Bottom-Up Merging Segmentation for Color Images With Complex Areas

Haifeng Sima, Ping Guo, Senior Member, IEEE, Youfeng Zou, Zhiheng Wang, and Mingliang Xu

Abstract—Most color images obtained from the real world usually contain complex areas, such as nature scene images, remote sensing images, and medical images. All these type of images are very difficult to be separated accurately and automatically for complex color and structures included. In this paper, we focus on detecting hybrid cues of color image to segment complex scene in a bottom-up framework. The main idea of the proposed segmentation method is based on a two-step procedure: 1) a reasonable superpixels computing method is conducted and 2) a Mumford–Shah (M–S) optimal merging model is proposed for presegment superpixels. First, a set of seed pixels is positioned at the lowest texture energy map computed from structure tensor diffusion features. Next, we implement a growing procedure to extract superpixels from selected seed pixels with color and texture cues. After that, a color-texture histograms feature is defined to measure similarity between regions, and an M–S optimal merging process is executed by comparing the similarity of adjacent regions with standard deviation constraints to get final segmentation. Extensive experiments are conducted on the Berkeley segmentation database, some remote sensing images, and medical images. The results of experiments have verified that the segmentation effectiveness of the proposed method in segmenting complex scenes and indicated that it is more robust and accurate than conventional methods.

Index Terms—Color-texture histograms, image segmentation, Mumford–Shah (M–S) merging, structure tensor diffusion, superpixels.

I. INTRODUCTION

I

MAGE segmentation is a basic vision task that facilitates image analysis and understanding, and it is a fundamental and crucial step in many computer vision applications. The primary aim of segmentation is to divide an image into meaningful and spatially connected regions (such as object and background) on the basis of similar properties. Existing segmentation approaches can predominantly be divided into the following four categories: 1) region-based methods [1]–[3]; 2) feature-based methods [4]; 3) boundary-based methods [5], [6]; and 4) model-based methods [7]–[10]. It is difficult to define a perfect assessment criterion to measure segmentation quality, because there is no universal solution for color image segmentation. In most applications, image segmentation methods are designed to adapt to specific applications and requirements.

In the segmentation of images with complex scenes, superpixels extraction is an ideal preprocessing technique to simplify the segmentation task. A superpixel is defined as a set of connected homogeneous image region, which have attracted increasingly attention in recent years. Various types of superpixels extraction algorithms differ in solving measures which result in different presentations including shape, areas, boundaries, compactness, etc. For further task, the boundaries precision is the most important index regarded in this paper as a preparations commission for merging stage.

The region merging process is the critical step in bottom-up segmentation approaches. The merging technique is region-based manipulations set by making decision of merging region via region properties and features. This process starts from an initial over-segmentation (superpixels) and all these superpixels are iteratively merged until a given termination criterion is satisfied. Thus, effective superpixels computing method and merging strategy are the key issues of complex scenes segmentation. The goal of this paper is thereby to find ideal method to extract superpixels with accurate boundaries and establish a consistent optimal region merging model for the bottom-up segmentation framework.

In this paper, we propose a two-step segmentation method based on the pixel growing and Mumford–Shah (M–S) optimal merging. First, the texture information is obtained by total variation flow (TV flow) diffusion on structure tensors. Next, we select seeds in a texture gallery and make them grow to be uniform superpixels according to similarity of color and texture features. Then we define a combined color and texture descriptor for superpixels based on local color pattern (LCP) and design a standard deviation-based M–S minimization merging method to gather adjacent and homogeneous superpixels as final segmentation regions. A flowchart of our proposed

method is depicted in Fig. 1. The main contribution of this paper are as follows.

1) The total variation diffusion of structure tensor is developed on three channels for computing texture features and pixels growing method is first used for producing superpixels that contain complete texture and accurate boundaries using mixture features.

2) In the second stage, we establish a two-tier merging criterion by M–S model and standard deviation of region features to control region merging. We design a combined histogram feature by combining color statistical characteristic and distribution information in a mixture form to describe commonality and distinction of the superpixels. These two measures are very effective for distinguishing homogeneous and inhomogeneous superpixels in the merging process that leading to accurate segmentation.

The remainder of this paper is organized as follows. In Section II, we describe the related work. Section III describes the proposed texture detection and growing strategy to obtain the superpixels. Section IV describes the basic concept of the color-texture feature and merging scheme in detail. Section V describes extensive experiments conducted on the Berkeley segmentation dataset (BSD300) and some selected images to test and validate the efficiency of our method. Section VI summarizes and concludes this paper.

