

Texture superpixels merging by color-texture histograms for color image segmentation

Haifeng Sima^{1,2} and Ping Guo¹

¹ School of Computer Science and Technology, Beijing Institute of Technology
5 South Zhongguancun street, Haidian District, Beijing, P.R.China, 100081
[e-mail: smhf@hpu.edu.cn]

² School of Computer Science and Technology, Henan Polytechnic University
2001 Century Avenue, Jiaozuo, Henan, P.R.China, 454003
[e-mail: Pguo@bit.edu.cn]

*Corresponding author: Ping Guo

*Received March 28, 2014; revised April 27, 2014; accepted May 21, 2014
; published July 29, 2014*

Abstract

Pre-segmented pixels can reduce the difficulty of segmentation and promote the segmentation performance. This paper proposes a novel segmentation method based on merging texture superpixels by computing inner similarity. Firstly, we design a set of Gabor filters to compute the amplitude responses of original image and compute the texture map by a salience model. Secondly, we employ the simple clustering to extract superpixels by affinity of color, coordinates and texture map. Then, we design a normalized histograms descriptor for superpixels integrated color and texture information of inner pixels. To obtain the final segmentation result, all adjacent superpixels are merged by the homogeneity comparison of normalized color-texture features until the stop criteria is satisfied. The experiments are conducted on natural scene images and synthesis texture images demonstrate that the proposed segmentation algorithm can achieve ideal segmentation on complex texture regions.

Keywords: image segmentation; Gabor filters; color-texture histograms; region merging;

This work was supported by the grants from the [National Natural Science Foundation of China Grant. 61375045,61202207]], [Beijing Natural Science Foundation Grant.4142030] and National Science and Technology Support Program [Grant.2013BAH23F01]. We express our thanks to Dr. Mingliang Xu who checked our manuscript.

<http://dx.doi.org/10.3837/tiis.2014.07.011>

1. Introduction

Color image segmentation is an important step of image processing and computer vision. It has been widely applied to many image tasks, such as image annotation[1], image retrieval[2], scene analysis[3], pattern recognition[4,5] and multimedia tagging [6,7] etc. It is still a difficult problem to segment images accurately to satisfy all visual perception requirements and actual applications. The conventional image segmentation methods of color image usually bring large error segmentation and over segmentation when dealing with complex texture regions. Therefore, many researchers have introduced texture features to assist image segmentation by integrating different kinds of characteristics including color and texture for boundaries extracting [8,12,13,14,15,16,17]. Hence, how to detect textures is critical for ideal segmentation. Textures are common patterns in visual perception and they have no fixed morphological characteristics. Consequently, it is difficult to give an accurate and strict definition on texture. Despite all that, researchers had reached a consensus on the textures: homogeneous regions with regular gray-scale or color distribution performance of image can be regarded as textures.

Depending on the difference of theories and technologies, Tuceryan and Jain[4] classified texture extraction methods into four types: statistical type, model type, signal processing type, and structure type. From the view of human perception, a texture region displays irregular color or intensity in detail, but it displays repeatability and regularity in the overall visual performance. There are many homogeneous texture regions detection methods including co-occurrence matrix[9,10], wavelet and Gabor feature[11,12], Markov random fields[13], and multi-scale texture calculation models[14,15]. Each method has its own applicability and limitation of computing textures in the overall image. It is difficult to obtain accuracy boundaries and complete regions from complex image containing textures of various patterns and scales. More and more researchers are concerning about detecting texture features with multi-scale and multi-level technologies. Hence, we focus on exploring a method to improve the textures extraction by expanding regions from local area to global segmentation.

Since image segmentation is a complex and difficult problem, it is a good choice to divide the segmentation task into two parts. Many segmentation algorithms adopt merging strategy for superpixels(over-segmentation)[28,29,30]. In this work, both pixel-level features and region-level homogeneous cues (texture features of superpixels) are employed in color image segmentation. In order to guarantee the ideal segmentation of texture images, two criterions should be considered. The first criterion is to ensure that all superpixels are homogeneity, that means a single superpixel should belong to a single target region in the image. The second criterion is to obtain superpixels as large as possible for less superpixels can reduce time consuming of region merging. Complete and accurate homogeneity superpixels lead to better segmentation results by region merging.

In many computer vision tasks, various object regions are represented by corresponding histograms. Histogram features are widely used in computer vision applications for describing various characteristics, such as color distribution of the object, the edge gradient of an object, and the probability distribution of target position etc. Histograms are statistic data that assembled the statistical values to a series of pre-defined bins. The bin values can be calculated with various data types, such as gradient, direction, color, or any other features[16]. In this paper, we define a composite histograms features with the color and texture information of superpixels to characterize the regularity of texture and spatial distributions.

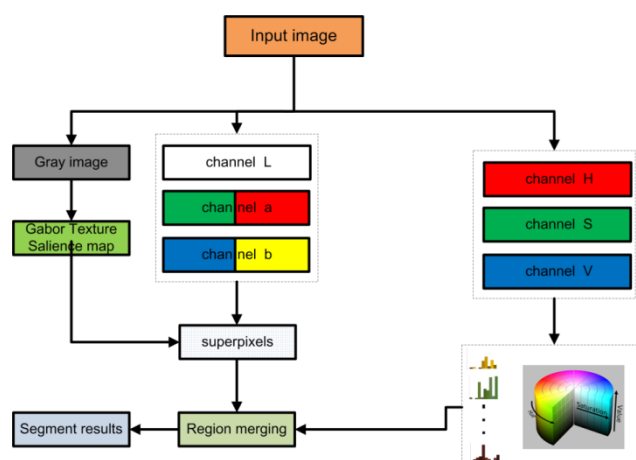


Fig. 1. Schema of the proposed approach

Inspired by previous works of texture segmentation and above discussion, we present a color image segmentation algorithm based on merging adjacent and similarity superpixels. First, we choose a bank of Gabor filters to extract texture responses of intensity image, and a novel saliency computing method is employed to compress the high dimension Gabor features into 1-D texture map. Then, we select clustering centers in the texture map at local areas and adopt pixels clustering to aggregate homogeneous pixels as superpixel based on color-texture similarity. Second, we define a novel composite histograms feature to measure the similarity among regions. Both adjacent and similar superpixels should be merged for final segmentation until the stop criterion is satisfied. **Fig. 1** Illustrates the pipeline of our method.

The main contribution of this paper includes: 1). A novel compressed method is defined to compute texture features with mixed scale and orientations Gabor amplitude. Integrated information of color and texture are introduced into simple clustering of local homogeneous pixels super pixels that contains complete texture and accurate boundaries. A better precision is obviously shown in section 3. 2). In the region merging section, a novel color texture descriptor is defined by color and distribution characteristics as composite histograms. The novel integrate feature has advantage in discriminating similar and dissimilar superpixels for reasonable merging. The experimental results in section 5 demonstrate that our method has better precision and universal performance.

The rest of the paper is organized as follows. In section 2, we briefly describe the related superpixels segmentation and merging algorithms, the section 3 is the superpixels extraction method of our paper, and the section 4 is merging strategies. Section 5 gives the partial results of the experiment and contrast with popular segmentation method, the final conclusion is given in section 6.

2. Related Work

Superpixels have been employed to aid color image segmentation in different segmentation algorithms[33,37,38,48]. The most popular superpixels extraction methods are graph-cut[17], Mean Shift (MS) [18] and watershed algorithm[19], and there are various of superpixels segmentation methods existed based on the above technologies[20-22]. The differences among these superpixels methods are the inherent segmentation theories or designing purpose.

Therefore, no single method is able to achieve the panorama control of superpixels on size, number, shapes and area compactedness[29].

Graph-cut makes use of the principle of max-flow/min-cut to find the optimal connection regions. It is based on the framework of global optimal function, and it has the capability of multi-feature fusion. However, the graph-cut performs poorly in texture boundary positioning [24,25]. In many applications, Mean Shift(MS) segmentation algorithm can bring better segmentation results with color and coordinates information. The essential process of MS is searching for local peaks based on density estimation in sliding windows. Watershed is a segmentation method based on mathematical morphology, and the attribute value of each pixel in the image is considered as altitude of the point. All local minimum values and their orbit areas are defined as the catchment basin, and the boundary of catchment basin is defined as the watershed. All catchment basin areas are obtained as segmentation results by developing different calculation methods[26,27]. The watershed methods always show frangible in texture areas[28].

