Natural Scene Image Segmentation Based on Multi-Layer Feature Extraction

Fariba Zohrizadeh, Mohsen Kheirandishfard, and Farhad Kamangar
University of Texas at Arlington

Abstract. This paper addresses the problem of natural image segmentation by extracting information from a multi-layer array which is constructed based on color, gradient, and statistical properties of the local neighborhoods in an image. A Gaussian Mixture Model (GMM) is used to improve the effectiveness of local spectral histogram features. Grouping these features leads to forming a rough initial over-segmented layer which contains coherent regions of pixels. The regions are merged by using two proposed functions for calculating the distance between two neighboring regions and making decisions about their merging. Extensive experiments are performed on the Berkeley Segmentation Dataset to evaluate the performance of our proposed method and compare the results with the recent state-of-the-art methods. The experimental results indicate that our method achieves higher level of accuracy for natural images compared to recent methods.

Keywords: Image segmentation, Feature extraction, Gaussian mixture model, Distance function

1 Introduction

Image segmentation is a low-level central problem in image processing and computer vision. The primary goal of image segmentation is to automatically partition a single image into locally coherent regions. The partitioned image can be used in the high level applications such as object recognition, scene analysis, and target tracking.

Although recent proposed methods have addressed many of the problems in image segmentation, some challenges still remain. Two of the major issues in image segmentation are encountered when an image contains pixels with extremely diverse colors or when it contains complicated visual textures. Natural images usually present both of these problems, mainly because nature scenes inherently contain different colors and ambiguous visual textures. As a consequence, natural image segmentation is considered as a hot topic in recent publications.

Many studies have tried to tackle the segmentation of natural images by fusing the visual properties of pixels and therefore reducing the color complexity of image. One such kind of method is based on using the concept of superpixel. A superpixel is a group of neighboring pixels that share similar visual attributes. The Simple Linear Iterative Clustering (SLIC) is one of the most popular
superpixel generator algorithms. One of the main advantages of the SLIC algorithm is that it can generate high quality, boundary-preserving and uniform superpixels with an efficient implementation. Recently, Yao [2] has proposed a coarse-to-fine optimization technique to improve the performance of SLIC. The technique iteratively updates the boundary blocks at each level based on an efficient energy function. The results have indicated that this algorithm outperforms the original SLIC not only in terms of speed, but also in all other evaluation metrics.

The texture segmentation algorithms are usually based on clustering the features extracted from the local patches [3,4,5,6,7]. These features can be perceived as descriptors to represent the information of local patches. Malik [3] proposed to construct the texture features by applying a filter-bank on the image. In this method each pixel is presented as a multi-dimensional vector corresponding to the outputs of the bank of filters. The vectors are grouped into clusters to form a channel called texton. This channel is adopted as a fundamental layer in the multiscale segmentation technique proposed in [4]. Van de Sande [5] and Larlus [6] have proposed to use the Bag-of-Visual-Words (BoVW) features as a tool to encode the texture information of a local neighborhood. The BoVW features are adopted in [5] to get a rough segmentation of image which is used for object recognition. The main problem associated with the segmentation based on the local patch descriptors is that these methods are not suitable for representing the local neighborhood near the segment boundaries. In order to address this problem, Larlus [6] has proposed to combine a random field to the extracted features to provide local spatial regularization. The random field is defined over the rectangular grid of patches and these patches are combined with a Dirichlet process mixture to shape the larger scale structures in segmentation.

Recently, Yuan [7] has also developed a factorization-based method for texture segmentation based on singular value decomposition (SVD) and nonnegative matrix factorization (NMF). Local Spectral Histogram (LSH) [8] features are computed for each pixel to construct a feature matrix. The feature matrix can be decomposed as a product of a matrix which contains the basic representative features and a weight matrix corresponding to the contribution of each representative feature at a pixel. The texture of different regions can be expressed as a linear combination of these representative features. Final segments are shaped by grouping the pixels into many clusters based on their combination weights.

In this paper, we have proposed a method which seeks to address the problem of natural image segmentation. In the first step, a rough segmentation of input image is created using LSH-based features which are extracted from a multi-layer of input image. In the next steps, the quality of rough segmentation is improved by merging the similar regions to create larger structures. The final segmentation can be obtained by merging the larger structures to form segments. The main contributions of our proposed algorithm can be summarized as follow:

1. We have adopted Gaussian Mixture Model (GMM) posterior probability to quantize each layer of multi-layer into different levels. Since the multi-layer elements are softly assigned to different levels with
a certain probability, the extracted features are more effective than
the original LSH.
2. We have proposed a new function for measuring the distance between
two neighboring regions in one dimension which is mainly based on
the statistic relation between their elements in each separate layer.
3. We have proposed a new function for taking decision about merging
the neighboring regions based on the information extracted from
regions in high dimensional space.