II. RELATED WORK
A number of superpixels-based segmentation methods have shown good performance. Yang et al. [26] proposed a compression-based texture merging (CTM) segmentation method based on superpixels. The method uses a slide window to extract texture features in each channel of a Laboratory color image in superpixels, and reduces the color-texture feature to 8-D by principal component analysis. Then, it uses a mixture Gaussian degradation distribution to describe texture features and uses data clustering to merge superpixels for image segmentation. Its segmentation performance significantly depends on the texture difference parameter, and it is very time-consuming when texture feature computing is conducted. Xie et al. [15] proposed a Laplacian sparse subspace clustering for salient object extraction based on similarity between superpixels. Li et al. [35] proposed
a novel segmentation framework based on bipartite graph partitioning that is able to aggregate multilayer superpixels in a principled and effective way, which can lead to fast segmentation of images. However, there are limitations. First, it needs to predefine segment numbers. Second, texture images are poorly displayed. GL-graph [36] refined the B-graph using medium-sized superpixels through a sparse representation of superpixels features by solving a $\ell_0$-minimization problem. Taylor [16] explored approaches to accelerating segmentation and edge detection algorithms based on the gPb framework, it invokes a globalization procedure which captures the essential features based on an analysis of the normalized cuts eigenvectors.

The superpixels extraction methods usually divide an image into hundreds of homogeneous segments instead of many redundant pixels. These methods gather homogeneous pixels that meet corresponding demands. The most popular methods are based on graph-cut [9], Lattice cut [23], local clustering [22], geometric flows [24], etc. Graph-cut [9] tries to segment homogeneous connection regions based on the principle of max-flow/min-cut, and it performs poorly on texture regions. Lattice cut [23] employs a supervised binary segmentation method to conduct unsupervised over-segmented superpixels based on an alternating optimization strategy. Local clustering [22] chooses cluster centers from uniform blocks, and each pixel is assigned to a similar superpixel by linear iterative clustering algorithm with color and coordinates feature. Geometric flows [24] is an effective superpixels extraction method by evolution contour model. Besides, some novel models are designed for superpixels computing. Linear spectral clustering (LSC) [25] uses a kernel function leading to an explicitly mapping of pixel values and coordinates into a high dimensional feature space and optimizes the cost function of normalized cuts by iteratively applying simple $K$-means clustering in the proposed feature space. Yang et al. [26] proposed a $K$-means-like clustering method applied to the RGB-D data for over-segmentation using an 8-D distance metric constructed from both color and 3-D geometrical information. As the first step of images segmentation, its first job is to obtain homogeneous regions and accurate boundaries but all these methods perform inadequately at the boundaries detection on images with complex scenes. For color images with complex scenes, color and texture are the basic and essential visual features which are used to separate these complex areas. Many researches have paid attention to improving their adaptability on nonuniformity of color distribution, input noise, and textures discrimination [26]. Rousson et al. [21] proposed TV flow to describe texture characteristics. The method is not only able to express the details of the texture features including direction and magnitude, but also can estimate local scale information of texture by variation evolution. TV flow also makes up for the loss of texture scale in features coming from the structure tensor operator. Hence, the variation tensor diffusion model is appropriate to compute texture information for superpixels extraction in this paper.

In general, growing methods often result in over-segmentation regions and they are often followed by merging procedure to improve the segmentation results [45]. Region merging process aims to yield a larger meaningful region with respect to the input image content by gathering a set of adjacent segments. Both similarity of feature space and the spatial relations of pixels or regions need to be computed in the merging process. The measurement of similarity usually employs color and texture information.

Many researches have developed the merging techniques in various aspects. Nock and Nielsen [27] improved the merging strategy by establishing a statistical feature of initial quantitative image and enhancing the merging model by a nonparametric mixture model estimation. Kuan et al. [44] presented a novel region merging strategy for segmenting salient regions in color images. The technique generated an initial set of regions by extracting dominant colors in the input image using a non-parametric density estimation methodology. Subsequently, a merging protocol based on importance index and merging likelihood criterions was used to refine the initial set. A good segmentation algorithm should preserve certain global properties according to the perceptual cues. So that Peng et al. [29] proposed a graph based merging order by defining region merging predicate with homogeneous, continuity and similarity of regions. In numerous texture feature detection methods, local binary pattern (LBP) is used to describe the spatial structure of local texture. The LBP approach was proposed by Ojala et al. [17] and provided highly discriminative texture information. It is invariant to monotonic changes in gray level and its computation is simple. For texture description, histograms of LBP patterns are used for segmentation [18].

### III. FORMING SUPERPIXELS BY GROWING PIXELS

Of all the image segmentation strategies, region growing method is a stable method for obtaining a homogeneous and continuous region. In order to find reliable segments, it is necessary to introduce both color and texture features when the growing process deals with complex images. To enhance the description ability of local regions, both structure tensor and color channels are combined as pixels features which are integrated into the region growing strategy for superpixels extraction. The basic idea underlying TV flow is to blur some image information and compute the minimum TV difference simultaneously so as to achieve denoising and edge preservation. So we adopt nonlinear diffusion to extract texture areas and enhance the edges simultaneously. The structure tensor calculation methods provided by Bigun et al. [20] can be extended to color images with three channels. The structure tensor of a color image $I$ with $c(=3)$ channels is defined as