Every type of method has its own advantages. The three segmentation methods above display different segmentation ability on local area, and perform well in uniform color areas. Because texture feature is of group pixels that require superpixels encompassed enough pixels to represent the textures patterns. To obtain the boundaries that surrounding complete texture is difficult for superpixels computing. In complex texture images, smaller scale superpixel is conducive to saving the real boundary. Alex Levinstein[29] proposed an effective superpixels extraction method by evolution contour model. They select uniform distribution seed points in the image according to the given number of superpixels. They expand regions to non-overlap superpixels based on the seeds in accordance with structural tension geometric contours model. The sizes of superpixels obtained by this method are more average, and relatively easy to converge to the real target boundary. In Radhakrishna Achantas' [31] method, they divide the image into rectangular blocks accordance with a pre-set number to meet given requirements. The cluster centers are chosen from every blocks, and each pixel is assigned to a similar superpixel by linear iterative clustering algorithm in five-dimensional feature space.

With pre-segment results, the next step is to aggregate homogeneous regions by merging adjacent regions that belong to the same object. Region merging techniques usually work with a statistical likelihood to decide the merging criterion of regions[32]. The measurement of similarity is critical for merging. In graph-based segmentation, superpixels are mapped to a weighted undirected graph and the region merging is transformed into a minimum cut problem[33]. Many region merging algorithms build objective optimization strategy based on combined constraints of pixel features, by which a global objective function is introduced to supervise the searching manipulation to accomplish merging process[34]. There are also many approaches integrate the boundary pixels information into the overall merging strategy[35,36]. Usually, a superpixel has more than one adjacent region and its boundaries can be divided into several parts according to different neighbor areas. Considering adjacent relations[35], the regional borders length is measured for similarity metrics, which is an effective assistant of color and spatial information in the merging process. In addition to above methods, there are many interaction method based on homogeneity seeds selection for region merging [37,38]. In unsupervised segmentation model, it is difficult to realize region completeness and accurate boundary. The advantage of our method is decomposition of whole segmentation task and a complete framework for pre-segment and merging with color-texture informations is proposed. Followed is the segmentation strategy in detail.

3. Superpixels extraction

Superpixels extraction is a procedure of subdividing an image into a number of homogeneous regions (a collection of pixels). As a key step of segmentation, superpixels extraction is always used for image classification, and annotation. For color image segmentation, the merging process is easier than segmenting pixels to right regions. Non-texture areas are relatively flat and it is easy to obtain the desired and precise boundaries by clustering pixels with color and spatial information. In texture areas, the boundaries are the combined portion of different adjacent regions and texture boundaries cannot be embodied by characteristics of single pixels. Hence, to obtain these boundaries requires united properties of neighbor area like frequency or statistical characteristics etc. Many algorithms have verified that multi-scale features are able to get the main outline for various texture regions of the image. For single scale, it lacks of adaptability to various textures due to the fixed size of sampling in the image. The coarse scale is conducive to detect big texture region, and the fine scale is more precise on perceiving the boundaries of different textures. The perception capability of fine scale is better than the coarse scale on little changes and it contributes toward boundaries keeping. Therefore, it is necessary to establish multi-scales texture descriptors for grouping homogeneous pixels. In segmentation tasks, it is difficult to fulfill both regional consistency and boundary keeping simultaneously. Therefore, the goal of segmentation is to reconcile the two requirements according to different requirements.

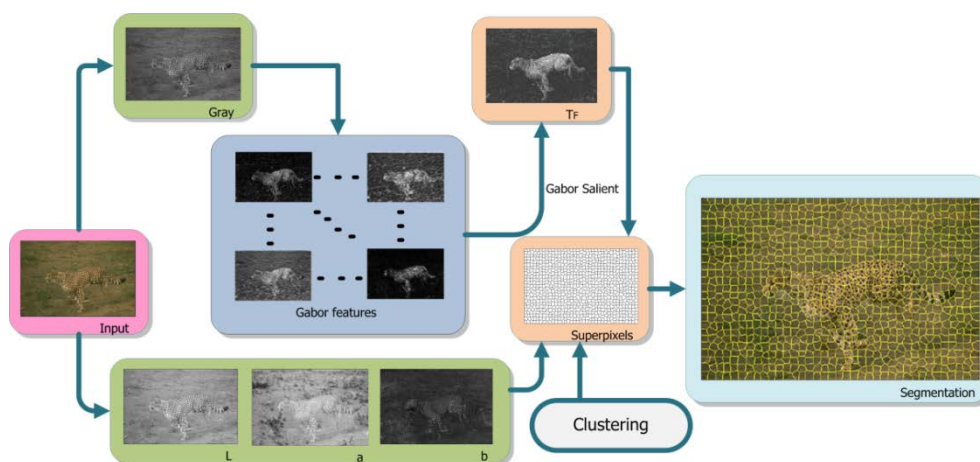


Fig. 2. The Flow chart of superpixels extraction in our approach

In this paper we propose a superpixels extraction strategy that takes full account of two factors: regional consistency and boundary keeping. It draws on the clustering strategy in SLIC[29]: clustering homogeneous pixels to produce superpixels based on six-dimensional features including color information in *Lab* color space, pixel locations (x,y) and texture map. *Lab* color space is chosen for clustering for it is better to perceive chromatic aberration.

3.1 Computing of texture features

Gabor filter response is an effective descriptors of texture features and widely used in texture classification and image segmentation[44]. To cope with complex texture regions, we use a bank of filters as multi-scale detectors work on the gray image for texture extraction. The filters is defined as G_n^k , k is the orientations, n denote the scales of the filters. Here a bank of

filters of four scales and six orientations ($n=1,3,5,7$; $k=0^\circ, 30^\circ, 60^\circ, 90^\circ, 120^\circ, 150^\circ$) are employed to detect local response of the intensity map $I(x,y)$. In this paper, the calculation of local response is as following:

$$F(x,y;n,k) = I(x,y) * g(x,y;n,k) \quad (1)$$

$$g(x,y) = \frac{1}{2\pi\sigma_u\sigma_v} \exp\left(-\frac{u^2}{\sigma_u^2} - \frac{v^2}{\sigma_v^2} + 2\pi i nu\right) \quad (2)$$

$$u = x \cos k + y \sin k, v = -x \sin k + y \cos k \quad (3)$$

$F(x,y;n,k)$ is a set of Gabor amplitude features consisting of $n*k$ dimension texture responses, where σ in equation (2) denotes the standard deviation. After convolution filtered, the textures area are enhanced and show visual salience than other area. The high-dimensions features are always compressed via vector quantization and clusters[51]. Here we introduce and spread a simple saliency computing method[52] to compute texture descriptors with Gabor amplitude features, where the F is smoothed by a nonlinear function $\text{Log}(1+F)$.

$$T_F = \sum_{n=1,k=1}^{n=4,k=6} (F(n,k) - \text{mean}(F(n,k))) \quad (4)$$

3.2 Simple Clustering for superpixels

In order to get homogeneous superpixels, K cluster centers are sampled in fixed intervals of the image plane. The size of the intervals is computed by $m = \sqrt{N * K / M}$ and $n = \sqrt{M * K / N}$, where M, N represents the size of image. To avoid the initialized centers position at edge and noise pixels, the location of the seeds are fixed to the lowest gradient points in the texture map T_F . The simple linear clustering is employed to group contiguous pixels for superpixels by computing pixels similarity. The clustering is carried out iteratively until the cluster centers are no longer changed. Color, coordinates and texture information bring distinct contributions to the clustering, and they cannot be unified into one distance metric framework. Thus, three types of differences are integrated and calculated as follows:

$$Dis = \alpha dis_{Lab} + \beta dis_{x,y} + \gamma dis_{texture} \quad (5)$$

$$dis_{Lab} = \frac{\sqrt{(L_i - L_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2}}{\max(\|L_i, a_i, b_i\|, \|L_j, a_j, b_j\|)} \quad (6)$$

$$dis_{x,y} = \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{m * n} \quad (7)$$

$$dis_{texture} = \frac{|T_F(i) - T_F(j)|}{\max(T_F) - \min(T_F)} \quad (8)$$

$$\alpha = \frac{R(xy) + R(T_F)}{R(Lab) + R(xy) + R(T_F)} \quad \beta = \frac{R(Lab) + R(T_F)}{R(Lab) + R(xy) + R(T_F)} \quad \gamma = \frac{R(x,y) + R(Lab)}{R(Lab) + R(xy) + R(T_F)} \quad (9)$$

The Dis is a combined distance between two pixels of three properties: color, texture and plane distance. Where α, β, γ respectively represent the contribution of the three properties in distance computing. In this paper, $R(Lab)$, $R(x,y)$, and $R(T)$ represent the domains of given properties. While the clustering segmentation is completed, there are still scattered pixels and small areas have not been classified or communicating. Classify unlabeled pixels and spurious

regions (smaller than 0.05 percentage of the image) to the most similar superpixels nearby. The process of superpixels extraction is show in Fig. 2. The detail comparison of superpixels results with *SLIC*[31] and *Tubor*[29] are shown in Fig. 3.

