IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 18, 2019, accepted October 30, 2019, date of publication November 4, 2019, date of current version November 14, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2951228*

# A Novel Color-Texture Descriptor Based on Local Histograms for Image Segmentation

**YANG LIU[1], GUANGDA LIU[1], CHANGYING LIU[1], AND CHANGMING SUN[2]**
[1]College of Instrumentation and Electrical Engineering, Jilin University, Changchun 130021, China
[2]CSIRO Data61, Epping, NSW 1710, Australia

Corresponding author: Changming Sun (changming.sun@csiro.au)

This work was supported in part by the Jilin Scientific and Technological Development Program under Grant 20150204053GX, and in part by the Jilin Scientific and Technological Development Program under Grant 20160101286JC.

**ABSTRACT** In this paper, we propose a novel color-texture image segmentation method based on local histograms. Starting with clustering-based color quantization, we extract a sufficient number of representative colors. For each pixel, through counting the number of pixels with each representative color within a circular neighborhood, a local histogram is obtained. After the circular neighborhood is extended to several scales, a local histogram with an appropriate scale is adopted as a color-texture descriptor at the corresponding pixel for image segmentation. Further, we correct the color-texture features near boundaries and obtain a initial segmentation by a clustering method with the color-texture descriptors. Finally, in order to obtain a better segmentation result, we merge the over segmented regions guided by the obtained boundaries. Experiments are performed on both synthetic and natural color-texture images, and the results show that our proposed method performs much better compared with state-of-the-art methods on image segmentation, particularly in textured areas.

**INDEX TERMS** Image segmentation, texture segmentation, local histogram, adaptive scale descriptor.

## I. INTRODUCTION

Color image segmentation deals with partitioning a given image into several visually distinct regions, so that each region has a homogeneous texture. It is an essential step towards content analysis and image understanding. Applications based on segmentation are numerous, and they can be from medical image analysis and face identification to remote sensing.

A variety of classic color image segmentation methods have emerged for different applications, including those using local binary patterns (LBP) [1], Gabor filtering and clustering [2], [3], graph cuts [4]–[6], active contours [7]–[9], watersheds [10], [11], level sets [12], [13], region growing [14], [15], mean shift [16], knowledge-based approaches [17], and machine-learning based methods [18]. The processes of these segmentation methods are basically the same, which can be summarized as the following three steps: (1) feature extraction; (2) segmentation; and (3) post-segmentation processing.

The associate editor coordinating the review of this manuscript and approving it for publication was Kumaradevan Punithakumar.

Color and texture information collectively have strong links with human perception. They are powerful visual cues and provide useful features for the segmentation of complex images, which exhibit significant inhomogeneities in color and texture. Texture is different from color in that it reveals the spatial organization and arrangement of a set of basic elements or sequences (i.e., textons), and it has three primary characteristics: continuous repetition of some textons, non-random arrangements, and roughly uniform in the texture regions.

Feature extraction is a vital step in color image segmentation. It is necessary to distinguish textures by measuring their special properties. The special property used to represent the texture is called texture feature. There are a wide variety of classic methods for extracting texture features, including gray level co-occurrence matrices (GLCM), LBP, Gabor filters, and wavelet methods. However, most of them are designed for gray-level image segmentation, and fail to combine the information contained in color channels (i.e., the red, green, and blue channels). In many practical scenarios, the color-alone or texture-alone image information is not sufficiently robust to accurately describe the image content. Therefore, image segmentation based on the integration of color and

texture information has attracted attention. For example, in [19], the color-texture feature is extracted based on the integration of the Gabor filters with the measurement of color in the wavelength-Fourier domain, and it is demonstrated to be accurate in capturing texture statistics. Yang *et al.* [20] propose a more involved color-texture integration strategy based on compression-based texture merging, which extracts the color-texture features at pixel level by stacking the intensity values within a $7 \times 7$ window for each band of the converted CIELAB image. Han *et al.* [21] propose a graph cuts based color-texture cosegmentation method, where a comprehensive texture descriptor is designed by integrating the nonlinear compact multi-scale structure tensor and total variation flow.