$$
T_S = \begin{bmatrix}
T_{S,1}^{1,1} & T_{S,1}^{1,2} \\
T_{S,1}^{1,2} & T_{S,2}^{2,2}
\end{bmatrix} = g \ast \nabla I_c \nabla I_c^T
$$

$$
= \begin{bmatrix}
(G_{c})^2 & (G_{c}^x)(G_{c}^y) & (G_{c}^y)^2 \\
(G_{c}^x)(G_{c}^y) & (G_{c}^y)^2 & (G_{c}^x)^2
\end{bmatrix}
$$

where $\nabla$ denotes the gradient vector, $g$ is the Gauss kernel function to smooth the structure tensor, and $G_{c}^x(\cdot)$ denotes the partial derivative of a certain channel $c$ in a certain direction ($x$ or $y$). $T_s$ is easy to be interfered by noise and it is usually necessary to be filtered for the texture region detection relying.
\[ S = \sqrt{\text{moment matrix}} \] on nonlinear diffusion filtering. As a consequence, the second moment matrix \( T_s \) is replaced by its square root by eigenvalue decomposition [16]

\[
\tilde{T}_s = \sqrt{T_s} = \frac{T_s}{|\nabla|} = \begin{bmatrix} T_s^{1,1} & T_s^{1,2} \\ T_s^{2,1} & T_s^{2,2} \end{bmatrix}
\]

Thus, the texture feature in this section is denoted by \( U = (u_1^c, u_2^c, u_3^c, u_4^c, u_5^c) \), which consists of 15-D matrices in color image (\( c = 3 \)). \( c \) represents arbitrary channel of image in RGB color space. \( u_1^c \) denotes the original channel \( c \) of the image, \( \langle u_c^1, u_c^2, u_c^3 \rangle \) are the corresponding structure tensors of channels \( c : \langle \tilde{T}_c^{1,1}, \tilde{T}_c^{1,2}, \tilde{T}_c^{2,2} \rangle \), and \( u_c^k \) is scale estimation of channel \( c \). It is considered that the scale of the region is inversely proportional to the changes in pixel density and the detail scale measurement is described in [21]. In order to ensure consistency of all characteristics, the values in \( U \) are all mapped to the intervals \([0, 255]\)

\[
U_c^k = \frac{255 \times (U_c^k - \min(U_c^k))}{\max(U_c^k) - \min(U_c^k)}. \quad (3)
\]

An image size of \( m \times n \) is divided into average lattices as \( S = \sqrt{m \times n/K} \) and \( K \) is the initial superpixels number predefined. The seed pixels of superpixels are located at the minimum gradient points in the average lattices on the image and the specific seed pixels sampling method can be find in [22]. When computing superpixels, pure texture features show blurring of the true edges so that the boundary information of superpixels is weaken. So we integrate RGB channel for locating accurate boundaries in growing superpixels. The color-texture descriptor of a pixel is denoted by \( F(R, G, B, U') \). The superpixels growing process is implemented in the adjacent lattice (\( r = 1.5S \)) around the seed pixels for compact and uniform superpixels with complete textures. The normalized distance between pixels and regions in the combined feature space is calculated as follows:

\[
d_{i,j} = \frac{||\text{RGB}_i - \text{RGB}_j|| + ||U'_i - U'_j||}{2 \times 255}. \quad (4)
\]

where \( ||\cdot|| \) denotes the Euclidean distance in the combined feature space of color and texture. Let \( s_1, s_2 \ldots s_\ell \) denote the initial seeds of the local areas and \( \Omega_i \) denotes the growing area corresponding to \( s_i \). The feature RGB and \( U \) of growing regions \( \Omega_i \) are averages of pixels in \( \Omega_i \). The growing algorithm is thus outlined in Algorithm 1.

For keeping the real boundaries and homogeneous regions, the explicit iteration argument is in proportion to the pixels number in superpixels \( S/2 \).

We show our advantages in Figs. 2 and 3 by comparing with superpixels produced by SLIC [22], Turbo [24], and LSC [25]. In Fig. 2, we display the boundaries affinity comparison of our method against SLIC, Turbo, and LSC on two images with complex texture. One can note that our method can retain good localization of boundaries on test images. For the sake of fairness, we test SLIC, Turbo, and LSC with combined features \( F(R, G, B, U') \) and show the results in Fig. 2. In the cheetah image, there are a few improvements in the real boundary detection with combined features \( F(R, G, B, U') \). In the zebra

\begin{algorithm}
  1: Initialise the selected seed pixels as area \( \Omega_k \);
  2: Assign a label \( p_i \) for each area \( \Omega_i \);
  3: Record all labeled pixels in a list \( T \);
  4: Select the seed pixels \( x_i(x_j \in \Omega_k) \) from \( T \) and check its eight-neighbor pixels \( \text{Neigh}(8) \). If the \( d(i, j) \) between \( \Omega_i \) and \( \text{Neigh}(j) \) is smaller than the \( d(i, R) \) between \( \Omega_i \) and the average \( F \) of neighbor regions (\( r = 1.5S \)) around the seed of \( \Omega_i \), set the label of \( \text{Neigh}(j) \) as \( p_i \). Next, add neighbors to region \( \Omega_i \) and \( T \), update the \( F \) of this region \( \Omega_i \) and delete \( x_i \) from \( T \).
  5: Connect all small regions < 20 pixels to similar and adjacent regions.