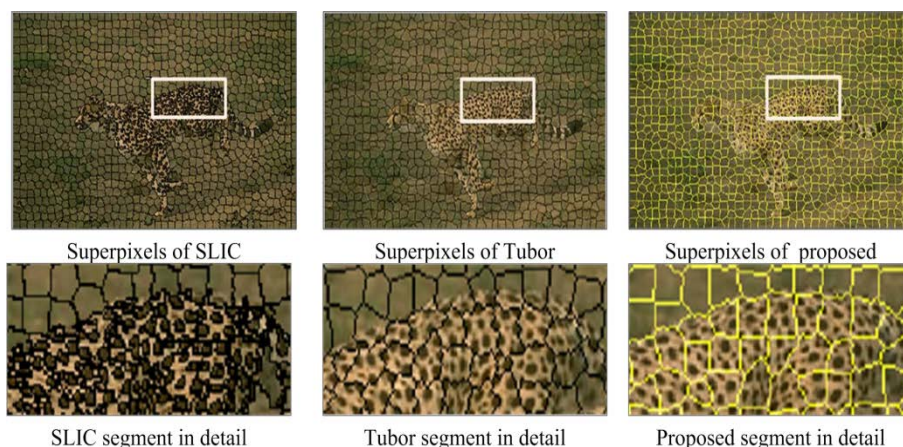


Fig. 3. Detail comparison of three methods

The segmentation performance is suffering from initial clusters number K in unsupervised strategy. Experimentally, the K value is set to be 900 to ensure that the boundary accuracy and texture integrity in the proposed algorithm. We employ the evaluation method of boundary accuracy rate in [23] to evaluate the pre-segmentation effect of our method. Some results and the boundary-recall curves against the K values are shown in Fig. 4.

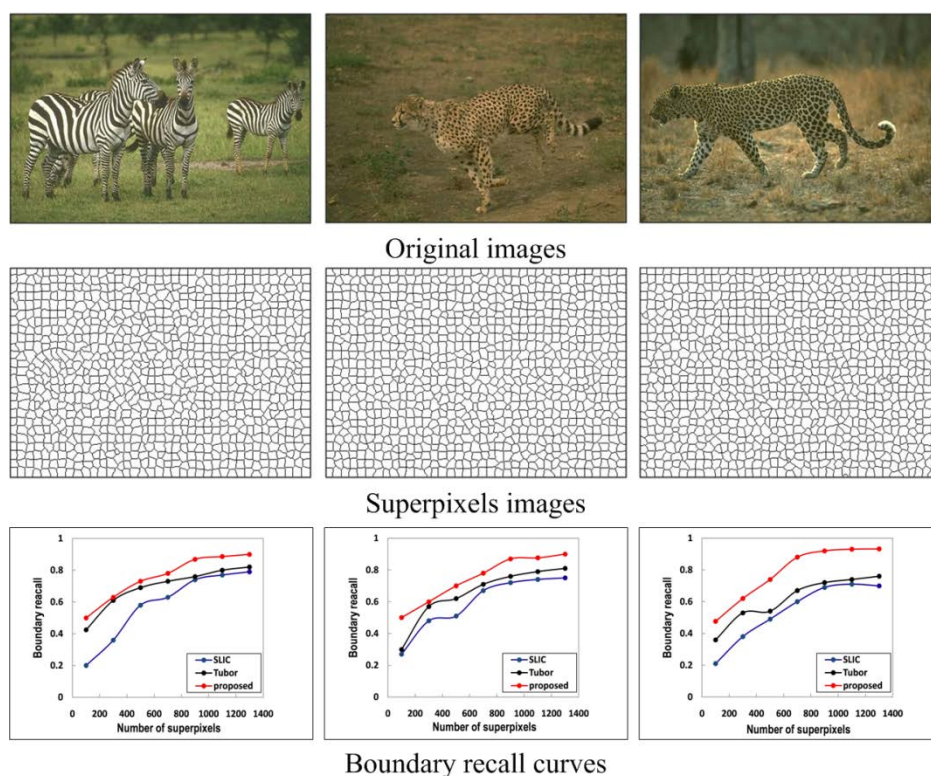


Fig. 4. Superpixels of proposed method and comparison of boundary recall with SLIC and Tubor

4. Merging rules

Many segmentation approaches are on the basis of merging over-segmented regions, and histogram-based representation of regions is very popular in many computer vision tasks. The measurement of similarity between regions are commonly rely on color differences, clustering distance, texture features, or geometry features, etc. Histogram feature of regions capture rich informations and possess powerful discriminability[3]. A color-texture histograms feature is employed as basis of similar computing because it has following advantages. Firstly, as a set of pixels, the superpixel contains more information, and it is easier for them to be classified than single pixel. Secondly, histograms contain discriminative information of color texture and exhibit the regularity of the texture to a certain extent[40]. Thirdly, as a statistical feature, histograms can provide classification criterion in statistical sense. The trend and dominant colors of the histograms are approximate between homogeneous texture areas, whereas histograms of different texture areas are obviously disparate. The histograms of homogeneous texture and inhomogeneous texture are shown in Fig.5:

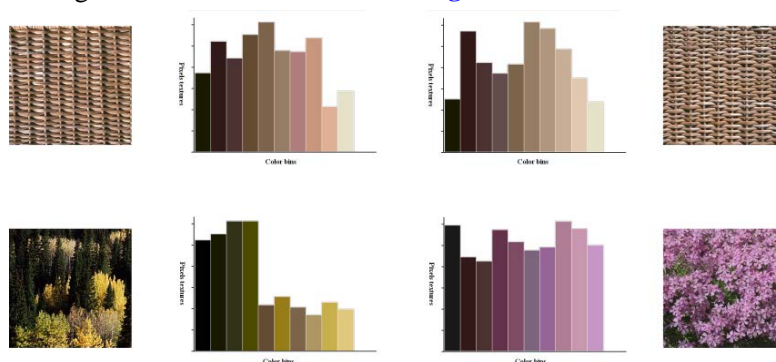


Fig. 5. Illustration of histograms of homogeneous textures (a and b) and inhomogeneous textures

4.1 Color-distribution histograms

The colors of natural image account for only fraction of color space, so we choose the more frequent colors to instead of similar colors by which frequent colors can provide sufficient region features and visual quality[41]. In this section, a color texture histograms is defined to represent color-texture information of regions by extracting major colors in HSV color space. Colors represented in HSV space are consistent with human visual system perception, and it is more conducive to analysis and quantification of colors than other color space.

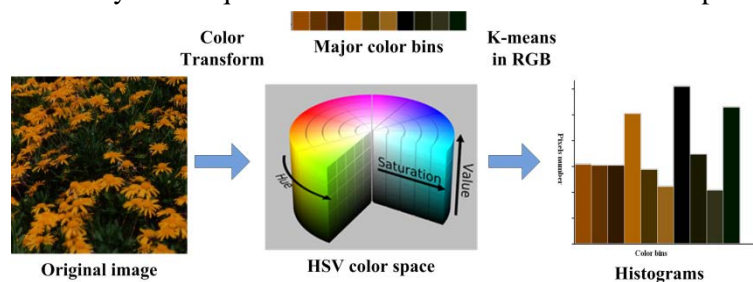


Fig. 6. Illustration of histograms producing

A fast building method for histograms is proposed based on k-means clustering. In the HSV space, all pixels of superpixels are mapped to ten clusters to build color bins of the histogram features. In order to find more frequent colors, the HSV space is divided into $12 * 5 * 5 = 300$ bins. For any superpixels mapped to HSV, we select ten bins who contain most pixels as initial clusters, and these bins must meet that they are not adjacent in the color space. With ten initial clusters, we classify all pixels to the nearest clusters by K-means clustering (iterate 5 times in this paper for convenience) in RGB space iteratively. It is better for measuring color difference in RGB space. The histograms producing process is shown in Fig. 6.