The main advantage of using decision function in high dimensional space is that
it considers the relation between different layers which has not considered in the
distance function. Since the dimension of space has been increased, the regions
are rarely close to each other and those regions which are close to each other
should be merged with high probability. In other words, this decision function
prefers to merge less but merge with high level of confidence.

The rest of this paper is structured as follows. In Section 2, we briefly present
an overview of the related works. Section 3 presents our proposed algorithm
in details. The experimental results can be found in Section 4 and the final
conclusion is presented in Section 5.

2 Related Works

Most of the recent proposed methods for image segmentation can be grouped into
two main categories. The first category, called contour segmentation, includes
the methods which adopt contour detectors to extract the contour cues used for
image segmentation. Arbelaez [4] presented one of the most cited approaches
in contour segmentation. The proposed method includes globalized proba-

bility of boundary (gPb) contour detection algorithm followed by a segmentation
algorithm which uses Oriented Watershed Transform (OWT) and Ultrametric
Contour map (UCM) called (OWT-UCM). Multiple local cues such as color, inten-
sity and texture are combined into a globalization framework using spectral
clustering to detect the contours in gPb. The result of gPb contour detector can
be converted to a hierarchical segmentation using OWT-UCM algorithm. Since
the gPb and OWT-UCM are two separate steps, one can adopt each of them
separately for specific purposes. Isola [9] has proposed an effective algorithm for
detecting the contours based on the statistical dependencies among pixels val-
ues. The OWT-UCM algorithm is then applied on this algorithm to construct
an accurate segmentation map.

The second category, called region segmentation, contains the methods mainly
based on the concept of grouping connected pixels which share the same visual
attributes such as color and texture. In addition to the early approaches such as
Mean Shift (MS) [10], Normalized Cuts (NCuts) [11] and the graph-based re-
gion merging algorithm presented by Felzenszwalb and Huttenlocher (Felz-Hutt)
[12], some of the recent proposed methods [13,17,14] also fall into this category.
Lobacheva [13] has defined a novel energy function which includes color clus-
tering term, appearance model and segmentation boundary term. The proposed
approach jointly optimizes this energy function to make a distinction between the pixels belonging to the background and the ones belonging to the foreground. Segmentation by Aggregating Superpixels (SAS) [14] has used superpixels to encode the complex visual textures of natural images. The SAS has generated superpixels using different techniques (such as MS [10] and Felz-Hutt [12]) to construct multi-layers of superpixels. The layers of superpixels are aggregated in a bipartite graph partitioning framework to get an accurate segmentation. In this method the number of groups which gives the best segmentation result needs to be pre-determined and manually set for each image. It can be considered as a limitation for automatic image segmentation using this method.

Recently, many region segmentation approaches tend to use a hybrid of contour cues and region attributes to effectively decompose an image into coherent regions. Fu [15] has developed a Contour-guided Color Palette (CCP) algorithm for robust image segmentation. The first step of CCP is to create a quantized image by clustering the color samples which are collected along both sides of long contours. This quantized image contains regions which are enclosed by long contours. The final segmentation result is obtained by applying three effective post-processing techniques which are leakage avoidance using contours, fake boundary removal, and small region merging. To get the best segmentation result, the number of segments needs to manually determined, the same as SAS [14]. Liu [16] has proposed to represent the image as a graph where each node represents a superpixel and each edge corresponds to the common boundary of adjacent superpixel. The proposed framework is based on the merging initial superpixels in a specific order to find the final segmentation. A full binary tree structure called hierarchical merge tree is used to represent the priority of merging regions. The probability of each merge is computed using a proposed boundary classifier trained by the boundary and region features. The final segmentation is obtained by optimizing a constrained conditional model constructed from the hierarchical merge tree.

In this paper, we have proposed an effective segmentation algorithm which adopts the contour and region cues to partition the image into different segments. In the first step, a multi-layer is constructed by combining different layers which are obtained from the image responses to different visual process. We have proposed a modification on the LSH features [8] based on GMM posterior probability to extract powerful region-based features from the multi-layer for representing complicated textures in an effective way. Grouping the extracted features into many clusters lead to creating a rough segmentation of image with locally coherent regions. In the second step, some of these regions are merged using some contour-based features introduced in [17] and the post-processing steps of CCP. In the last step, a hierarchical tree is constructed based on a new distance function defined over the neighboring regions. The tree determines the order of merging neighboring regions to create larger segments. Applying a new decision function on the hierarchical tree lead to discovering a set of subtrees, which are corresponding to the final segmentation results.
Since our proposed method mainly rely on the statistical characteristics of adjacent regions, it would be appropriate for segmentation of natural images with ambiguous visual texture and complicated color space.