\end{algorithm}
image, the boundary affinity with truth are improved in certain areas (in red rectangles), but also preform not very well in yellow rectangles. Fig. 3 shows the average performance of boundaries precision on BSD300 compared with SLIC, Turbo, and LSC and it is obvious that our method obtains the better boundaries affinity with manual boundaries and gets uniform superpixels with complete textures.

From the experimental results in Fig. 3, we can see that the average boundaries precision of segmentations with 900 superpixels is higher than 95% compared with the ground truth, which can meet the requirement of the segmentation task. Moreover, more superpixels will increase the computation complexity of merging and characteristics calculation, and it is not conducive to calculate the complete description of the texture in single superpixels. Therefore, we set the superpixels number \( K = 900 \) to ensure the boundary accuracy and texture integrity in the growing procedure.

IV. GRAPH-BASED MUMFORD–SHAH OPTIMAL REGION MERGING

In recent works, many region merging algorithms based on statistical properties [27], [28], graph properties [29], and interactive model [30] have been developed for image segmentation. The basic merging criteria which relies on color homogeneity always suffers from small and meaningless regions. Some other merging algorithms use global optimization energy terms, region number and the region area to control the merging process rather than setting termination condition.

To find an optimal sequence of merging which produces a union of optimal labeling for all regions, the minimization of a certain objective function \( F \) is required. At the same time, M–S function is a variational model proposed for segmentation, and it detects optimal result by using objective function. M–S segmentation uses the energy functional to judge all pixels whether can be merged into homogeneous regions and obtain qualitative region boundaries.

In addition to the merging strategy, feature selection is the key issue for reliable segmentation. A combined feature including color-texture information is proposed to describe the region properties, which is a reliable similarity measurement for region discrimination and the standard deviation is introduced for further control of the merging process.

Therefore, we design a hybrid constraints merging method based on the M–S model that can optimize multiple features of image for reasonable merging. The M–S model is described in Section IV-A, the feature definition and distance metric are described in Section IV-B, and the detailed description of the merging algorithm is in Section IV-C.

A. Mumford–Shah Model

In the M–S model [32] in (5), \( I_0 \) represents the image to be segmented and \( I \) represents a differentiable segmented image. In [32], \( R \) is the image located in planar domain, \( R_1 \) is the disjoint connected area and \( \Gamma \) are the union of the portions of the boundaries of \( R_1 \) inside \( R \). The positive constant coefficient \( \mu \) and \( \nu \) are the scale controller of M–S model. Then the M–S functional is defined as follows:

\[
E(I, \Gamma) = \mu^2 \int \int_R (I - I_0)^2 \, dx \, dy + \int \int_{R - \Gamma} (\nabla I) \, dx \, dy + \nu |\Gamma|.
\]

(5)

M–S theory will minimize energy \( E(I, \Gamma) \) to achieve a set of \( I \) as the best segmentation result. The three terms on the right side of (15) means: the first term is used to maintain the approximation between \( I \) and \( I_0 \) as far as possible; the second term is used to control the smoothness of \( I \) in each segmented region on the energy; the third term \( \nu |\Gamma| \) is used to control the boundary length to be parsimonious. Through the controls of three subitems, the M–S energy equation tries to minimize the boundary length and obtain continuous and smooth segments. By restricting \( I \) to be a local constant function in each region of the segmentation, the M–S model is simplified as

\[
E(I, \Gamma) = \sum_i \int \int_{R_i} (I - c_i)^2 \, dx \, dy + \nu |\Gamma|
\]

(6)

where \( \Gamma \) represents the length of all the objects boundaries, \( I \) represents the image to be segmented, \( c_i \) represents the average intensity of the \( i \)th region \( R_i \) and \( \nu \) is a constant parameter.

The key issue of M–S segmentation model is to design an efficient minimization energy function to combine the 2-D feature and 1-D (boundary) constraints together and act related effects. We study two typical M–S segmentation models to design our merging strategy. One model is trying to detect the largest energy decrease by using the heuristic greedy manner and it will merge two regions when the largest decrease is appeared. The energy model is simplified to keep an inequality relations between boundary and regional. The other model assumes the two neighboring regions contain pixels number \( N_i \) and \( N_j \) with average intensity as \( c_i \) and \( c_j \), the length of the common border between the region is \( \Gamma \). This model defines the energy function for the stage of before and after merging with region features and common boundaries. If the difference of the energy after merging becomes smaller, two regions should be merged.