Superpixels obtained from section 3 are homogeneous area in complex image. The histograms of color characteristics alone cannot entirely distinguish the difference between textures regions, and the localization of the pixels within the superpixels is also crucial to measure dissimilarity between texture regions. Here we define a characteristic named color coherence vector (CCV) to represent the spatial distribution information of colors appeared in superpixels. Firstly, the length to width ratio (LWR) of all the cluster regions of one color in superpixels is defined to describe the color spatial-layout.

Definition 1. LWR: assuming a connected region r , compute the max coordinate value (x_{max}, y_{max}) and minimum coordinate value (x_{min}, y_{min}) in 2-D image, the region area of r is A , the LWR of r is computed by

$$LWR(r) = \frac{y_{max} - y_{min}}{x_{max} - x_{min}} \times A \quad (10)$$

Definition 2. CCV: Let $R(r_1, r_2, \dots, r_k)$ denoted the cluster regions sets of one color $C(r, g, b)$, the CCV of the color $C(r, g, b)$ is the sum of all LWR of region in R :

$$CCV(r, g, b) = \sum_{i=1}^k LWR(r_i) \quad (11)$$

The texture features are embedded into superpixels histograms as follows:

$$His(superpixels) = Union[CCV(r, g, b) * Bin(r, g, b)] \quad (12)$$

Where Union is combination of all color Bins. It is common to employ bin-to-bin distances for comparing histograms when the bins are aligned. Histogram intersection, correlation and CHI-square[41] are popular methods of calculating histogram similarity. The histogram intersection between two histograms P and Q is calculated as following:

$$D(P, Q) = \sqrt{(P - Q)^T A (P - Q)} \quad (13)$$

The numbers of pixels in superpixels are not constants, and different color texture provide inconsistent histogram bins with unpredictable colors and corresponding values. In order to avoid wrong judgment of homogeneous regions due to misplacement of similar colors, the two histograms must be calibrated by color distance. We construct a similar matrix for bins of histograms, and the bins are descending sorted by color distances from origin of color space. Based on the previous description, the similarity between two histograms can be calculated as formula (14), where the bin difference and color difference is in formula (15,16):

$$Similar(X, Y) = 1 - \sum_{i=1}^{10} \sqrt{(X_i - Y_i)^T C_i^{dif} (X_i - Y_i)} \quad (14)$$

$$(X_i - Y_i) = \frac{S_{V_i}^X - S_{V_i}^Y}{S_{V_i}^X + S_{V_i}^Y} \quad (15)$$

$$C_i^{dif} = \sqrt{\frac{(R_i^X - R_i^Y)^2 + (R_i^X - R_i^Y)^2 + (R_i^X - R_i^Y)^2}{\max((R_i^X + G_i^X + B_i^X)^2, (R_i^Y + G_i^Y + B_i^Y)^2)}} \quad (16)$$

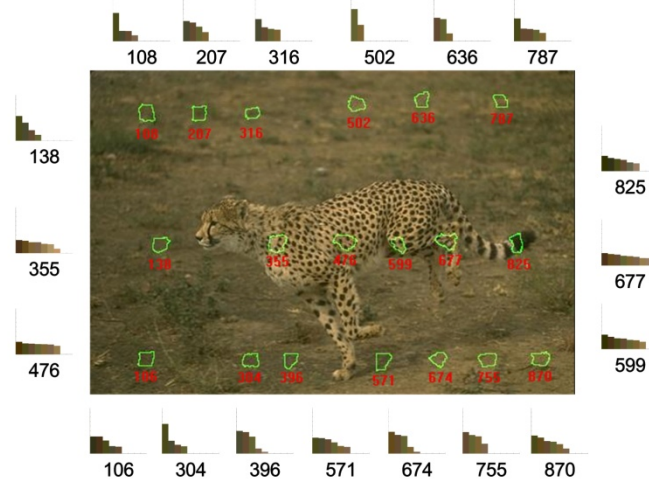


Fig. 7. Scheme of superpixels merging and iteration curves of merging

4.2 Merging algorithm:

Both spatial proximity and internal similarity of the superpixels are employed to supervise the merging process. Those homogeneous and adjacent regions can be merged, but nonadjacent regions cannot be merged even though they possess homogeneous features. Some histograms features of sample superpixels in the test image are shown in Fig. 7.

In the merging algorithm, how to determine the merging threshold of similarity is crucial for final segmentation. Larger threshold lead to under-segmentation and smaller threshold results in over-segmentation. It is well know that the standard deviation is a measure of dispersion of data set in statistics, and it is a reference index of the difference between samples[53]. The combined histograms is a mixture feature of pixels group that has advantage on recognizing different regions. At the same time, the difference between unmerged region and merged region is an important parameter to judge the similarity degree, denoted by $J_{histogram}$: if the difference is greater, the regions are not similar, otherwise, the regions are similar. Based on the above considerations, the standard deviation of the histograms feature set and $J_{histogram}$ are computed as following:

$$V = \sqrt{\frac{1}{K-1} \sum_{i=1}^K (His(i) - avg(His(K)))^2} \quad (17)$$

$$J_{histogram} = \frac{similar(His(N(i)), His(S))}{similar(His(S), His(S + N(i)))} \quad (18)$$

The merging threshold is defined as follows :

$$J_{threshold} = (1-V) \times J_{histogram} \quad (19)$$

The merging criteria relies on comparing the similarity of given region S and its adjacent region $N(i)$. While the similarity value is smaller than given threshold $J_{threshold}$, merge $N(i)$ into S , otherwise give up merging. The proposed threshold computing method brings global properties contrast for the merging, by which the threshold can avoid under-merging and over-merging effectively. A region adjacency graph is construct to represent superpixels to

facilitate the description of superpixels relations. The specific merging procedure is displayed in algorithm 1. The merging procedure is performed iteratively until the stop criteria is fulfilled.

Algorithm 1

Input: Labeled superpixels image; histograms set of all superpixels: $His(Superpixels)$

Initialize S_r, N_r . $-S_r$ denotes node sets of superpixels; $-N_r$ denotes adjacent table of nodes in S_r ;

Output: Merged results.

- 1: Construct region adjacency graph $G\{S_r, N_r(k)\}$ for K superpixels by spatial relationship ;
- 2: For each S in S_r , compute similarity between adjacent regions to build adjacent matrix $Sim(G)$,
- 3: For each node S in S_r , $N(i)$ is neighbor region of S
 - if $Sim(S, N(i)) > J_{threshold}$, then merged $N(i)$ to S ; $L = L - 1$,
 - 3.1 change the region label of $N(i)$ to S ,
 - 3.2 modified the histogram feature of two merged regions
 - 3.3 update $S_r, N_r, Sim(G)$ and $J_{threshold}$;
- 4: If no regions satisfied $Sim(S, N(i)) > J_{threshold}$, end the merge process, otherwise, return to 2;
- 5: Merge meaningless and small areas to adjacent regions

The histogram feature of merged region is updating by merging two sorted histograms to simplify calculating. We show the segment process and the iteration curves of test image in Fig. 8 and Fig. 9.

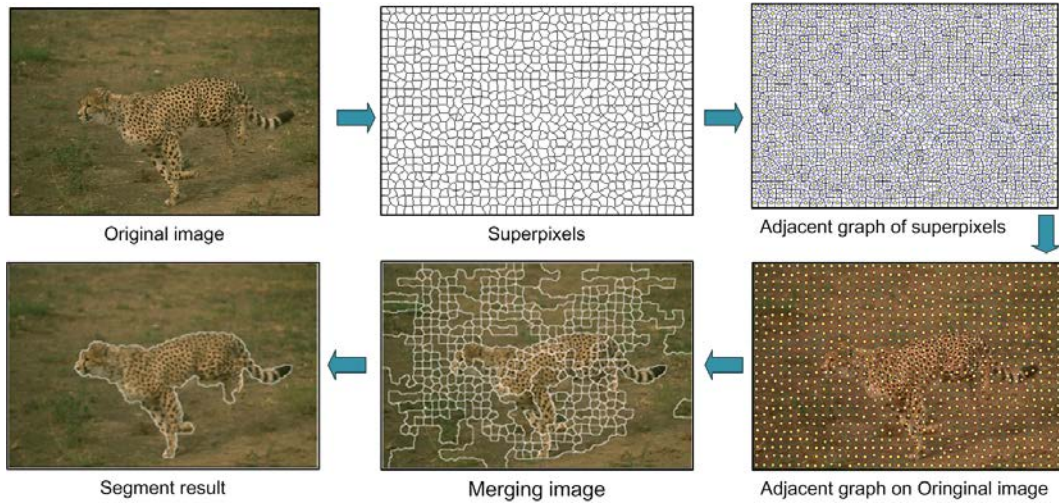


Fig. 8. Scheme of superpixels merging and iteration curves of merging

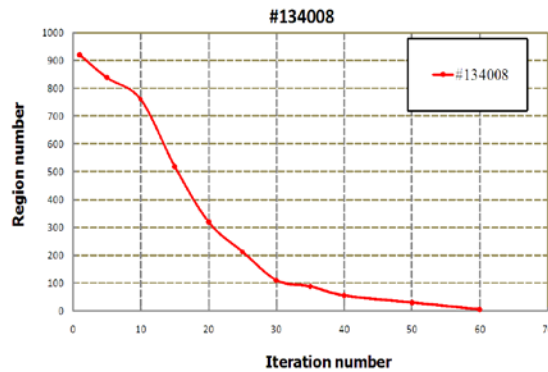


Fig. 9. Iteration curve of image 134008

5. Experimental results

To evaluate the efficiency of the segmentation algorithm in this paper, we employ two test databases: Berkeley segmentation dataset (BSD)[51] and MIT Vistex textures[46]. We deploy the test plat with Matlab2010 on Windows XP. The following sections describe the experimental setup, experiments results and comparison with other segmentation algorithms.