3 Proposed Method

3.1 Feature Extraction

Given an image $I$ of size $N_x \times N_y$, let define a multi-layer array $M : M^j, j = 1, \ldots, H$ which is constructed from the color, gradient and texture information layers extracted from the original image. In the proposed method, the multi-layer array, $M$, includes five layers based on color information, five layers based on image gradients, three layers based on soft segmentation proposed in [18], and one layer based on the texture. The five color layers contain $L$, $a$, $b$ of CIE-LAB color space and the $C_b, C_r$ components of $YC_bC_r$ color space. The five gradient based layers include three layers obtained by applying Laplacian of Gaussian (LoG) filters with different size $h$ and standard deviation $\sigma : (h, \sigma) : (3, 0.5), (5, 0.8), (7, 1.2)$; and two layers obtained from magnitude of gradient of blurred intensity image with two Gaussian filters $(h, \sigma) : (5, 0), (5, 1.5)$. Three layers are calculated by applying the PCA to soft segmentation layers presented in [18] and selecting the first three components. The last layer is the texton proposed by Malik [3]. In this multi-layer array each pixel in the original image is presented by a vector in a 14 dimensional space.

In order to encode the information of a local neighborhood in each layer, we propose a modification on the LSH features defined by Liu in [8]. Liu suggested to quantize the elements of each layer into $K$ different equal levels $L_k, k = 1, \ldots, K$. As a consequence, each element of this multi-dimensional presentation is expressed by an integer number between 1 and $K$. To effectively compute the local histogram of each pixel, Liu proposed to represent each element as $e_k \in R^{K \times 1}, k = 1, \ldots, K$ where $e_k$ denotes a vector with a 1 in the $k^{th}$ element and 0 elsewhere. In other words, the elements are strictly assigned to different levels based on their values. In order to reduce the quantization error, we represent the values in each layer using a GMM distribution with $K$ components where the center of each component corresponds to the center of a specific quantization level. Figure 1a shows our quantization and LSH quantization on a layer of multi-layer $M$.

$$P(x) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(x | \mu_k, \sigma_k) \quad (1)$$

where $\alpha_k$, $\mu_k$ and $\sigma_k$ indicate the prior probability, mean, and standard deviation of the $k^{th}$ component of GMM respectively. It should be noted that in our implementation the prior, $\alpha_k$, for all the GMM components are assumed to be equal and set to $\frac{1}{K}$ and the standard deviation of each component, $\sigma_k$, is set to one third of the width of the quantization level. Using the GMM posterior probability each element in a specific layer, $M^j$, is represented as a vector
in $R^{K \times 1}$. Where the value of the $k^{th}$ dimension of this vector represents the posterior probability of the element belonging to the $k^{th}$ quantization level. Assuming that $C_i = \{c_1^i, \cdots, c_H^i\}$ represents the vector corresponding to pixel $p_i$ in the multi-layer representation of the image, the probability of $c_j^i$ belonging to $k^{th}$ quantization level of $M^j$ is represented by

$$P(c_j^i \in L_k^j | c_i^j) \propto \mathcal{N}(c_j^i | \mu_k^j, \sigma_k^j)$$ (2)

The posterior probability vector of the $j^{th}$ layer of $M$ can be defined as $q_i^j \in R^{K \times 1}, q_i^j = \langle P(c_j^i \in L_1^j | c_i^j), P(c_j^i \in L_2^j | c_i^j), \cdots, P(c_j^i \in L_K^j | c_i^j) \rangle$ which contains the probability of belonging $c_j^i$ to different levels of $M^j$. The vector $Q_i \in R^{H \times K \times 1}$ is created for each pixel $p_i$ by combining the posterior vectors of all layers into a single vector $Q_i = \langle q_1^i, q_2^i, \cdots, q_H^i \rangle$.

Fig. 1: (a) LSH represents these points $A$ (red) and $B$ (yellow) with completely different vectors $(e_1$ and $e_2$), however, MLSH represents them with two nearly similar vectors (b) One dimensional space (c) Two dimensional space

At this stage the image is represented by a multi-layer array with the size of $N_x$ by $N_y$ by $(H \times K)$ where each pixel $p_i$ corresponds to a vector $Q_i$. To extract the information of a local neighborhood around pixel $p_i$, the vector $Q_i$ is substituted by $H_{W_i}$ which is computed over a window $W_i$ centered at $p_i$.

$$H_{W_i} = \frac{1}{|W_i|} \sum_{l \in S} Q_l$$ (3)

where $S$ is a set of all pixels within the window $W_i$ and $|.|$ indicates the number of pixels inside $W_i$. For simplicity, the $H_{W_i}$ obtained in this step is called MLSH (which is a abbreviation for Modified LSH) and the multi-layer created by MLSH features is called $M_{MLSH} \in R^{N_x \times N_y \times (H \times K)}, M_{MLSH}^j, j = 1, \cdots, (H \times K)$.