Nonetheless, there are still many difficulties in extracting accurate color-texture features that can locally adapt to the variations in the image content. Because features are only observable on a certain range of scales, it is obviously not a good solution for most of the methods that use a fixed local window to extract features from images. For an image with multiple types of textures, the use of non-adaptive local windows inevitably leads to blurred regions with small-scale textures and misses the complete information in the regions with large-scale textures.

To solve the problems above, we design a new segmentation method for color-texture images which includes the following four aspects: proposing a more effective color-texture descriptor, extracting local scales adaptively, correcting the descriptors near boundaries, and merging homogeneous or small regions. The segmentation method is described as follows.

At first, a novel color-texture feature descriptor (we name it local histogram descriptor and it is abbreviated as LH) is designed, which is based on color quantization techniques and local color histograms within circular neighborhoods. In order to make the LH descriptors suitable for textons of different scales, the scales of the receptive fields of the LH descriptors are adaptively set based on the similarity between adjacent regions. Because the LH descriptor is used for performing statistical operations within circular neighborhoods, the LH descriptors of the pixels near texture boundaries cannot accurately describe the pixels' texture features. To solve this problem, we propose a texture boundary detector to obtain the texture boundaries, which are used to guide the correction of the LH descriptors for pixels near texture boundaries. And then by clustering the LH descriptors at suitable scales using the density-based spatial clustering of applications with noise (DBScan) [22] method, an initial segmentation result is obtained. Although there may be severe over segmentations, the segmentation provides a good basis for the subsequent operation, i.e., region merging. Based on the mean value of the LH descriptors on each initial segmentation region and obtained texture boundaries, a region merging method for merging homogenous and small regions is designed to obtain the final segmentation result.

Compared with previous unsupervised color-texture segmentation methods, the main contributions of our method are twofold. Firstly, a new texture feature descriptor is designed, which can effectively and robustly handle not only complex shapes but also multiscale textures. Secondly, due to the coupling with reliable boundaries, the image segmentation can effectively eliminate the risk of over-segmentation because of the slow change of brightness.

The rest of the paper is organized as follows. Related work is reviewed in Section II. In Section III, we describe how the proposed image segmentation method is constructed. Section IV presents quantitative comparisons of our method with existing image segmentation methods. We conclude in Section V.

## II. RELATED WORK

Existing color-texture image segmentation algorithms can be generally classified into three major families, i.e., contour-based, region-based, and machine learning based approaches.

Early contour-based methods try to detect edges by convolving a grayscale image with edge detection operators, such as Sobel, Roberts, Prewitt, and Canny. These detectors tend to overemphasize small, unimportant edges which are often caused by repeated or stochastic textures. More recent contour-based methods take into account color and texture information for cue combination. Martin *et al.* [23] define a linear operator *Pb*, which provides the probability of a boundary at each image location and orientation by the means that the features of the local brightness, color, and texture cues are fed into a regression classifier that predicts edge strength. Based on the *Pb* detector and combined with local and global image information, a multiscale and global *gPb* contour detector is defined, and then the oriented watershed transform and ultrametric contour map techniques are used for converting contours to a hierarchical region tree. The edge detection method has a good effect on image segmentation with large contrast between regions. But it is difficult to segment images with no clear or many boundaries, and it is difficult to produce a closed curve or boundary. Besides, comparing with other methods, contour-based methods are more susceptible to noise.

