$ that returns two new set of pixels $l'_i = l_i \cap c_{ij}$ and $l'_j = l_j \cap c_{ij}$ that represent the pixels of l_i and l_j respectively that are located close to the common border. Finally, the distance metric is defined as:

$$\delta_B(l_i, l_j) = \min\left\{\Delta M\left(l'_i, l'_j\right), \Delta M\left(l'_j, l'_i\right)\right\} \tag{8}$$

where l'_i and l'_j are the two outputs of $B(l_i, l_j)$.

Finally, we investigate a fourth metric based on the color histogram distance. One possible solution to measure histogram difference is the Bhattacharyya distance [3], which is the general case of the Mahalanobis distance. Given two histograms h_1 and h_2 composed each by B bins, the Bhattacharyya distance is defined as

$$\Delta H(h_1, h_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{h}_1 \bar{h}_2 B^2}} \sum_{i=1}^{B} \sqrt{h_1(i) \cdot h_2(i)}} \tag{9}$$

where $h(i)$ is the number of pixels in the bin i, while $\bar{h} = \frac{1}{B}\sum_{i=1}^{B} h(i)$. Since images in the L*a*b* color space have three channels, ΔH is going to be computed on each channel independently, and then the maximum value of the three is going to be used as dissimilarity measure; this has been chosen over other possibility, like taking the mean of the three distances, as it yields higher discriminating power in finding differences just on one of the channels. In conclusion, the last dissimilarity measure between two regions l_i and l_j having respectively histograms $H_i = \{h_{L^*,i}, h_{a^*,i}, h_{b^*,i}\}$ and $H_j = \{h_{L^*,j}, h_{a^*,j}, h_{b^*,j}\}$ is defined as:

$$\delta_H\left(l_i, l_j\right) = \max \left\{ \begin{matrix} \Delta H\left(h_{L^*,i}, h_{L^*,j}\right), \\ \Delta H\left(h_{a^*,i}, h_{a^*,j}\right), \\ \Delta H\left(h_{b^*,i}, h_{b^*,j}\right) \end{matrix} \right\} . \tag{10}$$

3 Complexity

In this section the complexity of the proposed algorithm is going to be discussed. We will start by analyzing the complexity of the distance metrics presented in Sect. 2.1. To this end let us consider any two regions l_1 and l_2 with a total number of pixels $n = |l_1 \cup l_2|$. The complexity of the different distance metrics is discussed in the following.

δ_C Computing the color mean of both regions requires $O(n)$ time while computation of distance between the mean values has unitary cost.

δ_M All the operations required to compute Mahalanobis distance (mean and color covariance estimates) are in the order of $O(n)$.

δ_B Since the computation is going to be performed on the $n' = |l'_1 \cup l'_2|$ pixels in the border area, the complexity is again $O(n')$, with $n' < n$.

δ_H The dominant cost is assigning every pixel to a bin; then, the cost of calculating the actual distance is negligible. Therefore the overall complexity is $O(n)$ also in this case.

To recap, computing any of the distances we proposed is linear to the number of pixels in the considered segments. Then, according to (1) computing all distances for a whole partition L of an image of N pixels will require $O(N)$ time.

Finally, we can discuss the overall complexity of all the algorithm steps:

1. The starting step of the algorithm is to compute the m superpixels. For that purpose, using SLIC, $O(N)$ time is required [1];
2. Next, the graph G^m needs to be constructed. The time required for this task is in the order of $O(N)$, as all the weights needs to be computed once;
3. Then, the m merging iterations are performed. At every iteration just a small number of the weights is going to be updated, and since all the regions are going to be merged once, the overall complexity is once again $O(N)$.

In conclusion, the overall time required by the algorithm is linear to the size of the image.

We can conclude that the proposed technique exhibits lower complexity than both merging techniques that works on pixels, like the Felzenszwalb-Huttenlocher algorithm which has complexity of $O(N \log N)$ [4], and other widely used techniques that works on superpixels, like SAS [10] and ℓ_0-sparse-coding [20], which both have complexities higher than linear.

To verify our claims, in Fig. 3 the running times of the different components of the algorithm are shown. It can be noted that the time needed by both SLIC and the clustering algorithm using all the different distance measures here proposed are growing linearly to the size of the input.