B. Hybrid Region Histogram Feature and Distance Metric

In this section, we design a statistical histogram features as assistant descriptor of superpixels with color and texture information for two reasons: 1) histogram is a rotation invariant feature of superpixels and 2) complex regions cannot be properly characterized by simplex color information. The histograms feature we defined is able to describe the local color
distribution feature and enhance the categorized ability of superpixels.

For the number of pixels in superpixels being different and the number of color that can be discerned visually being limited, a quantized image is sufficient to describe the color distribution of the local area for segmentation. Quantization on the basis of uniform cutting of color space is too scattered, thus it will generate error and unnecessary texture. Therefore, an image is quantized to 64 colors by the minimum variance threshold to constrain the merging speed, which may lead to over-segmentation. Thus, the standard deviation of all image superpixels \( K \) superpixels) features is computed for setting termination condition and it is defined as

\[
V^2 = \frac{1}{k} \sum_{i=1}^{k} (\text{Hist}(i) - \text{Hist(image)})^2. \tag{9}
\]

From the test, it is found that the standard deviation changes too fast to control the merging speed, which may lead to over-segmentation. Therefore, we introduce an exponential function to modify the standard deviation \( V \) as follows, \( N \) is the current iteration times:

\[
\text{th} = \frac{V}{\exp(N)}. \tag{10}
\]

1) Standard Deviation-Based Threshold: The hybrid histograms is a combined feature of pixels group that has the advantage of discriminating different regions. To define the merging threshold, the standard deviation is a validated measure of dispersion of data set in statistics, and it is an important index to describe the differences between samples [47]. Thus, the standard deviation of all image superpixels \( K \) superpixels) features is computed for setting termination condition and it is defined as

\[
V^2 = \frac{1}{k} \sum_{i=1}^{k} (\text{Hist}(i) - \text{Hist(image)})^2. \tag{9}
\]

2) M–S Minimization: Inspired by the region competition merging model in [46], we develop the M–S model by introducing the novel histogram feature into the definition of M–S equation. The hybrid feature is integrated as a coefficient of region feature in the energy computing equation. Thus the boundaries \( \Gamma \) in (6) of two given adjacent segment pairs \( O_i \) and \( O_j \) can be replaced by \( \partial(O_i, O_j) \). The energy before merging \( \partial(O_i, O_j) \) and the energy after merging \( \partial(O_i, O_j) \) are defined as

\[
E_{\text{pre}} = \xi_i \int \int_{R_i} (I - c_i)^2 \, dx \, dy + \xi_j \int \int_{R_j} (I - c_j)^2 \, dx \, dy + \lambda \cdot \ell(\partial(R_i, R_j)) \tag{11}
\]

\[
E_{\text{after}} = \xi_i \int \int_{R_i} (I - c_{i+j})^2 \, dx \, dy + \xi_j \int \int_{R_j} (I - c_{i+j})^2 \, dx \, dy \tag{12}
\]

where \( \xi_i \) is the ratio of relative hybrid histogram feature of region \( i \) to standard deviation of the all histograms features of superpixels calculated by \( \xi_i = |\text{his}_i - \text{his(image)}|/V \). Where the standard deviation \( V \) is defined in (9), \( c_i \) denotes color mean value of region \( i \) in color image. \( I \) is the original image in \( R \) and \( \mu \) is the mean value of \( I \) in \( R \). \( \lambda \) is the weight of common boundaries in M–S model. \( \ell \) is the 1-D Hausdorff distance measure of boundaries length [32], which means the maximum distance between the common boundary points of the RGB space. \( \partial(R_i, R_j) \) denotes common boundaries of the two regions. Therefore the energy difference is

\[
E_{\text{after}} - E_{\text{pre}} = 2\xi_i \left( |R_i|(c_i - c) + |R_i| \left( c_i^2 - c^2 \right) \right) + 2\xi_j \left( |R_j|(c_j - c) + 2|R_j| \left( c_j^2 - c^2 \right) \right) - \lambda \cdot \ell(\partial(R_i, R_j)) \tag{13}
\]

where \( |R| \) is the surface measurement.
3) M–S Merging Strategy: The region adjacency graph (RAG) is a convenient model to represent adjacent relations between superpixels, so we employ RAG of all superpixels to illustrate the optimization merging process. All superpixels are mapped to an undirected graph \( G = (V, E) \). \( V \) is the set of nodes corresponding to superpixels, and \( E \) is the set of edges connecting the pairs of adjacent nodes whose corresponding superpixels have common boundaries. Corresponding weights of edges connected adjacent nodes are computed from the distance between hybrid histogram feature space of corresponding superpixels.

In the merging algorithm, the adjacency graph is established according to the boundary coincidence of all superpixels. Arbitrary pairs of adjacent nodes (superpixels) \((v_i, v_j)\) is connected by an edge \( E \) \((v_i, v_j)\) with the similarity weight calculated by \( \text{dist}(v_i, v_j) \). We establish adjacency matrix \( \text{Sim}(G) \) and a dynamic \( \text{Queue} \). The similarity matrix \( \text{Sim}(G) \) is constructed based on the adjacency relationships.