Integral image can be adopted to reduce the complexity of Equation (3). Further details of using integral image can be found in [8].
3.2 Rough Segmentation

Using the GMM clustering method implemented in [19], the MLSH features are grouped into $N_c$ clusters to provide a rough segmentation of image. Each cluster corresponds to a specific representative segment of the image. A common problem at this step of segmentation is that the pixels which are located close to the boundary of segments may form new clusters which are different from both neighboring segments. In order to prevent the creation of these redundant clusters, we do not consider MLSH features of those pixels which are closer than $\delta$ to the strong contours in the clustering process. The counter map used in this process is generated by [17]. The pixels which were excluded from the clustering process are then assigned to a segment based on the maximum posterior probability of membership to a neighboring cluster. Figure 2 shows rough segmentation obtained by MLSH and LSH.

Fig. 2: Comparing the effect of using MLSH on rough segmentation (a) Original Image, (b) MLSH, (c) LSH

In the next step, the result of the rough segmentation is superimposed on a superpixel segmentation generated by the algorithm described in [17]. Those superpixels which are completely within a particular segment in the rough segmentation, are merged to create larger regions. The post-processing techniques in CCP are also used to merge the small regions, remove the fake boundaries and avoid the leakage problem. The precise details of these techniques can be found in [15].

Although these steps improve the segmentation process, the result is still not satisfactory, which is mainly because we do not have any a-priori information about the optimum number of segments for each image. In the next section, we propose a novel post-processing step to merge similar regions and improve the segmentation results.

3.3 Region Merging

The first step of our merging technique is to construct a hierarchical tree for representing the merging priority of adjacent regions. This tree is constructed based on a weighted graph $G = (\mathcal{N}, \mathcal{E}, \mathcal{W})$ where nodes, $\mathcal{N}$, correspond to the
regions, and edges, $E$, connect the neighboring regions. The edge weights, $W$, are calculated based on a novel distance function between the neighboring regions.

Let $R_1^j = \{x_1^j, x_2^j, \ldots, x_n^j\}$ and $R_2^j = \{y_1^j, y_2^j, \ldots, y_m^j\}$ represent two sets of values corresponding to the two neighboring regions $R_1$ and $R_2$ in layer $j$, i.e., $M^j_{MLSH}$. The means and standard deviations of $R_1^j$ and $R_2^j$ are also shown as $\mu_{R_1^j}, \mu_{R_2^j}$ and $\sigma_{R_1^j}, \sigma_{R_2^j}$ respectively. The region obtained by merging these two regions is defined as $R_1^j \cup R_2^j = R_1^j \cup R_2^j = \{x_1^j, \ldots, x_n^j, y_1^j, \ldots, y_m^j\}$ with mean and standard deviation $\mu_{R_1^j \cup R_2^j}$ and $\sigma_{R_1^j \cup R_2^j}$. As shown in the Figures 1b, 1c if there is no overlap between the values in the $R_1^j$ and $R_2^j$ then $\sigma_{R_1^j \cup R_2^j} \geq \sigma_{R_1^j} + \sigma_{R_2^j}$, however if there is a lot of overlap between the values in these regions then $\sigma_{R_1^j \cup R_2^j} \leq \sigma_{R_1^j} + \sigma_{R_2^j}$. Our proposed dissimilarity function is defined based on this fact. The distance between the two regions $R_1^j$ and $R_2^j$ is defined as:

$$dist(R_1, R_2) = \frac{average(ST_{(R_1, R_2)})}{C \times \sum_{j \in S} v_{R_1, R_2}^j}$$

(4)

where $average(ST_{(R_1, R_2)})$ is the average of strength boundary pixels in contour map presented by [17], $S = \{j|(\sigma_{R_1^j} + \sigma_{R_2^j}) \geq \sigma_{R_1^j \cup R_2^j}\}$, and $C = \sum_{j=1}^{(H \times K)} v_{R_1, R_2}^j$ is used to make the weights add up to one, $\gamma$, and $v_{R_1, R_2}^j$ is defined as:

$$v_{R_1, R_2}^j = \frac{|(\sigma_{R_1^j} + \sigma_{R_2^j}) - \sigma_{R_1^j \cup R_2^j}|}{(\sigma_{R_1^j} + \sigma_{R_2^j}) \times (\sigma_{R_1^j \cup R_2^j})}$$

(5)

$|.|$ denotes the absolute value function and $v_{R_1, R_2}^j$ is the $j^{th}$ element of $V_{R_1, R_2} \in R^{(H \times K) \times 1}$ which represents the weights in all multi-layer $M_{MLSH}$. The hierarchical tree is constructed by using the distance function presented in Equation (4) in the Algorithm 1.