For a quantiative evaluation of segmentaions, Estrada, F.J.[41] proposed an excellent measure strategy to evaluate the segmentation accuracy. It defines precision and recall to be proportional to the total number of matched pixels between two segmentations S_{source} and S_{target} . Where S_{source} is the boundaries extracted from segmentation results by computer algorithms, and S_{target} is the boundaries of humane segmentation provided by BSD. A boundary pixel is identified as true position when the smallest distance between the extracted and groundtruth less than a threshold ($\epsilon=5$ pixels in this paper). The precision, recall and F-measures are defined in [45]. Precision is low when image is over segmented, and low recall value means under-segmentation. We employ this method to evaluate the performace of our proposed algorithm. There are also three common quantitative indexes are employed to evaluate segmentation performance: Probabilistic Rand Index (PRI)[48], Variation of Information (VOI) [49], Global Consistency Error (GCE)[50] to evaluate the segmentation performance. The value of PRI is in[0, 1), and high values ndicate a large similarity between the segmented images and the round-truth; the value of VoI and GCE are in[0, 1), and the smaller, the better.

5.1 Datasets and Experiment setting

The BSD is the most universal dataset for evaluating the performace of segmentation algorithms. It includes many kinds of natural images with complex textures of the real world. The BSD includes 300 images, and the images are divided into a trainning set including 200 images and 100 images for test with uniform resolution 321*481. It also provide the human segmentation results as ideal segmentation for evaluating the accuracy of testing algorithms. These images include different types of natural scene containing persons, buildings and animals, etc. MIT texture database was created with the intention of providing a large set of high quality textures for computer vision applications. All these we construct 6 texture mosaics from MIT Vistex textures dataset that including various textures in different scales and types from real world.

In our algorithm, the Gabor filter bank used in superpixels extraction is consist of four scales and six orientations filters: scales $n=1,3,5,9$; orientations $k=0^\circ, 30^\circ, 60^\circ, 90^\circ, 120^\circ, 150^\circ$. More superpixes lead to higher accuracy, the superpixels seeds number is set to be 900 to ensure obtaining complete regions and real boundaries for two test datesets. The minimum area is set to be 300 pixels for BSD and 1000 pixels for MIT dateset . The precision and recall evaluation method is implement for two test datasets, and the PRI, VOI and GCE index are employed to measure segmentation on MIT textures as further evaluation to demonstrate effectiveness of our method on complex textures.

5.2 Experiments on BSD

In this section, we conduct our methods on 100 images in BSD. The segmentation outlines are compared with two classic texture segmentation method JSEG[44] and UGC[45]. The JSEG algorithm has 3 parameters that need to be specified: quantized colors =12, number of scales=3, merging parameter=0.78. The balancing and texsons parameters in UGC is set to be $\lambda=0.2$ and $K=12$ as proposed in the original literature. Some segmentation results from the experimental images are shown in Fig. 10. The Fig. 10 shows six texture image segmentation

results of three algorithms. and better segmentation results are obtained by our algorithm compared with the state of art algorithms JSEG and UGC.

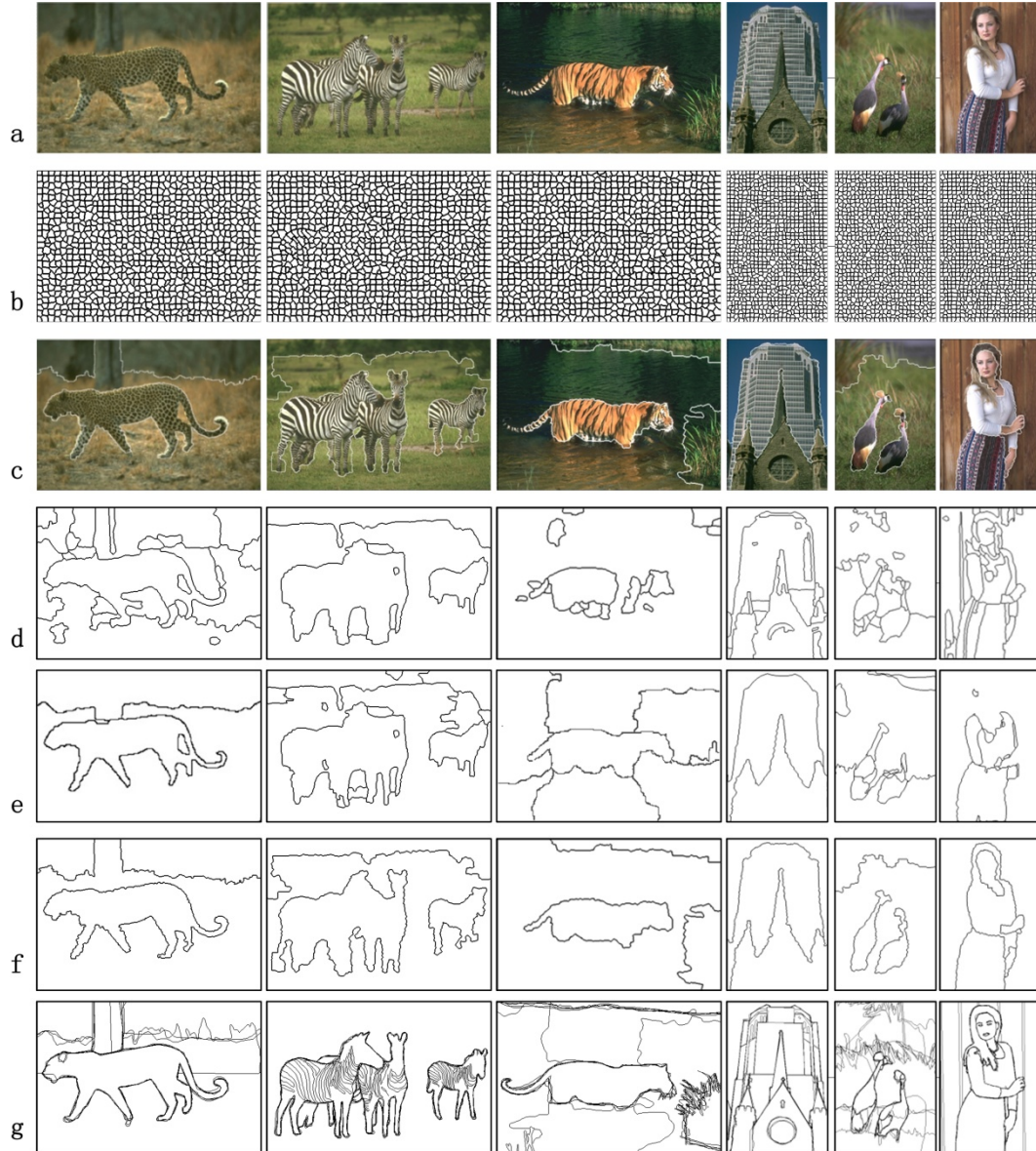


Fig. 10. Comparison of some segmentation results used JSEG, UGC and human boundaries in database BSD. (a) Original images, (b) Segmentation results of proposed method. (c) Superpixels by our method (d) Boundaries of JSEG. (e) Boundaries of UGC. (f) Boundaries of our method. (g) Multi-Human segmentation results.