Region-based methods look for the similarity among spatially connected pixels and group them into multiple homogeneous regions based on feature descriptors. Deng and Manjunath [24] designed a fully unsupervised color-texture segmentation method (referred to as JSEG), which implements a multiscale region growing method with a novel segmentation criterion. One weakness of JSEG is that it has no good solution for the over-segmentation problem. In [16], a mean-shift approach is adopted for the analysis of a complex multimodal feature space and for delineating arbitrarily shaped clusters in the application of color image segmentation. The technique has a small number of parameters and is very robust. Based on the normalized cut algorithm [25], the authors in [26] propose a spectral segmentation method with multiscale graph decomposition, which provides high-quality segmentations efficiently. Min *et al.* [27] proposed a level set segmentation model

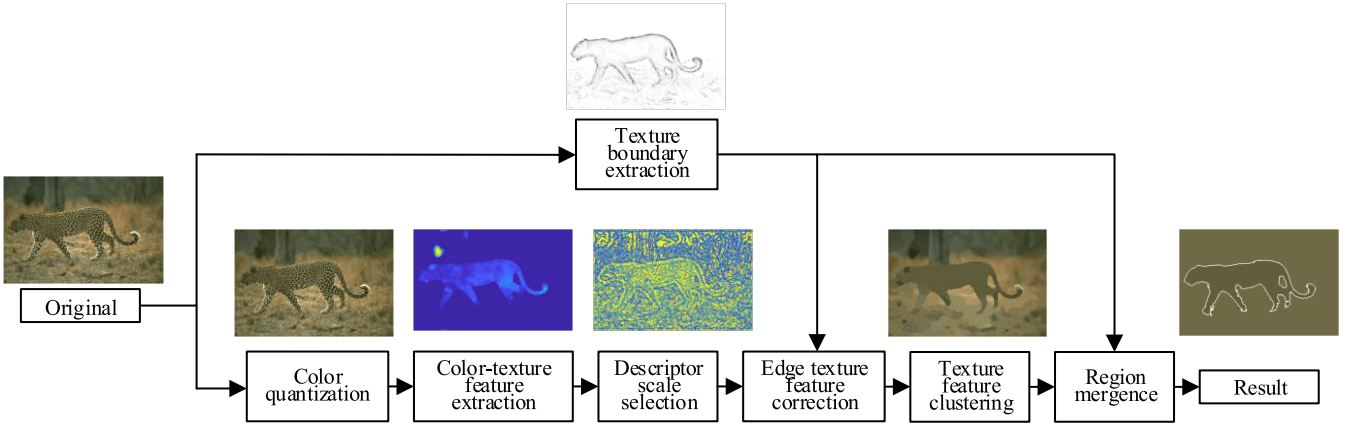



**FIGURE 1.** Diagram of feature extraction and fusion.

integrating intensity and texture terms for segmentation of natural images, which can better capture intensity information of images than the Chan-Vese model [28], and the texture feature is extracted by the adaptive scale local variation degree algorithm. Gao *et al.* [29] proposed a factorization-based level set model for texture image segmentation (referred to as FACM), which utilizes the local spectral histogram as the texture features and establishes an energy function based on the theory of matrix decomposition.

In addition, image segmentation based on machine learning is also a hot topic. In recent years, there are great advances in supervised machine learning based image segmentation using deep learning methods such as U-Net [30] and RNN [31], [32], and these methods have greatly improved segmentation performances compared with traditional methods. Despite this, these supervised machine learning based methods require a large number of manually annotated training samples, and obtaining such samples is still a time-consuming task. There are also some image segmentation methods based on unsupervised machine learning. For example, Xu *et al.* [33] improved the pulse coupled neural network model, which can only be used for grayscale images, to a method that can be used for color images. Unsupervised methods reduce the need for manually annotating datasets, but these methods are not mature enough to achieve accurate segmentation results. In addition, for both supervised and unsupervised machine learning based image segmentation methods, their memory usage and computational time are always excessive during the training process.

## III. PROPOSED METHOD

In this section, the proposed method for color-texture image segmentation is introduced in detail, and it includes the following four parts: (1) multiscale color-texture feature extraction based on the LH descriptor, (2) scale selection of the LH descriptor, (3) correction of the LH descriptors near texture boundaries and initial segmentation, and (4) region merging.