**Algorithm 1 Hierarchical Tree**

1: procedure **ConstTree($M_{MLSH}, ST, G(N, E, W)$)**  
2: \hspace{1em} $T \leftarrow N$  \hspace{6em} $\triangleright$ Insert all nodes into a tree  
3: \hspace{1em} while $|N| > 1$ do  \hspace{6em} $\triangleright$ do while all nodes are merged  
4: \hspace{2em} $(N_1^*, N_2^*) \leftarrow \arg\min_{(N_1, N_2) \in N} distance(M_{MLSH}, ST, N_1, N_2)$  \hspace{6em} $\triangleright$ Find closest pair  
5: \hspace{1em} $N \leftarrow N \setminus N_1^* \setminus N_2^*$  \hspace{6em} $\triangleright$ Remove them from list of nodes  
6: \hspace{1em} $\mathcal{T} \leftarrow \mathcal{T} \setminus (N_1^*, N_2^*)$  \hspace{6em} $\triangleright$ Insert merged node to the list of nodes  
7: \hspace{1em} $T \leftarrow T \cup \{N_1^*, N_2^*\}$  \hspace{6em} $\triangleright$ Insert merged node to the tree  
8: \hspace{1em} end while  
9: return $T$  
10: end procedure

The tree which is constructed using Algorithm 1 determines the order of region merging based on the minimum distance between two neighboring regions.
Other distance functions may also be used to create a merging tree. Arbelaez [4] and Liu [16] have proposed to measure the distance between $R_1$ and $R_2$ using $\text{average}(Pb(R_1,R_2)) \text{ and } 1 - \text{median}(Pb(R_1,R_2))$, respectively. Where $Pb(R_1,R_2)$ is a set of values corresponding to the common boundary of the two adjacent regions $R_1$ and $R_2$ in contour map. A typical problem associated with using the algorithms in [4] and [16] is that when two regions are different but do not have a strong common boundary, then these algorithms assign a high priority to merge these regions, even though the regions may belong to different segments. In our proposed algorithm, if $\sigma_{R_1 \cup R_2}$ is greater than $(\sigma_{R_1} + \sigma_{R_2})$, the denominator of Equation (4) becomes small and the distance between $R_1$ and $R_2$ is increased, even if the contour detector could not detect the boundary between the two regions. On the other hand, if $R_1$ and $R_2$ have similar properties and belong to the same segment, then the denominator of Equation (4) tends to be near 1 and decision is made mostly based on the nominator.

Using the hierarchical tree, we develop a novel function to take decision about merging the neighboring regions. Given $S$, let define $S^c = \{ j | (\sigma_{R_1} + \sigma_{R_2}) \leq \sigma_{R_1 \cup R_2} \}$. The weights over these sets are also defined by $V^+_{R_1,R_2} = \frac{1}{C} \times \sum_{j \in S} v^+_j R_1, R_2$, $V^-_{R_1,R_2} = \frac{1}{C} \times \sum_{j \in S^c} v^-_j R_1, R_2$. The degree of confidence for the decision represents as:

$$\text{ratio}_{V_{R_1,R_2}} = 1 - \frac{\min(V^+_{R_1,R_2}, V^-_{R_1,R_2})}{\max(V^+_{R_1,R_2}, V^-_{R_1,R_2})}$$

(6)

where $\text{ratio}_{V_{R_1,R_2}}$ gets values between 0 and 1. Larger values are corresponding to the firm decision (merge/not-merge).

We also propose another decision function which takes the decision based on the statistical characteristics of two regions in the high dimensional space. Let $M^R_{M^2L^L} = \{H^R_{W_1}, H^R_{W_2}, \ldots, H^R_{W_N}\}$ and $M^R_{M^2L^L} = \{H^R_{W_1}, H^R_{W_2}, \ldots, H^R_{W_N}\}$ be two sets with equal number of features randomly sampled from $R_1$ and $R_2$ in $M^R_{M^2L^L}$. For a typical region $R$ which represents by set $M^R_{M^2L^L}$ with $N$ elements, distance variance [20] is defined over elements as a scalar metric for representing the dispersion of the set in multi-dimensional space.

$$\mathcal{V}_N^2(M^R_{M^2L^L}) = \frac{1}{N^2} \sum_{k,l=1}^{N} A^2_{kl}$$

(7)

where,

$$A_{kl} = a_{kl} - \frac{1}{N} \sum_{l=1}^{N} a_{kl} - \frac{1}{N} \sum_{k=1}^{N} a_{kl} + \frac{1}{N^2} \sum_{k,l=1}^{N} a_{kl}$$

(8)

and $a_{kl} = \|H^R_{W_k} - H^R_{W_l}\|$, where $\|\cdot\|$ denotes the Euclidean norm.