The merging manipulation is executed as follows: insert all of the nodes into a queue \( Q \) according to the similarity of link edges in ascending order. For each iteration, check each superpixel \( S \) in the \( \text{Queue} \) and all its adjacent regions \( N(i) = \{0, 1, 2...k\} \). If the two region meet merging conditions \( E_{after}(S, N(i)) > E_{pre}(S, N(i)) \) and \( \text{dist}(R_S, R_{N(i)}) < \text{th} \), merge the corresponding regions into one region, delete the edges \( e(v_S, v_{N(i)}) \) from the graph, and then remove the two nodes from the \( Q \). Insert the new region \( S' \) to the tail and update the adjacency graph of the new node \( S' \) and similarity matrix \( \text{Sim}(G) \). The loop is executed until all regions do not meet the merging criteria.

When merge two regions we should conduct three postprocessing: 1) change the bigger region label to the smaller one, and sum up the pixels number of two regions; 2) recompute the region histogram features of the merging region; and 3) update the adjacent relations of the adjacent table. The merging process is very efficient because no pixels processing is executed. In Fig. 5, we display the merging details of two images with intermediate results.

In this merging process, the adjacent properties, the region features and the geometric boundary are introduced into merging likelihood. By using the hybrid feature fusion M–S merging, it is easy to find a region merging sequence producing higher merging efficiency and more accurate segmentation results. The specific merging algorithm is described in Algorithm 2. In Fig. 6, we show some of segmentation results from the test images including the superpixels, merged results and region-iteration curves.

V. EXPERIMENTAL RESULTS

A. Evaluation of BSD Images

In this section, we test the segmentation effect of the proposed method on BSD300. The BSD300 is the most popular dataset for segmentation performance evaluation. It includes 300 images of the same size 321 × 481 pixels including various types of texture from the real world.

We first evaluate the proposed method qualitatively by using several representative images and comparing them with several well-known and novel algorithms: mean-shift [33], J measure based segmentation (JSEG) [34], CTM [26], and B-graph [35], GL-graph [36], and hybrid ordering and quartile analysis (HOQ) [37]. The comparison is based on three quantitative

Algorithm 2 M–S Merging Algorithm for Superpixels

**Input:**
- The initial RAG of the labeled superpixels \( G(L) \)

**Output:**
- The merged result of RAG and relabeled superpixels

1: Construct region adjacency matrix \( \text{Sim}(G) \) and dynamic \( \text{Queue} \) for \( G(L) \)
2: For each node \( S \) in \( G(L) \), compute similarity between adjacent regions to initialize adjacent matrix \( \text{Sim}(G) \) and insert all nodes into \( Q \) according to the order of the \( \text{dist} \) in descending order
3: For each node \( S \) in \( Q \), \( N(i) \) is neighbor region set of \( S \)
4: If \( E_{after}(S, N(i)) > E_{pre}(S, N(i)) \) & \( \text{dist}(R_S, R_{N(i)}) < \text{th} \) then
5: merged \( N(i) \) to \( S \); \( L = L - 1 \)
6: change the region label of \( N(i) \) to \( S \)
7: modified the combined histogram feature of new region
8: update \( Q, \text{RAG}, \text{Sim}(G) \)
9: end if
10: Return to 2 until no regions satisfy \( E_{after}(S, N(i)) - E_{pre}(S, N(i)) > 0 \), otherwise end the merge process
11: renumber the labels of merged segments
12: Return relabeled segments

![Fig. 5. Details merging process with intermediate results of two test images.](image-url)
performance measures: 1) probabilistic rand index (PRI) [38]; 2) variation of information (VoI) [39]; and 3) global consistency error (GCE) [40]. Many researchers have evaluated these indexes to compare the performance of various methods in image segmentation. Consequently, we believe these measures are able to demonstrate the segmentation effectiveness of the compared algorithms.

We select six images from BSD300 to display the visual segmentation effect. First, we show the segmentation results and compare them with five other algorithms in Fig. 7. The images in the first row are the original images, and those in the second row are the segmentation results of mean-shift with the bandwidth parameters $h_s = 8$, $h_r = 12$, and $\text{min}$-area = 500 pixels. The setting of the parameters is aimed to achieve more complete semantic regions of test images. The segmentation results of J-seg are shown in the third row. The JSEG algorithm needs three predefined parameters: 1) quantized colors = 10; 2) number of scales = 5; and 3) merging parameter = 0.78, which are provided in [34]. The images in the fourth row are the segmentation results of CTM ($\lambda = 0.1$), and those images in the fifth and sixth row are the segmentation results of B-graph and GL-graph method, in which the parameters are provided by the author in the open codes. The merger threshold $TH_l$ of 0.65 is default in HOQ, and its segmentation results are displayed in the seventh row. The images in the last row are our segmentation results.