### A. COLOR-TEXTURE FEATURES

Intuitively, texture is composed of locally repeating textons. Hence a local histogram within a small region can provide a robust mechanism for modeling the distribution of the textons. Inspired by the characteristics of local repeatability of textures, a feature descriptor based on local color histogram is designed. We give a definition on the descriptor as follows:

Taking each pixel as the center of a circular neighborhood in a given color-quantized image, the number of pixels with different intensities is counted within the circular neighborhood. Then a local histogram is obtained for collecting the occurrences of each intensity value. The obtained local histogram is named as the LH descriptor $H(x_c, y_c, r)$ at the central point, where $(x_c, y_c)$ is the pixel coordinate of the center point, and $r$ is the radius of the circular neighborhood.

For a given gray-quantized image $\bar{I}(x, y)_K$ with $K$ gray levels, it can be seen that the LH descriptor is a $K$-dimension vector. We use $H_K$ to represent an LH descriptor with $K$ dimensions, and its $k$th element can be expressed by

$$h(x_c, y_c, r)_k = \sum_{(x,y)\in N(x_c,y_c,r)} \bar{I}(x, y)_k \tag{1}$$

where $N(x_c, y_c, r)$ is a circular neighborhood with center point $(x_c, y_c)$ and radius $r$. $\bar{I}(x, y)_k$ is a binary image and the value of pixel $(x, y)$ in the image is 1 when it belongs to the $k$th gray level.

Moving along the image for every pixel, all the LH descriptors of the image form a hyperplane, named as an LH image for simplicity. The LH image provides local texture information for each pixel and its circular neighborhood for both grayscale and color images. For a given color-quantized image, the LH descriptor first converts the color image into multiple separate feature channels and then processes each channel independently. Lastly, the outputs of all the channels at each pixel are concatenated. For example, the multichannel LH descriptor for pixel $(x_c, y_c)$ in the RGB color space is obtained and is denoted by:

$$H(x_c, y_c, r)_{S,K} = \left[H_{R,K_R},\ H_{G,K_G},\ H_{B,K_B}\right] \tag{2}$$

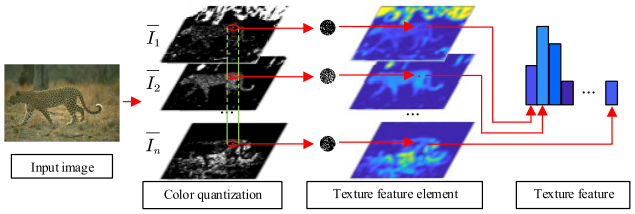



**FIGURE 2.** Feature extraction process using the LH descriptor.

$$K = \sum_{i \in \{R,G,B\}} K_i \tag{3}$$

where $S$ denotes the color space, $K$ denotes the vector dimension, and $H_{i,K_i}$ ($i \in \{R, G, B\}$) denotes the LH descriptor in the $i$th channel with $K_i$ quantized bins. That is, the LH descriptor can be seen as a local gray histogram within a circular neighborhood, and is extended to exploit color information. Then, we form the final feature vector for the input image by concatenating all these LH descriptors as shown in Fig. 2.

The feature extraction using the LH descriptor includes two processes, color quantization and obtaining the local histograms.

#### 1) COLOR QUANTIZATION

Color histograms have been widely used to capture image properties for various computer vision applications. However, there are issues with the color histograms to be directly used for texture segmentation. Firstly, the dimensionality of color histograms is too high, and this will present a challenge computationally. Furthermore, the majority of the histogram bins would be empty, which may result in unreliable estimations on the texture features. Besides, even in the same texture of a natural image, there may be a small difference in lightness, noise, etc. Thus we should classify pixels with similar colors into one class rather than classify pixels with exactly the same color into one class.

Hence, a color quantization technique is adopted before obtaining color histograms. Color quantization algorithms mainly contain two parts, i.e., palette design and pixel mapping. The process of palette design reduces the color complexity of the original image, yet keeps a sufficient number of representative colors. In this paper, the mean shift clustering algorithm [34] is used in the design of color palette and pixel mapping. After the pixels of the image are clustered in the RGB color space by the mean shift clustering algorithm, the cluster center obtained is the color palette, and the correspondence between each pixel and its corresponding cluster center is the pixel mapping, which is to find the closest color from the color palette to represent the original colors of the input image with minimum distortion.