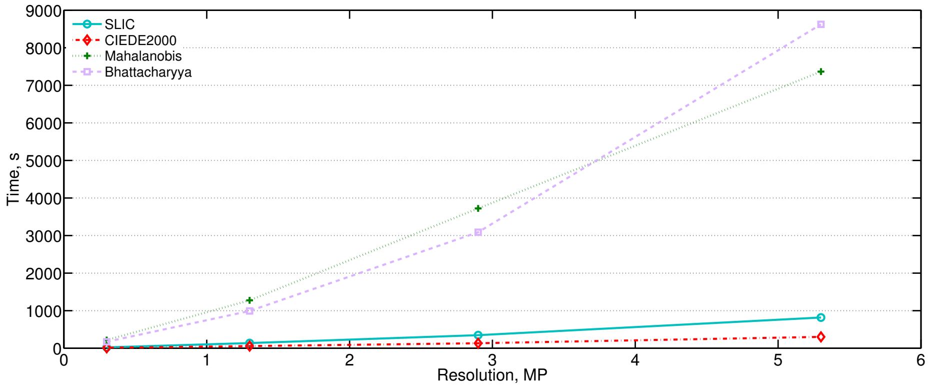

Fig. 3. Mean running times of SLIC and the proposed clustering algorithm using the different distance measures; these results are computed on three 5.3MP images scaled at different smaller resolutions.

4 Performance Evaluation

In this section the performance of the proposed algorithm is validated both visually and using objective metrics. To this end the standard Berkeley Dataset BSDS500 [2] has been used. This latter, although originally constructed for boundaries evaluation, has become a well recognized standard for evaluation of regions segmentation in images.

The discussion on objective metrics for an effective evaluation of the segmentation performance is still open [2]; still the usage of a standard set of images makes our results easier to reproduce and compare with past and future research.

In this work we have selected as benchmarks for performance evaluation two well known superpixel-based algorithms, namely SAS [10] and ℓ_0-sparse-coding [20]. Moreover, the Felzenszwalb-Huttenlocher algorithm [4] has been selected as representative of a merging approach that starts from individual pixels.

4.1 Metrics

Two common metrics have been used to evaluate the performance over the dataset. They have been chosen because results using these metrics are available for all the algorithms that have been cited in this work. For the Felzenszwalb-Huttenlocher algorithm they can be found in [2], while for ℓ_0-sparse-coding and SAS they can be found directly in the respective papers.

Probabilistic Rand Index. The Probabilistic Rand Index is a variation of the Rand Index, proposed for dealing with multiple ground-truths [19]. It is defined as:

$$PRI\left(S,\{G_k\}\right)=\frac{1}{T}\sum_{i<j}\left[c_{ij}p_{ij}+\left(1-c_{ij}\right)\left(1-p_{ij}\right)\right] \tag{11}$$

where c_{ij} is the event that pixels i and j have the same label while p_{ij} is its probability. T is the total number of pixel pairs. To average the Rand Index over multiple ground-truths, p_{ij} is estimated from the ground-truth dataset.

Variation of Information. The Variation of Information (VoI) metric allows one to compare two different clusterings of the same data [13]. It measures the distance between two segmentations in terms of their conditional entropy, given as:

$$VoI\left(S,S'\right)=H\left(S\right)+H\left(S'\right)-2I\left(S,S'\right) \tag{12}$$

where H represents the entropy and I the mutual information between two clusterings of data, S and S'. In the case presented here, these clusterings are the segmentations performed by the algorithms to be tested and the ground-truths.

4.2 Results

First of all in Fig. 2 and Fig. 4 we show some segmentation results obtained using the simple color metric difference δ_C; every segmented region is filled with its mean color. Figure 2 reports different segmentations of the same image obtained by stopping the hierarchical clustering at progressively lower numbers of regions showing that the proposed solution can achieve different levels of segmentation granularity down to the separation into foreground and background. The images shown Fig. 4 are obtained selecting the value of k that yields the best overlap with ground-truth segmentations in the BSDS500 dataset. It can be observed that the proposed solution is able to effectively segment images; the boundary accuracy clearly depends on the starting superpixel technique, e.g. in our case SLIC, whereas the proposed hierarchical merging criterion can group the main image regions very effectively.

We do not show images segmented using the other similarity metrics proposed in Sect. 2.1 since they yields similar visual results.