The experimental data of the six compared images are listed in Table I. The table shows that our method outperforms the other algorithms based on the PRI, VoI, and GCE measures. From the segments in Fig. 7, we can observe that the mean-shift algorithm performs well in smooth color regions, while its segmentation performance excessively rely on the setting of the color and spatial scale parameters: $h_s$ and $h_r$. Mean-shift segmentation tends to obtain more redundant regions as over-segments. The segmentation results of JSEG performs well in identifying texture areas with specific and proper scale parameters. However, the experimental data indicates...

---

**Fig. 6.** Some segmentation results with superpixels, merged results, and iteration curves.

**Fig. 7.** Some segmentation results and comparison. Each row from top to bottom presents original images and segmentation results produced by mean-shift, JSEG, CTM, B-graph, GL-graph, HOQ, and our method.

**TABLE I**

<table>
<thead>
<tr>
<th>PRI, VoI, and GCE Value of Six Test Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Ours</td>
</tr>
<tr>
<td>HOQ</td>
</tr>
<tr>
<td>JSEG</td>
</tr>
<tr>
<td>CTM</td>
</tr>
<tr>
<td>B-Graph</td>
</tr>
<tr>
<td>GL-Graph</td>
</tr>
<tr>
<td>MS</td>
</tr>
</tbody>
</table>

---
that JSEG easily generates over-or under-segmented regions with improper parameters. And it has difficulty in segmenting the low color contrast, small, and narrow regions. CTM often displays over-or under-segmented regions enveloped by fault boundaries and it is rather time-consuming. The B-graph method is a type of graph partitioning method based on multilayer superpixels, which facilitates pixel grouping depending on various superpixels results. GL-graph is able to encode adaptively both local and global homogeneity of an object than B-graph. It is clear that B-graph and GL-graph outperform other methods in homogeneous region detection, but they can not work well in complex areas. The results of HOQ are more consistent with subjective human perception than above methods. From Fig. 7, one can note the proposed growing-merging framework which can segment the image into compact semantic regions and decrease the over-segments. In addition, by using a histogram-based color-texture distance metric and M–S optimization, the proposed method can obtain results with more complete regions and more accurate boundaries. We show some of the segmentations from BSD300 in Fig. 8.

The average values of PRI, VoI, and GCE of comparison methods on the BSD300 are given in Table II. It can be seen that our method shows outstanding performance in all evaluation indexes. The PRI and GCE ranks second and third in VoI. The PRI is very close to the best performance.

In addition to regional measures, the boundary accuracy is also an important evaluation index of the segmentation performance. Therefore, we employ the boundary accuracy computing strategy to evaluate the proposed method [43]. It defines precision measures the proportion of boundary pixels in the automatic segmentation $S_{\text{source}}$ that correspond to a boundary pixel in the ground truth $S_{\text{target}}$. Recall measures the proportion of boundary pixels in the ground-truth $S_{\text{target}}$ that were detected in the automatic segmentation $S_{\text{source}}$. Where $S_{\text{source}}$ is the boundaries extracted from segmentation results by computer algorithms and $S_{\text{target}}$ is the boundaries of human segmentation provided by BSD300. A boundary pixel is identified as true position when the smallest distance between the extracted and the ground-truth is less than a threshold ($\epsilon = 3$ pixels in this paper). The precision, recall are defined as follows:

$$\text{Precision}(S_{\text{source}}, S_{\text{target}}) = \frac{\text{Matched}(S_{\text{source}}, S_{\text{target}})}{S_{\text{target}}}$$

$$\text{Recall}(S_{\text{source}}, S_{\text{target}}) = \frac{\text{Matched}(S_{\text{source}}, S_{\text{target}})}{S_{\text{source}}}$$

We show the boundary precision-recall curve of six test methods in Fig. 9 on BSD300. It is clear that our algorithm performs better than compared methods in most of the test images. Compared with above-mentioned strategies, the boundaries obtained by our method are more approximate to the manual segmentations. Moreover, excess segmentation contours are relatively less than other methods and complete regions are detected precisely by contrast.

In this section, all the resolution of the test images in BSD300 are $321 \times 481$. In our algorithm, the average of time consumption to segment an image is 16.8 s including all the calculations and iterations. The average running time and code type of all the other algorithm are shown in Table III (on a laptop with Windows 7 system with 2.70 GHz CPU and 2.0 GB RAM).

### B. Evaluation on Remote Sensing Images

The remote sensing images contains complex information of land cover. In order to better develop the remote sensing image understanding and interpretation, the segmentation of
land cover is the key issue for exploiting the semantic information. Here, we choose several remote sensing images to demonstrate the segmentation performance of our method. All these test images are slices of Iowa in Brazil including different land covers acquired on July 5, 2015 by the advanced land imager on earth observing-1 in size of $512 \times 512$ pixels. We compared the segmentation results against the following algorithms: mean-shift, JSEG, CTM, and Bi-graph-cut, where the parameters are the same as Section V-A. The visual comparison is presented in Fig. 10. It is obvious that our method is able to obtain more accurate boundaries of homogeneous regions than compared algorithms.