Images in Fig.10-a are the original images, and in Fig.10-b are superpixels segmented by algorithm proposed in section 3.2. Images in Fig.10-c are the final segmentation results of our method. The images in Fig.10-d-f are contours contrast of JSEG, UGC, and our method. The images in Fig.10-g are the artificial segmentation contours provided by BSD database. The segmentation results demonstrate that our algorithm is an ideal method who can extract

complete and accurate target regions in texture images. It is able to segment images contains various color and spatial resolution in different texture regions. It is effective to mitigating wrong and mistaken segmentation compared with the other two methods. Fig.11 shows the iteration merging curve of the merging procedure of the six test images in Fig.10. The iteration merging of pre-segmentation results of sample images all converge reasonably well in linear complexity. Supervised by the $J_{threshold}$, the merging process result in accuracy segmentations. The visual results of the proposed algorithm show great advantages on texture perception, and the appearance of wrong and mistaken segmentation are reduced more than other two methods.

For an objectively evaluation of the segmentation performance, an excellent segmentation algorithm should provide low segmentation error with high boundary precision and recall. We compute all the precision and recall values of segmentation boundaries of selected images with humane segmentation results provide by BSD and draw the precision and recall curves. In Fig. 12, we show the boundary precision-recall curve of three test algorithms on BSD. It is very clear that our algorithm performs better than JSEG and the UGC method in most of the test images.

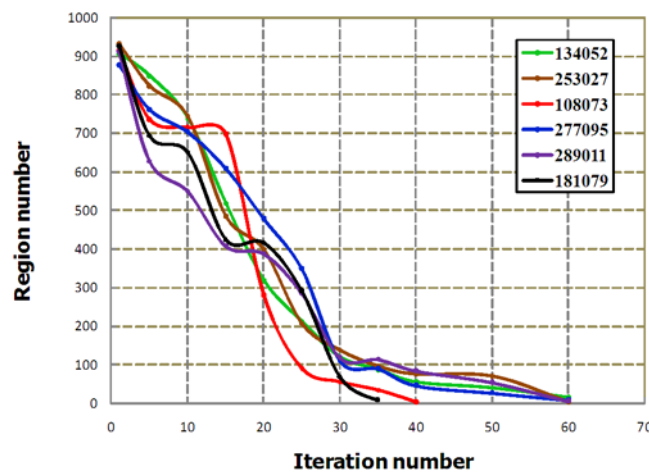


Fig. 11. Merging iteration curves of images in Fig. 10-a

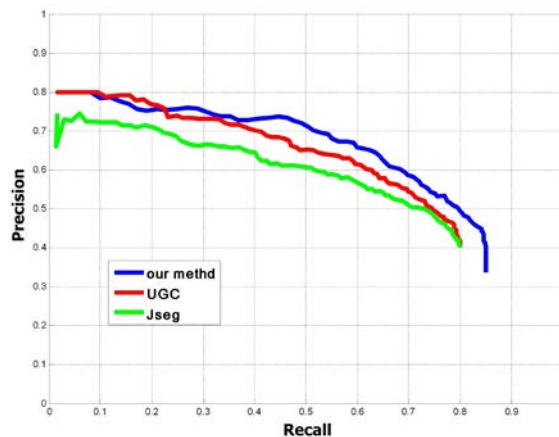


Fig. 12. ROC curve of test images in BSD at $\epsilon=5$

5.3 Experiments on MIT Textures

Furthermore, six synthesis texture mosaics are synthesized from MIT texture database[46] to verify the effectiveness on the mix-textures of our method. The resolution of test mosaics is 512×512 . In this part, we add two comparing methods Mean-Shift[18], and Bipartite-Graph-cut[47] except JSEG and UGC. In Fig. 13, we show the segmentation results of six mix-texture mosaics by above five algorithms. The images in Fig.13-a are the original images, and images in Fig.13-b are the results of Mean-Shift, and the parameters are set to be: $hs=8$, $hr=16$, $Min-area=2000$ pixels. The images in Fig.13-c are segmentation results of Bipartite-Graph-cut, and the initial regions of six mosaics are set to be $N_{tm1}=2$, $N_{tm2}=2$, $N_{tm3}=4$, $N_{tm4}=4$, $N_{tm5}=5$, $N_{tm6}=5$. Images in Fig.13-d-f are boundaries extracted by JSEG, UGC and our method. The parameters in JSEG and UGC obtained at JSEG(colors=12, scales=3, merging parameter=0.78) and UGC ($\lambda=0.2$ and $K=12$).

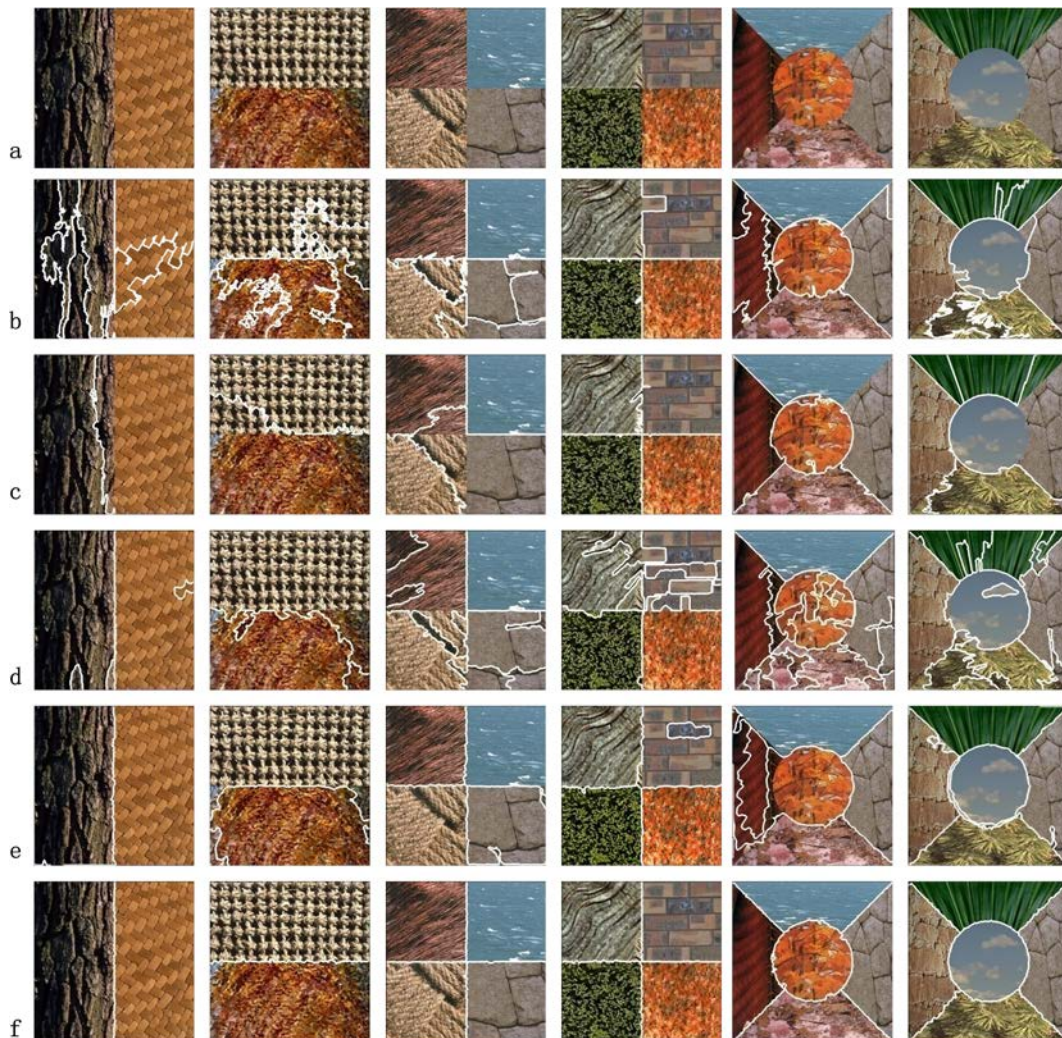


Fig. 13. Segmentation result on texture mosaic in MIT Vis database. (a) Original images. (b) Result of Mean shift. (c) Result of Bipartite-Graph-cut (d) Result of JSEG (e) Result of UGC (f) Results of proposed approach.