In Tab. 1 objective segmentation metrics computed on the BSDS500 dataset are shown. In particular, we report PRI and VoI results yielded by our method with the four different similarity metrics proposed in Sect. 2.1 and other benchmarks in the literature. We started with 600 superpixels, then for the calculation of boundary-based metric δ_B we have set $r_1 = 3$ and $r_2 = 11$ respectively, while for δ_H we have set $B = 20$. From the obtained results it can be noted that all the techniques we compare exhibits about the same value of PRI. Moreover, it can be noted that the proposed solution yields better VoI results than the Felzenszwalb-Huttenlocher pixel based algorithm and competing superpixel

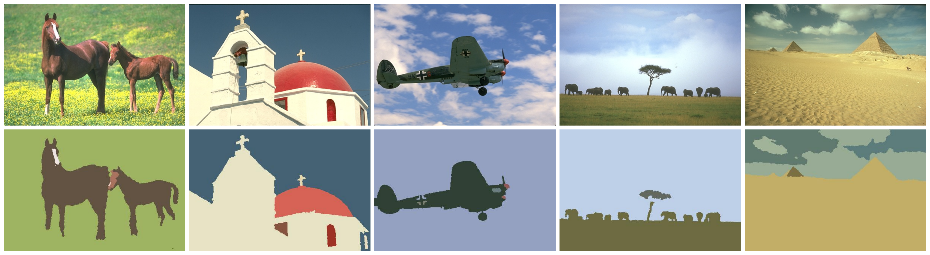

Fig. 4. Sample images from BSDS500 (top) and their best corresponding segmentation outputs (bottom) using δ_C metric.

based ℓ_0-sparse-coding [20]. Only the SAS [10] algorithm exhibits a lower value for VoI. At the same time, it is worth recalling that the proposed technique is by far the cheapest in terms of computational cost with respect to the other benchmarks.

Table 1. Results obtained by the proposed technique in all its variations compared to other state-of-the-art techniques over the BSDS500

Algorithm	**PRI**	**VoI**
SAS [10]	0.83	1.68
ℓ_0-sparse-coding [20]	0.84	1.99
Felzenszwalb-Huttenlocher [4]	0.82	1.87
Ours (using δ_C)	0.83	1.78
Ours (using δ_M)	0.83	1.71
Ours (using δ_B)	0.82	1.82
Ours (using δ_H)	0.81	1.83

We can also note that color and Mahalanobis metric provides the same segmentation accuracy. On the other hand the histogram and boundary based metrics are slightly less effective. This slight difference in performance can be explained by considering that superpixel over-segmentation is able to i) retain very homogeneous areas; ii) accurately follow image boundary. The first feature makes the advantage of a more statistically accurate metric for the description of intra-pixel color variation, such as Mahalanobis distance, negligible with respect to simple color distance in L*a*b* space. Finally, the fact that superpixels does not cut image edges makes the usage of a boundary based criterion ineffective.

In Fig. 5 we conclude the analysis of our results by showing the precision/recall curves yielded by the four proposed region distance metrics. The curves have been obtained by comparing the segmentation generated by our algorithm setting different values for k with ground-truth data in BSDS500 dataset. It can be observed that δ_C and δ_M appears to be slightly superior to both δ_B and δ_H also in terms of precision/recall trade-off.

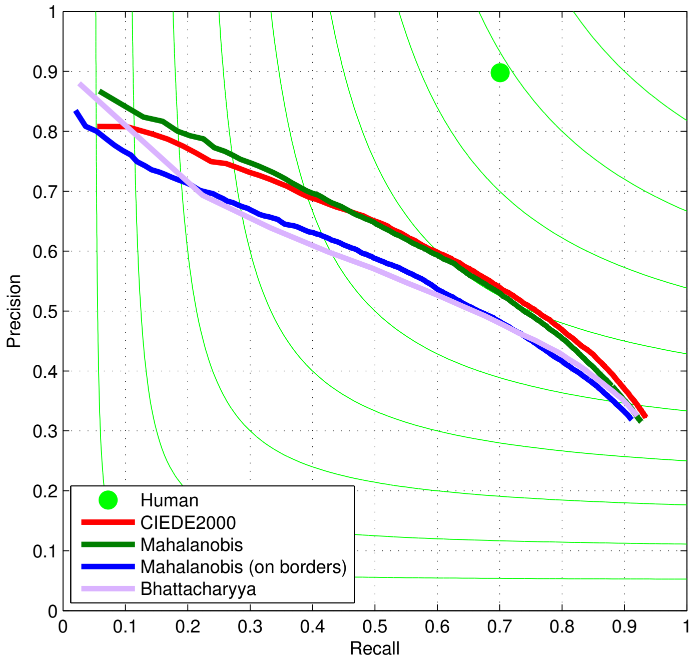

Fig. 5. Precision and recall of the proposed technique, using δ_C, δ_M, δ_B and δ_H