Using the distance variance, we propose a new decision function over $R_1$ and $R_2$. Our function is defined by extending the concept of Figure 1c to the multi dimensional space. As a consequence, the neighboring regions $R_1$ and $R_2$ should
merge when $V_N^2(M_{MLSH}^{R_1 \cup R_2}) \leq V_N^2(M_{MLSH}^{R_1}) + V_N^2(M_{MLSH}^{R_2})$ and they should not be merged when $V_N^2(M_{MLSH}^{R_1 \cup R_2}) \geq V_N^2(M_{MLSH}^{R_1}) + V_N^2(M_{MLSH}^{R_2})$. The degree of confidence for this decision is also defined as:

$$ratio_{V_N^{R_1,R_2}} = 1 - \frac{\min(V_N^2(M_{MLSH}^{R_1}) + V_N^2(M_{MLSH}^{R_2}), V_N^2(M_{MLSH}^{R_1 \cup R_2}))}{\max(V_N^2(M_{MLSH}^{R_1}) + V_N^2(M_{MLSH}^{R_2}), V_N^2(M_{MLSH}^{R_1 \cup R_2}))}$$ (9)

where $ratio_{V_N^{R_1,R_2}}$ gets values between 0 and 1. Larger values are corresponding to the firm decision (merge/not-merge). The final decision is taken based on a linear combination of Equation (6) and Equation (9).

$$F(R_1, R_2) = (\alpha)(d_\alpha)(ratio_{V_N^{R_1,R_2}}) + (1 - \alpha)(d_{(1-\alpha)})(ratio_{V_N^{R_1,R_2}})$$ (10)

where $\alpha$ is a threshold which determines the importance of each decision function on the final decision function, $d_\alpha$ and $d_{(1-\alpha)}$ correspond to the decision taken by $ratio_{V_N^{R_1,R_2}}$ and $ratio_{V_N^{R_1,R_2}}$ respectively. It should note that $ratio_{V_N^{R_1,R_2}}$ and $ratio_{V_N^{R_1,R_2}}$ just represent the degree of confidence about decisions. The parameters $d_\alpha$ and $d_{(1-\alpha)}$ are defined to represent the decisions and get +1 for merge and -1 for not-merge. Two neighboring regions are merged if the result of Equation (10) is positive, and they will not merge if the result is negative.

we start merging procedure from the bottom of the tree by labeling all leaf nodes to 1. The parents are labeled to 1 or 0 for merge or not-merge based on the result of Equation (10). This procedure is repeated until root node is also labeled.

Final segmentation of image is obtained by finding those subtrees which all their nodes are labeled 1, however, their parents are labeled 0. These subtrees are corresponding to the final segments.

4 Experimental Results

In this section, we present the setup of our algorithm to perform a practical experiment on natural images. We also compare our results with the results of some recent proposed algorithms on the same data set.

4.1 Setup

Data Set: we report the result of our segmentation algorithm on the Berkeley Segmentation Data Set (BSDS300) [21], which is one of the most commonly used data set for natural image segmentation. The BSDS300 contains 300 images of size $481 \times 321$ captured from different natural scenes. Images are manually segmented by human in different levels of detail to construct multiple ground truth segmentation for each image.
Parameters: the required parameters in the feature extraction step of our algorithm are the number of quantization levels, $K$, and the width of window, $W$, which have set 11 and 27 respectively. In rough segmentation step, we have set $\delta = 13$ and the number of clusters, $N_c$, are considered equal to 10. The required parameters in the regions merging are, $\alpha$ in Equation (10), $N$ in Equation (7) which get 0.78 and $0.2 \times \min(|R_1|, |R_2|)$ respectively, where $|.|$ represents the number of pixels.

4.2 Evaluation

Metrics: there are a number of standard evaluation metrics which are used for measuring the quality of segmentation results. We have used five metrics, Covering (Cov) [4], Probabilistic Rand Index (PRI) [22], Variation of Information (VoI) [23], Global Consistency Error(GCE) [21], and Boundary Displacement Error (BDE) [24] to evaluate our algorithm.

Cov measures the average per pixel matching between a segmentation result, $S_t$ and a ground truth segmentation, $S_g$.

$$C(S_t, S_g) = \frac{1}{N} \sum_{R_t \in S_t} |R_t| \max_{R_g \in S_g} O(S_t, S_g)$$  \hspace{1cm} (11)

where $R_t$ and $R_g$ are two segments in $S_t$ and $S_g$, and $O(R_t, R_g)$ represents the overlap between these two regions. The symbol $|.|$ indicates the cardinality of pixels within a region and $N$ is the total number of image pixels. Further details about Cov is provided in [4].

PRI is a metric for measuring the similarity between two different clusterings of the same data set. In image segmentation, PRI measures the relationship between pairs of pixels in $S_t$ and $S_g$

$$PRI(S_t, S_g) = \frac{1}{\binom{N}{2}} \sum_{i<j} \gamma(p_i, p_j)$$  \hspace{1cm} (12)

where $N$ is the number of image pixels and $\gamma(.)$ is a function which returns 1 if the pairs are assigned to the same segment in $S_t$ and $S_g$ or assigned to different segments in both $S_t$ and $S_g$, otherwise it returns 0. This measure is normalized by the total number of all possible pixel pairs, $\binom{N}{2}$.