#### 2) LOCAL IMAGE STATISTICS

The high dimensionality problem of the traditional color histograms can be alleviated by color quantization techniques.

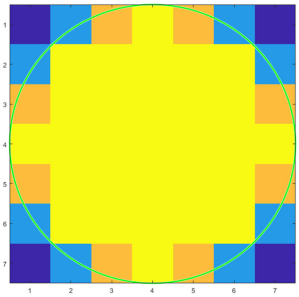


**FIGURE 3.** A circular mean filtering with $r = 3.5$.

Now there is another issue which needs to be addressed. That is, the color histogram of the entire image does not include any spatial distribution of colors. Hence the shape and texture information of the image may not have been fully used. To overcome this problem, the local color histogram is computed within a circular neighborhood. As expressed in (1), for each bin of the local histogram, the counting can be obtained by convolving a mean filtering $C(r)$ with a binary image, which is obtained by selecting the corresponding representative color in the quantized image. The values for $C(r)$ can be obtained in two steps: (1) The values of the pixels completely within the circular boundary are the same, and equal to 1, which are denoted by the yellow pixels in Fig. 3. The values of the pixels partially within the circular boundary are estimated by an interpolation operation, and their values are equal to the percentage of the area within the circular boundary, which are denoted by the orange and light blue pixels in Fig. 3. (2) All the values for $C(r)$ are normalized to make their summation equal to 1.

### B. DESCRIPTOR SCALE SELECTION

The size of the LH descriptor is closely related to the scale of textons. If the size of the LH descriptor is smaller than the scale of the textons, the obtained information about the textons is incomplete and with missing details. In addition, a real-world image usually contains more than one types of textons, and the scale of a texton may vary with the imaging angle and the undulating surface of the object. For example, Fig. 4 shows a color-texture zebra image where there is only one kind of texton on the zebra but with different scales. The neck area of the zebra contains large-scale textures, while the hip and waist of the zebra contains small-scale textures. When we obtain the LH descriptors with a single scale for the zebra image, it can be seen intuitively that if the scale is small, the LH descriptor is inadequate to represent the texture information on the zebra neck, while if the scale is large, the LH descriptor blurs the texture boundary on the hip and waist of the zebra. Therefore, it is necessary to set an adaptive scale for the LH descriptor in different areas of a given image.

In general, when the LH descriptor at a pixel is expanded to a larger scale, there will be more pixels to be included in the larger circular neighborhood. Based on their locations, these pixels can be divided into two categories, i.e., pixels within

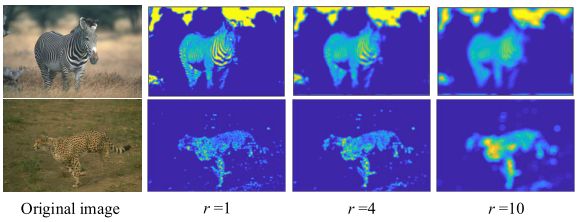



**FIGURE 4.** Two examples of visual results obtained by an increasing scale for the LH descriptor.

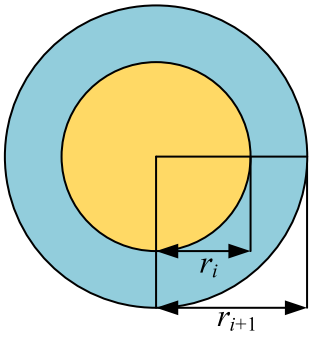



**FIGURE 5.** Separability between two categories of pixels. One category which is within a circular neighborhood is denoted by a yellow region and the other category is denoted by a blue region.

a circular neighborhood and pixels on the ring, as shown in Fig. 5. If the two categories of pixels are homogenous, they are inseparable; otherwise, they are separable. So, the separability can be used to find a suitable scale. In order to obtain a quantitative measure on how separable the two categories are, a distance measure is required. In this paper, the Bhattacharyya distance [35] is used to represent the separability. For each pixel $(x, y)$ in a given image, the Bhattacharyya distance between the color-texture feature descriptors $h(x, y, r_i)$ and $h(x, y, r_{i+1})$ on adjacent scales is calculated as follows.