It is observed that the proposed algorithm has successfully demonstrated a more excellent segmentation results compared with other algorithms. Texture regions of different shapes and scales had been extracted completely and accurately as shown in Fig.13-f. The results of Mean-Shift segmentation show that it works well on local and flat regions and doesn't adaptive to textures images. In the detail part, Mean-Shift cannot recognize texture transition area and relies on the color difference greatly. So it works well in fine texture such as sea water, leaves, and show over-segmentation in coarse textures such as brick, big leaves etc. In Bipartite-Graph-cut, it is a merging algorithm based on hybrid over-segmentation results. This method combines diverse superpixels using a principled bipartite graph partitioning framework, and it needs determine how many pieces could be classified in advance. If the initial parameters are set improperly, the time-consuming is significantly higher than the other algorithms. It classifies pixels based on several different algorithms to build bipartite graph for merging. This method displays excellent performance on flat and non-texture images, and indicates bad performance of vary scales and types of texture regions. It is obviously to see misclassification of pixels in junction areas in Fig.13-c.

In JSEG, it is not necessary to construct textons database for pixels classification, but it needs to estimate texture patterns called J-map for images. This method requires pre-set kernel parameters, seeds and window parameters for computing J-map. The segmentation results are influenced and restricted greatly by initial parameters on global image. In the Fig.13-d, it do a good job in fine textures under the given similarity parameters: 0.88. In coarse textures, J-seg lead to wrong segment on big leafs, bricks and fabrics mosaics. UGC is a graph partitioning framework to achieve the minimum cut based on texsons computing. The texture extraction requires training primitive database texsons and querying to extract texture feature maps. From Fig.13-e, we can see that UCG can recognize complex texture segmentation results better than previous three algorithms: JSEG, MS and Bipartite-Graph-cut. In Fig.13-f, our approach shows accuracy boundaries results than former approaches. The proposed texture descriptor brings better perception on different types of textures with various scales.

In Table 1, we show precision and recall values of five compared algorithms on test texture mosaics($\epsilon=1$ pixels in this section). It indicates that our approach has performed excellent on boundaries extraction of synthetic texture images.

Table 1. Precision and recall values of five segmentation methods on texture mosaics TM1-TM6

Segmentation methods	Source images											
	Image TM1 toTM3						Image TM3 toTM6					
	precision			recall			precision			recall		
Mean-Shift	0.71	0.75	0.56	0.45	0.58	0.52	0.68	0.81	0.62	0.73	0.85	0.69
Graph-cut	0.03	0.36	0.51	0.05	0.31	0.64	0.26	0.83	0.69	0.30	0.92	0.51
JSEG	0.85	0.69	0.84	0.83	0.27	0.57	0.54	0.85	0.87	0.59	0.73	0.86
UCG texson	0.91	0.73	0.49	0.94	0.69	0.53	0.72	0.69	0.92	0.89	0.75	0.88
proposed	0.97	0.98	0.99	0.95	0.97	0.99	0.89	0.92	0.91	0.85	0.89	0.94

The time complexity of our method can be divided into two parts: superpixels extraction and iterative merging. The time complexity of computing superpixels is $O(N)$, where N is pixels number in image. The time complexity of iteration merging is $O(N+K^2/2+E^2/2)$, where K is the superpixels number, E is the adjacent relations between superpixels in the graph. The average time consuming is 16 sec on BSD, when carrying out in the workspace of Matlab2010.

The computation time depends on the size of the image regions, as well as the global complexity of an image.

The average values of Probabilistic Rand Index (PRI), Variation of Information (VoI), Global Consistency Error (GCE) and F-measure for the five methods on the synthesis images are given in **Table 2**. It is clearly that our method performs the best among the four evaluation criteria compared with these state of art methods.

Table 2. The average values of PRI, VoI, GCE and F-measure for synthetic images

	MS	BGC	JSEG	UCG	Ours
PRI	0.860	0.831	0.802	0.937	0.965
VOI	1.315	0.406	1.094	0.283	0.270
GCE	0.281	0.279	0.216	0.164	0.135
F-measure	0.664	0.509	0.692	0.824	0.972

6. Conclusion

By decomposing the segmentation into two stages, we present a new segmentation method for color textures image in this paper. First, A texture extraction technique is developed in an efficient way based on high dimension Gabor texture responses, and a multi-feature clustering is employed to extract superpixels as pre-segment regions. Second, we define a statistical texture descriptor(color-texture histograms) for superpixels with color and spatial distribution information. The segmentation results is obtained by merging adjacent and similar superpixels and the similarity is computed on the basis of color-texture histograms distance. The proposed segmentation strategy leads to precise segmentation and reduction of faulty assigned of pixels greatly. Experimental results show that our method provides more accurate segmentation compared with other segmentation methods. Future researches need to address the issue of choosing proper superpixels number and design high-level visual features to achieve optimal segmentation of given images automatically.

References

- [1] Wang M, Hua X S. "Active learning in multimedia annotation and retrieval: A survey." *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 2, 10, 2011. [Article \(CrossRef Link\)](#).
- [2] Hong, Richang, et al. "Image Annotation By Multiple-Instance Learning With Discriminative Feature Mapping and Selection," *IEEE Transactions on Cybernetics*, vol. 44, no. 5, pp. 669-680, 2014. [Article \(CrossRef Link\)](#).
- [3] Nianhua Xie, Haibin Ling, Weiming Hu, and Xiaoqin Zhang, " Use Bin-Ratio Information for Category and Scene Classification." in *Proc of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2313-2319, 2010. [Article \(CrossRef Link\)](#).
- [4] Tuceryan M, Jain A K, *Texture Analysis: Handbook Pattern Recognition and Computer Vision*, 2nd Edition, Singapore World Scientific, pp. 207-248, 1999. [Article \(CrossRef Link\)](#).
- [5] Zheng H, Ye Q, Jin Z. "A Novel Multiple Kernel Sparse Representation based Classification for Face Recognition," *KSII Transactions on Internet & Information Systems*, vol. 8, no. 4, pp. 1463-1480, 2014. <http://www.itiis.org/tiis/download.jsp?filename=TIIS%20Vol%208.%20No%204-17.pdf>
- [6] Wang, Meng, et al. "Assistive tagging: A survey of multimedia tagging with human-computer joint exploration," *ACM Computing Surveys*, vol. 44, no. 4, 25, 2012. [Article \(CrossRef Link\)](#).
- [7] Yang K, Hua X S, Wang M, et al. "Tag tagging: Towards more descriptive keywords of image

- content,” *IEEE Transactions on Multimedia*, vol. 13, no. 4 : 662-673, 2011. [Article \(CrossRef Link\)](#).
- [8] D.Martin, C.Fowlkes, and J. Malik, “Learning to detect natural image boundaries using local brightness, color, and texture cues,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no.5, pp. 530-549 , 2004. [Article \(CrossRef Link\)](#).
 - [9] Christoph Palm, “Color Texture Classification by Integrative Co-occurrence Matrices,” *Pattern Recognition*, vol. 37, no .5, pp. 965- 976, 2004. [Article \(CrossRef Link\)](#).
 - [10] Mousumi Gupta, Debasish Bhaskar et al. “Target detection of ISAR data by principal component transform on co-occurrence matrix,” *Pattern Recognition letter*, vol. 33, no. 13, pp. 1682-1688, 2012. [Article \(CrossRef Link\)](#).
 - [11] S E Grigorescu, N Petkov, P Kruizinga, “Comparison of Texture Features Based on Gabor Filters,” *IEEE Transaction on Image Process*, vol. 11, no. 10, pp. 1160-1167, 2002. [Article \(CrossRef Link\)](#).
 - [12] M K Bashar, T Matsumoto, N Ohnishi, “Wavelet Transform based Locally Orderless Images for Texture Segmentation,” *Pattern Recognition letter*, vol. 24, no. 15, pp. 2633-2650, 2003. [Article \(CrossRef Link\)](#).
 - [13] Zoltan Kato, Ting-Chuen Pong, “A Markov random field image segmentation model for color textured images,” *Image Vision Computing*, vol .24, no. 10, pp. 1103-1114, 2006. [Article \(CrossRef Link\)](#).
 - [14] Liang KH, Tjahjadi T, “Adaptive scale fixing for multiscale texture segmentation,” *IEEE Transaction on Image Process*, vol. 15, no. 1, pp. 249-256, 2006. [Article \(CrossRef Link\)](#).
 - [15] Osvaldo Severino Jr., Adilson Gonzaga, “A new approach for color image segmentation based on color mixture,” *Machine Vision and Application*, vol. 24, pp. 607-618, 2013. [Article \(CrossRef Link\)](#).
 - [16] Xiuwen Liu, DeLiang Wang, “Image and Texture Segmentation Using Local Spectral Histograms,” *IEEE Transaction on Image Process*, vol.15, no.10, pp. 3066-3077, 2006. [Article \(CrossRef Link\)](#).
 - [17] T. Cour, F. Benezit, and J. Shi, “ Spectral segmentation with multiscale graph decomposition,” in *Proc. of the IEEE International conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1124-1131, 2005. [Article \(CrossRef Link\)](#).
 - [18] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603-619, 2002. [Article \(CrossRef Link\)](#).
 - [19] Luc Vincent and Pierre Soille, “ Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583-598, 1991. [Article \(CrossRef Link\)](#).
 - [20] D. Hoiem, A. Efros, and M. Hebert, “Geometric context from a single image,” in *Proc. of the IEEE International conference on Computer Vision*, pp. 654-661,2005. [Article \(CrossRef Link\)](#).
 - [21] G. Mori, X. Ren, A. Efros, and J. Malik, “Recovering human body configurations: Combining segmentation and recognition,” in *Proc. of the IEEE International conference on Computer Vision and Pattern Recognition*, vol. 2, pp.326-333, 2004. [Article \(CrossRef Link\)](#).
 - [22] D. Hoiem, A. Efros, and M. Hebert, “Automatic photo pop-up,” *ACM Trans. Graph.* 24(3), 577-584, 2005. [Article \(CrossRef Link\)](#).
 - [23] X. Ren and J. Malik, “Learning a classification model for segmentation,” in *Proc. of the Asian Conference on Computer Vision*, pp. 10-17, 2003. [Article \(CrossRef Link\)](#).
 - [24] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, Rama Chellappa, “Entropy rate superpixel segmentation,” in *Proc. of the IEEE International conference on Computer Vision and Pattern Recognition* , pp. 2097-2104, 2011. [Article \(CrossRef Link\)](#).
 - [25] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph based image segmentation,” *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167-181, 2004. [Article \(CrossRef Link\)](#).
 - [26] D.Wang, “A multiscale gradient algorithm for image segmentation using watersheds,” *Pattern Recognition*, vol. 30, no. 12, pp. 2043-2052,1997. [Article \(CrossRef Link\)](#).
 - [27] Meyer, F, “An overview of morphological segmentation,” *IEEE Transactions on Pattern Analysis*