VoI is used to measure the distance between two different clusterings of the same data set. The distance is defined based on the information difference between the two clusterings of data.

$$VoI(S_t, S_g) = H(S_t) + H(S_g) - 2I(S_t, S_g)$$  \hspace{1cm} (13)

where $H$ and $I$ show the entropy and mutual information, respectively.

GCE is used to measure the value of local refinement error over all image pixels.

$$GCE(S_t, S_g) = \frac{1}{N} \min \left\{ \sum_i E(S_t, S_g, p_i), \sum_i E(S_g, S_t, p_i) \right\}$$  \hspace{1cm} (14)
where \( E(S_t, S_g, p_i) \) and \( E(S_g, S_t, p_i) \) represent local refinement error at pixel \( p_i \) in two different directions. Further details about the GCE can be found in [21].

BDE measures the average displacement error of boundary pixels between two different segmentations of an image. The displacement error of a boundary pixel \( p_i \in S_t \) from \( S_g \) is defined as
\[
d_E(p_i, S_g) = \min_{y \in S_g} d_E(p_i, y),
\]
where \( d_E(.) \) indicates the Euclidean distance function.

\[
BDE(S_t, S_g) = \frac{1}{2} \left( \frac{1}{|B_t|} \sum_i d(p_i \in B_t, S_g) + \frac{1}{|B_g|} \sum_i d(p_i \in B_g, S_t) \right)
\]  

where \( B_t \) and \( B_g \) are the set of boundary pixels of \( S_t \) and \( S_g \), \( |.| \) indicates the cardinality of a set.

Discussion: we have reported our results on BSDS300 to make a comparison with leading methods MS [10], NCuts [11], Felz-Hutt [12], SAS [14], CCP [15], gPb-OWT-UCM [4], and Fact-LSH [7]. The evaluation numbers of SAS has obtained by running the source code provided on the author’s webpage. The evaluation numbers of gPb-OWT-UCM and Fact-LSH have obtained from their original papers [4,7] (gPb-OWT-UCM [4] has not reported BDE and GCE, Fact-LSH [7] has not reported Cov, BDE and GCE) and the evaluation numbers for MS [10], NCuts [11], Felz-Hutt [12], and CCP [15] have directly extracted from CCP [15].

Table 1 contains the evaluation results of these methods, in terms of the Cov, PRI, VoI, GCE and BDE metrics. As it can be seen, our method has achieved excellent performance in terms of BDE and GCE metrics, while it has preserved the amount of other metric near the best. Since our proposed method does not have any image-dependent parameter, we have compared our result with those methods which have the same parameter setting. The CCP have reported five different results which three of them are not based on the image-dependent parameters. In Table 1, we have used one of those versions called CCP-2. The SAS algorithm have not reported any results with fixed parameter setting, and the reported results have obtained by manually setting parameters for each image in dataset.

In order to evaluate the effect of using MLSH in our method, we have conducted two different experiments using MLSH and LSH features. The results show that using our modification can effectively improve the performance in terms of Cov, PRI, GCE, and BDE. It should be noted that the parameters of our method have been set to get a satisfactory result on all metrics and they have not optimized on any specific metric. As it has mentioned earlier, the \( ratio_{V_2, R^2_1} \) tends to merge few number of regions which most of them are correct. On the other hand, the \( ratio_{V_1, R^2_2} \) tends to merge more regions, but, it does not have accuracy as high as \( ratio_{V_2, R^1_2} \). The final result of our proposed algorithm is mainly based on these decision functions. Increasing the effect of \( ratio_{V_2, R^2_1} \) using \( \alpha \) leads to improving the BDE and GCE, however, the performance is reduced over VoI and PRI. Decreasing the effect of \( ratio_{V_2, R^2_1} \) using \( \alpha \) leads to
improving VoI, Cov and PRI, however, the performance is reduced in terms of BDE and GCE.

The results of the mentioned algorithms have provided in Figure 3 to make a qualitative comparison. It can be seen that our method outperforms the other methods in most cases, especially in images with complicated texture.

Table 1: Quantitative comparison on BSDS300. The best three results are highlighted with red, green, and blue colors respectively. All results have reported based on Optimal Dataset Scale (ODS), except SAS which has reported based on Optimal Image Scale (OIS)

| Algorithms            | Cov  | PRI  | VoI  | GCE  | BDE  |
|-----------------------|------|------|------|------|------|
| NCuts[11]             | 0.44 | 0.7242 | 2.9061 | 0.2232 | 17.15 |
| MS[10]                | 0.54 | 0.7958 | 1.9725 | 0.1888 | 14.41 |
| Felz-Hutt[12]         | 0.51 | 0.7139 | 3.3949 | 0.1746 | 16.67 |
| SAS[14]               | 0.594 | 0.8305 | 1.6992 | 0.1811 | 11.45 |
| CCP[15]               | 0.48 | 0.7932 | 2.7835 | 0.1077 | 11.17 |
| gPb-OWT-UCM[4]        | 0.59 | 0.81 | 1.65  | —     | —     |
| Fact-LSH[7]           | —   | 0.79 | 2.1   | —     | —     |
| Our Algorithm+LSH     | 0.57 | 0.8034 | 1.8113 | 0.1733 | 13.68 |
| Our Algorithm+MLSH    | 0.59 | 0.817 | 1.8864 | 0.1541 | 10.99 |