$$BC(x, y, r_i, r_{i+1}) = -\ln\left(\sum_{k=1,\cdots,K}\sqrt{h(x, y, r_i)_k \cdot h(x, y, r_{i+1})_k}\right) \quad (4)$$

Fig. 6 shows how to select the scale using the Bhattacharyya distance, and the red dots represent the selected scales in a series of radii with a pre-set range. Fig. 6(b) and Fig. 6(c) represent two different cases, i.e., pixel (169, 141) located in a region with the same texture and pixel (230, 137) located in a changing texture region respectively. As shown in Fig. 6(b), with the scale of the receptive field increasing within a certain range, the Bhattacharyya distance tends to become smaller and finally stabilizes. The Bhattacharyya distance is relatively small with the increasing scale, which indicates that the textures in the receptive field are always similar, so the largest circle is taken as the receptive field to obtain more accurate texture description information. While in the second case, as shown in Fig. 6(c), with the increasing

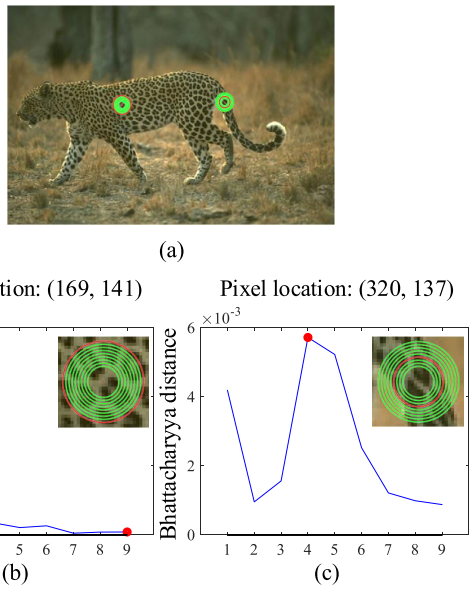



**FIGURE 6.** Different scales of the LH descriptors at different pixels. (a) The locations of the two pixels in the test image. (b) The plot of the Bhattacharyya distance at pixel (196, 141). (c) The plot of the Bhattacharyya distance at pixel (320, 137).

scale, the Bhattacharyya distance becomes higher between the 4th and 5th circles because on the 5th circle it detects a new texture. So, the receptive field for pixel (230, 137) is set at the 4th circle.

### C. CORRECTION AND INITIAL SEGMENTATION

In general, the positions of pixels in a given multi-texture image can be divided into two categories, i.e., pixels far away from the texture edges and pixels near or on the texture edges. For the first category of pixels, the LH descriptors can be effectively and accurately extracted within their circular neighbors. While for the second category of pixels, there may be more than one types of textons within their default circular neighbors. If the LH descriptor is directly used in image segmentation, it may lead to uncertain edges in the segmentation results. Therefore, for pixels in the second category, the LH descriptors need to be corrected. After correction, the LH descriptors can be used to obtain a initial segmentation result.

#### 1) CORRECTION OF LH DESCRIPTORS

To obtain the corrected receptive fields, we first design a texture boundary detection method, which is described as follows.

In [23], Martin *et al.* define a gradient-based boundary detector *Pb* which provides the posterior probability of a boundary with orientations at each image pixel. Based on the *Pb* detector, we propose a variation of the detector *Pb* for our application, which is described below.

For each pixel on the color-quantized image $\bar{I}(x, y)$, the circular neighborhood centered on the pixel is divided into two half-disk $g_\theta$ and $h_\theta$ by a line segment at an angle $\theta$ through this pixel. For each half-disk in the RGB color space,

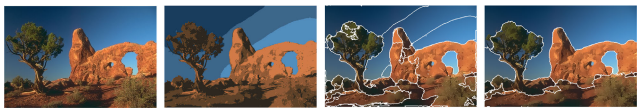


**FIGURE 7.** Merging of color quantized images. From left to right: original image, quantized image, merging result guided by texture boundaries, and merging result guided by optimized texture boundaries.