- and *Machine Intelligence*, vol. 15, no. 7, pp. 1089-1118, 2001. [Article \(CrossRef Link\)](#).
- [28] Fabio Drucker and John MacCormick, "Fast Superpixels for Video Analysis," *In: Proc of the International conference on Motion and Video computing*, pp. 55-62, 2009. [Article \(CrossRef Link\)](#).
- [29] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "TurboPixels: Fast Superpixels Using Geometric Flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2290-2297, 2009. [Article \(CrossRef Link\)](#).
- [30] Jeong Y S, Lim C G, Jeong B S, et al. "Topic Masks for Image Segmentation," *KSII Transactions on Internet & Information Systems*, vol. 7, no. 12, pp. 3274-3292, 2013. <http://www.itiis.org/tiis/download.jsp?filename=TIIS%20Vol%207,%20No%2012-18.pdf>
- [31] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-art Superpixel Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274 - 2282, 2012. [Article \(CrossRef Link\)](#).
- [32] S.-C. Zhu and A. Yuille, Region Competition, "Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884-900, 1996. [Article \(CrossRef Link\)](#).
- [33] Wenbing Tao, Hai Jin, Yimin Zhang, "Color Image Segmentation Based on Mean Shift and Graph Cuts," *IEEE Transaction on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37 no. 5, pp. 1382 -1389, 2007. [Article \(CrossRef Link\)](#).
- [34] H. D Cheng, Y. Sun, "A hierarchical approach to color image segmentation using homogeneity," *IEEE Transaction on Image Process* 9(12), 2071-2082, 2000. [Article \(CrossRef Link\)](#).
- [35] KuanYu-hsin, Kuo Chuag-ming, Yang Nai-chung, "Color-based image salient region segmentation using novel region merging strategy," *IEEE Transaction on Multimedia*, vol. 10, no. 5, pp. 832-845, 2008. [Article \(CrossRef Link\)](#).
- [36] A Y Yang, J Wright, Y Ma, "Unsupervised segmentation of natural images via lossy data compression," *Comput Vis Image Und.* 110(2),212-225, 2008. [Article \(CrossRef Link\)](#).
- [37] J. Ning, L. Zhang, David Zhang and C. Wu, "Interactive Image Segmentation by Maximal Similarity based Region Merging," *Pattern Recognition*, vol. 43, no. 2, pp. 445-456, 2010. [Article \(CrossRef Link\)](#).
- [38] Bo Peng, Lei Zhang and Jian Yang, "Iterated Graph Cuts for Image Segmentation," in *Proc. of the Asian Conference on Computer Vision*, pp. 677-686, 2009. [Article \(CrossRef Link\)](#).
- [39] Tamura H, Moil S, Yamawaki T, "Texture features corresponding to visual perception," *IEEE Transaction on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 8, no. 6, 460-473,1978. [Article \(CrossRef Link\)](#).
- [40] George Paschos, Maria Petrou, "Histogram ratio features for color texture classification," *Pattern Recognition letter*, vol. 24, pp. 309-314, 2003. [Article \(CrossRef Link\)](#).
- [41] M.-M. Cheng, G.-X. Zhang, N.J. Mitra, X. Huang, S.-M. Hu, "Global contrast based salient region detection," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 21-23. 2011. [Article \(CrossRef Link\)](#).
- [42] H. Ling and K. Okada, "An Efficient Earth Mover's Distance Algorithm for Robust Histogram Comparison," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 840-853,(2007) [Article \(CrossRef Link\)](#).
- [43] Y.Deng, B.S.Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 800-810, 2001. [Article \(CrossRef Link\)](#).
- [44] J. S. Kim and K. S. Hong, "Color-texture segmentation using unsupervised graph cuts," *Pattern Recognition*, vol.42, no. 5, pp. 735-750, 2009. [Article \(CrossRef Link\)](#).
- [45] Estrada, F.J., Jepson, A.D, "Quantitative Evaluation of a Novel Image Segmentation Algorithm," in *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1132-1139, 2005. [Article \(CrossRef Link\)](#).
- [46] MIT Vis Texture database. <http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>
- [47] Zhenguo Li,Xiao-Ming Wu, Shih-Fu Chang, "Segmentation using superpixels: A Bipartite graph

- partitioning approach,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 789-796, 2012. [Article \(CrossRef Link\)](#).
- [48] R. Unnikrishnan, C. Pantofaru, and M. Hebert, “Toward objective evaluation of image segmentation algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 929-944, 2007. [Article \(CrossRef Link\)](#).
- [49] R. Unnikrishnan, C. Pantofaru, and M. Hebert, “Toward objective evaluation of image segmentation algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 929-944, 2007. [Article \(CrossRef Link\)](#).
- [50] D. Martin, C. Fowlkes, D. Tal, J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proc. of IEEE International Conference on Computer Vision*, pp. 416-423, 2001. [Article \(CrossRef Link\)](#).
- [51] T. Leung, J. Malik, “Representing and recognizing the visual appearance of materials using three-dimensional textons,” *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29-44. 2001. [Article \(CrossRef Link\)](#).
- [52] Achanta R, Hemami S, Estrada F, et al. “Frequency-tuned salient region detection,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1597-1604, 2009. [Article \(CrossRef Link\)](#).
- [53] Ibrahim M T, Khan T M, Khan MA, “Automatic segmentation of pupil using local histogram and standard deviation,” in *Proc. of Visual Communications and Image Processing*, pp. 77442S-77442S-8, 2010. [Article \(CrossRef Link\)](#).



Sima Haifeng is currently a Ph.D candidate of Beijing Institute of Technology and a lecturer of School of Computer Science and Technology, Henan Polytechnic University. He received his M.S. degree in software from ZhengZhou University in 2007. His current research interests include image process, and pattern recognition.



GUO Ping (IEEE Senior Member) is currently a Professor at the Computer Science Department of the Beijing Normal University and the School of Computer Science & Technology of Beijing Institute of Technology. From 1993 to 1994 he was with the Department of Computer Science & Engineering at the Wright State University as a visiting faculty. From 2000.5 to 2000.8 he was with the National Laboratory of Pattern Recognition at Chinese Academy of Sciences as a research associate. From 2005.7 to 2005.12 he was with the Computer Science Department at The Chinese University of Hong Kong as a research associate. He received his M.S. degree in Optics from Peking University, his Ph.D degree in Computer Science from The Chinese University of Hong Kong. His current research interests include neural network, image process, pattern recognition, software reliability engineering, optical computing and spectral analysis.