5 Conclusions

We have presented a novel algorithm for natural image segmentation by combining multiple layers of information obtained from the original image in a systematic manner to obtain a locally coherent segmentation of image. We have proposed to describe the complicated image textures using a modified version of the Local Spectral Histogram features which is one of the most well-known texture descriptors. Our main contributions are presenting two simple and effective functions for merging those neighboring regions which have nearly the same visual properties. The first criteria has used to create a hierarchical tree which determines the priority of region merging and the second one has used for developing a decision function to shape the final segments by merging similar regions. The extensive experiments of our algorithm on Berkeley Segmentation Dataset has illustrated the proposed method can effectively reach to the high level of accuracy in terms of evaluation metrics and visual quality. Given the fact that our method does not need to any prior information about the image content, our results can be used in higher level applications in computer vision such as object recognition and scene analysis.
Fig. 3: Qualitative comparison results: (a) Original; (b) Felz-Hutt [12]; (c) MS [10]; (d) SAS [14]; (e) Fact-LSH [7]; (f) CCP [15]; (g) Our Algorithm+MLSH
1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. Pattern Analysis and Machine Intelligence, IEEE Transactions on 34(11) (2012) 2274–2282
2. Yao, J., Boben, M., Fidler, S., Urtasun, R.: Real-time coarse-to-fine topologically preserving segmentation. Energy 2 (2015) 2–3
3. Malik, J., Belongie, S., Shi, J., Leung, T.: Textons, contours and regions: Cue integration in image segmentation. In: Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on. Volume 2., IEEE (1999) 918–925
4. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33(5) (2011) 898–916
5. Van de Sande, K.E., Uijlings, J.R., Gevers, T., Smeulders, A.W.: Segmentation as selective search for object recognition. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 1879–1886
6. Larlus, D., Verbeek, J., Jurie, F.: Category level object segmentation by combining bag-of-words models with dirichlet processes and random fields. International Journal of Computer Vision 88(2) (2010) 238–253
7. Yuan, J., Wang, D., Cheriyadat, A.M.: Factorization-based texture segmentation. Image Processing, IEEE Transactions on 24(11) (2015) 3488–3497
8. Liu, X., Wang, D.: Image and texture segmentation using local spectral histograms. Image Processing, IEEE Transactions on 15(10) (2006) 3066–3077
9. Isola, P., Zoran, D., Krishnan, D., Adelson, E.H.: Crisp boundary detection using pointwise mutual information. In: Computer Vision–ECCV 2014. Springer (2014) 799–814
10. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. Pattern Analysis and Machine Intelligence, IEEE Transactions on 24(5) (2002) 603–619
11. Shi, J., Malik, J.: Normalized cuts and image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22(8) (2000) 888–905
12. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. International Journal of Computer Vision 59(2) (2004) 167–181
13. Lobacheva, E., Veksler, O., Boykov, Y.: Joint optimization of segmentation and color clustering. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1626–1634
14. Li, Z., Wu, X.M., Chang, S.F.: Segmentation using superpixels: A bipartite graph partitioning approach. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 789–796
15. Fu, X., Wang, C.Y., Chen, C., Wang, C., Jay Kuo, C.C.: Robust image segmentation using contour-guided color palettes. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1618–1625
16. Liu, T., Seyedhosseini, M., Tasdizen, T.: Image segmentation using hierarchical merge tree. arXiv preprint arXiv:1505.06389 (2015)
17. Dollár, P.: Piotr’s Computer Vision Matlab Toolbox (PMT). http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html
18. Leordeanu, M., Sukthankar, R., Sminchisescu, C.: Generalized boundaries from multiple image interpretations. Pattern Analysis and Machine Intelligence, IEEE Transactions on 36(7) (2014) 1312–1324
19. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms. [http://www.vlfeat.org/](http://www.vlfeat.org/) (2008)

20. Székely, G.J., Rizzo, M.L., Bakirov, N.K., et al.: Measuring and testing dependence by correlation of distances. The Annals of Statistics 35(6) (2007) 2769–2794

21. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. Volume 2., IEEE (2001) 416–423

22. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66(336) (1971) 846–850

23. Meil, M.: Comparing clusterings: an axiomatic view. In: Proceedings of the 22nd international conference on Machine learning, ACM (2005) 577–584

24. Freixenet, J., Muñoz, X., Raba, D., Martí, J., Cufí, X.: Yet another survey on image segmentation: Region and boundary information integration. In: Computer Vision ECCV 2002. Springer (2002) 408–422