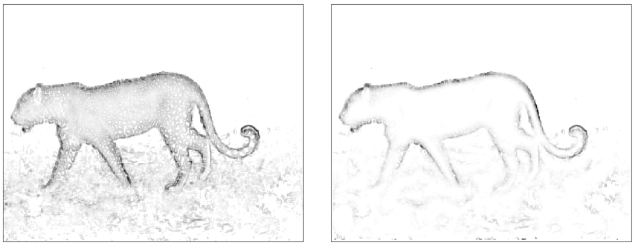


**FIGURE 8.** Comparison of extracted boundary results between our method and the method in [23]. It can be seen that our boundary map (right) is smoother in texture compared with the boundary map obtained by the method in [23] (left).

the histograms of every color channel are obtained. The $Pb$ with an arbitrary orientation is described as the histogram difference, which is computed by

$$G(x, y, r, \theta) = \frac{1}{2} \sum_{k=1,\cdots,K} \frac{(g_\theta(k) - h_\theta(k))^2}{g_\theta(k) + h_\theta(k)} \tag{5}$$

where $K$ denotes the number of color-quantized bins, and $r$ denotes the radius of $g_\theta$ and $h_\theta$. We use eight orientations for $\theta$: 0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135°, and 157.5°, and obtain a multi-orientation $Pb$ detector by adding eight values from the $Pb$ detector together. To adapt to different scales of textons, same with the LH descriptor, the $Pb$ detector is extended to multiscales by changing the radius $r$ of the circular neighborhood in a given range. The multiscale $Pb$ detector is the result of all the responses from the $Pb$ detectors in a given range of radius from $r_1$ to $r_n$, which is expressed by

$$mPb(x, y) = \prod_{r=r_1,\cdots,r_n} \sum_{\theta} G(x, y, r, \theta) \tag{6}$$

where $r_i$ is the radius of the $i$th scale of the $Pb$ descriptor.

We then apply the $mPb$ detector to the color-quantized image, and the initial texture boundaries are obtained. After that, the gradient edges of the given image are used to eliminate false boundaries generated by color quantization as shown in Fig. 7. Then, the texture boundaries are extracted to be used at a later stage. There is a difference in (6) from [23] in that the summation operation for the $Pb$ detectors with different scales is replaced by a multiplication operation to better blur the internal structure of the texture, as shown in Fig. 8.

After obtaining the boundary map, long boundaries are selected from the boundary map for correcting the descriptors for the pixels near boundaries because large regions are generally enclosed by long boundaries. For each long

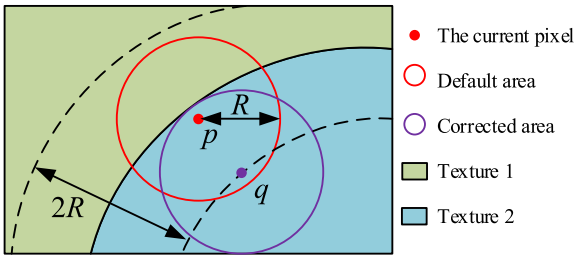



**FIGURE 9.** The receptive field of a pixel near a texture boundary: the red dot denotes the current pixel and the red circle denotes the minimum receptive field of the red dot. The purple circle denotes the corrected receptive field of the current pixel. The region inside the imaginary lines denotes B-ROM.


**FIGURE 10.** The effect of the correction of the LH descriptors. (a) Filtered image without correction. (b) Filtered image with correction.