5 Conclusions

In this paper a new approach to image segmentation has been presented. The proposed approach is based on iterative merging of nodes in a graph initialized with an over-segmentation of an image performed by a superpixel algorithm. The algorithm employs proper distance metrics to select regions to be merged. We have shown that both CIEDE2000 and Mahalanobis color distances are very effective in terms of segmentation accuracy. Our experimentation worked out on the BSDS500 dataset shows that the proposed tool yields competitive results with respect to other state of the art techniques that segments starting both with superpixels and single pixels. Finally, one of the most important achievements is that the overall complexity of the proposed method is kept linear to the dimension of the image as opposed to the other techniques we compare to.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(11), 2274–2282 (Nov 2012)
2. Arbeláez, P., Maire, M., Fowlkes, C.C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(5), 898–916 (2010)

3. Bhattacharyya, A.K.: On a measure of divergence between two statistical populations defined by their probability distributions. Bulletin of Calcutta Mathematical Society 35(1), 99–109 (1943)
4. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. International Journal of Computer Vision 59(2), 167–181 (2004)
5. Jain, V., Turaga, S.C., Briggman, K.L., Helmstaedter, M.N., Denk, W., Seung, H.S.: Learning to agglomerate superpixel hierarchies. In: Advances in Neural Information Processing Systems. pp. 648–656 (2011)
6. Kim, S., Nowozin, S., Kohli, P., Yoo, C.D.: Higher-order correlation clustering for image segmentation. In: Advances in Neural Information Processing Systems. pp. 1530–1538 (2011)
7. Kumar, M.P., Koller, D.: Efficiently selecting regions for scene understanding. In: Computer Vision and Pattern Recognition (CVPR). pp. 3217–3224. IEEE (2010)
8. Lee, Y.J., Grauman, K.: Object-graphs for context-aware visual category discovery. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(2), 346–358 (2012)
9. Levinshtein, A., Stere, A., Kutulakos, K.N., Fleet, D.J., Dickinson, S.J., Siddiqi, K.: Turbopixels: Fast superpixels using geometric flows. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(12), 2290–2297 (Dec 2009)
10. Li, Z., Wu, X.M., Chang, S.F.: Segmentation using superpixels: A bipartite graph partitioning approach. In: Computer Vision and Pattern Recognition (CVPR). pp. 789–796. IEEE (2012)
11. Mahalanobis, P.C.: On the generalized distance in statistics. In: National Institute of Sciences of India. vol. 2, pp. 49–55. Calcutta, India (1936)
12. Marfil, R., Molina-Tanco, L., Bandera, A., Rodrguez, J.A., Sandoval, F.: Pyramid segmentation algorithms revisited. Pattern Recognition 39(8), 1430–1451 (Aug 2006)
13. Meilă, M.: Comparing clusterings: an axiomatic view. In: Proceedings of the 22nd International Conference on Machine Learning. pp. 577–584. ACM (2005)
14. Nock, R., Nielsen, F.: Statistical region merging. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(11), 1452–1458 (2004)
15. Ohlander, R., Price, K., Reddy, D.R.: Picture segmentation using a recursive region splitting method. Computer Graphics and Image Processing 8(3), 313–333 (Dec 1978)
16. Ren, X., Malik, J.: Learning a classification model for segmentation. In: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. pp. 10–17. IEEE (2003)
17. Sharma, G., Wu, W., Dalal, E.N.: The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. Color Research and Application 30(1), 21–30 (2005)
18. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
19. Unnikrishnan, R., Pantofaru, C., Hebert, M.: Toward objective evaluation of image segmentation algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(6), 929–944 (Jun 2007)
20. Wang, X., Li, H., Bichot, C.E., Masnou, S., Chen, L.: A graph-cut approach to image segmentation using an affinity graph based on l0-sparse representation of features. In: IEEE International Conference on Image Processing 2013 (ICIP 2013). pp. 4019–4023. Melbourne, Australia (Sep 2013)