In addition, we carried out an experiment to test the performance on large size of images of the proposed algorithm. A larger scope remote sensing image of Iowa in Brazil is chosen for the segmentation experiment in size of $2332 \times 2332$ pixels. Because of large size, the superpixels number is set to be 3600 for obtaining accurate boundaries. The segmentation result is shown in Fig. 11 and compared with mean-shift. Because of the overflow and error in the process of computing, the segmentation result can not be obtained by JSEG, CTM, and Bi-graph-cut. It is noted that the proposed method has a stronger adaptability in dealing with large images.

C. Evaluation on Medical Images

As we all know, medical images are always difficult to be segmented accurately by common methods. The level set and active contours (AC) model are used widely in medical image segmentation. In this section, we demonstrate the effectiveness and robustness of proposed strategy on some medical images. We compare our method with the improved level set method [41] and AC model [42] on the skin lesions and cells images. In this paper, the parameters of level set are set as follows ($\mu = 0.2$, $\Delta t = 1$, $\lambda = 2$, $\alpha = -0.3$), and the $\alpha = 0.1$ in AC. The segmentation results from superpixels and comparison is shown in Fig. 12. From the results we can see that our method is able to detect more accurate boundaries of lesions and cells than other two methods. Another advantage of the proposed method is that it is able to segment multiobjects and arbitrary regions no matter what subordinative relationships between them.

D. Algorithm Complexity

The complexity of our algorithm contains two parts: 1) the superpixels computing and 2) the superpixels merging. The total number of image pixels is $N$. In superpixels growing process, the texture feature, gradients and all pixels of the distance function are computed. Thus, each item update takes $O(N)$ operations. In superpixels merging process, each iteration requires the computation of the similarity table of the adjacency matrix and the histograms features, each update takes $O(N/K) \times O(\log N)$ operations ($K$ is the superpixels number), and it will take $O(K)$ iterations for the algorithm to converge. The overall complexity of our algorithm is roughly $O(N \log N)$.

VI. CONCLUSION

In this paper, we present a novel bottom-up segmentation method for color image segmentation based on superpixels. This method groups adjacent pixels as superpixels by growing seed pixels, which allows us to divide a complex image into small homogenous regions. This process employs the color-texture and scale features by exploiting structure tensors diffusion to generate superpixels enveloped with accurate boundaries. Another contribution of our method is the proposal of a composite color-texture histograms descriptor for regions and M–S model to supervise the merging process, which is
robust for discriminating regions with different texture structures. Compared with some other texture-based methods, our segmentation method is less affected by color mutation. The results of the experiment conducted on the test database attest to the excellent performance of our proposed image segmentation method under visual and quantitative evaluation.

REFERENCES


Haifeng Sima received the B.E. and M.E. degrees in computer science from Zhengzhou University, Zhengzhou, China, in 2004 and 2007, respectively, and the Ph.D. degree in software and theory from the Beijing Institute of Technology, Beijing, China in 2015.
Since 2007, he has been with the Faculty of Henan Polytechnic University, Jiaozuo, China, where he is currently a Lecturer with the School of Computer Science and Technology. His research interests include pattern recognition, image processing, image segmentation, and image classification.

Ping Guo (SM’05) received the B.S. and M.S. degrees from Peking University, Beijing, China, in 1980 and 1983, respectively, and the Ph.D. degree from the Chinese University of Hong Kong, Hong Kong, in 2002.
He is currently with the School of System Science, Beijing Normal University, Beijing, where he is the Founding Director of the Image Processing and Pattern Recognition Laboratory. He is also an Adjunct Professor with the School of Computer Science, Beijing Institute of Technology, Beijing.
His research interests include computational intelligence, image processing, pattern recognition, and astronomy big data analysis systems and applications.
Prof. Guo was a recipient of the Science and Technology (Third Rank) Award of 2012 Beijing Peoples Government for his contributions to Studies of Regularization Method and Their Applications.

Youfeng Zou received the B.E. degree from the Jiaozuo Institute of Technology, Jiaozuo, China, in 1984, and the M.S. and Ph.D. degrees from the China University of Mining and Technology, Beijing, China, in 1988 and 1994, respectively.
He is currently a Professor with Henan Polytechnic University, Jiaozuo. His research interests include mine survey and remote sensing classification.

Zhiheng Wang received the B.Sc. degree in mechatronic engineering from the Beijing Institute of Technology, Beijing, China, in 2004, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2009.
He is currently an Associate Professor with the School of Computer Science and Technique, Henan Polytechnic University, Jiaozuo, China. His research interests include computer vision, pattern recognition, and image processing.

Mingliang Xu received the B.E. and M.E. degrees from Zhengzhou University, Zhengzhou, China, in 2005 and 2008, respectively, and the Ph.D. degree from the State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou, China, in 2012, all in computer science.
He is currently an Associate Professor with the School of Information Engineering, Zhengzhou University, and also the Director of Center for Interdisciplinary Information Science Research. His research interests include computer graphics and computer vision.