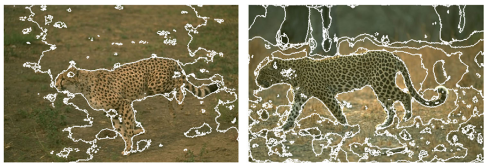


**FIGURE 11.** Initial segmentation results of two images with complex textures.

boundary and its neighborhood, there is a banded region of mixture (B-ROM), which is centered at the boundary location with a band with the width of $2R$ as shown in Fig. 9. For each pixel in B-ROM, the receptive field of its LH descriptor is changed to a circular neighborhood that is tangent to the contour and with a radius of $R$.

As shown in Fig. 10, we use the average filter instead of the LH descriptor to intuitively observe the corrected effect of the receptive field. After correcting the LH descriptors near boundaries, the edges of the feature map become sharper.

#### 2) INITIAL SEGMENTATION

We use a DBScan-based [22] approach to build local clusters of pixels with the LH descriptors and the normalized $x, y$ pixel coordinates, which are expressed by multiplying the coordinates by a fixed value. Indeed, the method can be initially used in this step when we apply the method of segmenting the image into several regions. However, to increase the robustness of the algorithm, we perform an over segmentation and then merge the regions. In Fig. 11, we can see the results of initial segmentations.

### D. REGION MERGING

The initial segmentation described earlier divides the image into many small regions. Although there may be severe over

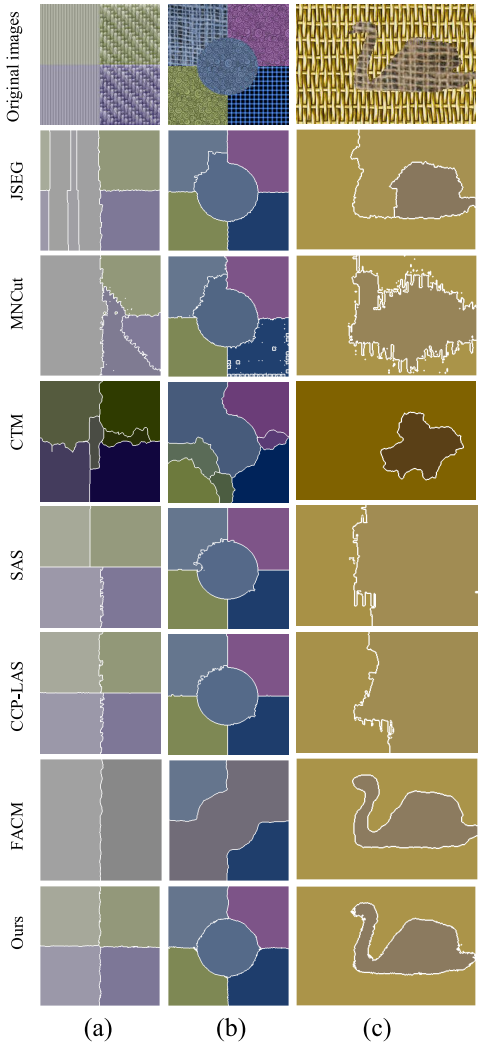



**FIGURE 12.** Comparison of segmentation results on special texture images. (a) A test image with rows differing in color and columns differing in texture. (b) A test image synthesized from five sub-images with different colors and textures. (c) A test image with two irregular-shape sub-images.

segmentations, the initial segmentation provides a good basis for region merging. In this section, we propose a fast and efficient region merging method based on the texture boundaries and initial segmentation. That is, the two nearby initial segmentation regions are determined whether to merge into one by two merging criterions, i.e., the texture boundaries and similarity in the mean value of the LH descriptors (mLH, for short) between two nearby initial segmentation regions. Texture boundaries can help avoid over segmentation because of brightness variations. The similarity is represented by the Bhattacharyya distance between the mLH descriptors. The merging process is as follows: first the mLH descriptors of each initial segmentation region are calculated; then the mLH descriptors are used to calculated the Bhattacharyya distance of each two nearby regions; at last, only when the Bhattacharyya distance between the two nearby regions is smaller than a certain threshold and there is no texture boundary between them, the two regions are merged into one. In addition, the threshold is obtained from a large number of practical experiments.

**FIGURE 13.** Twenty mosaic images from the DTD database.

## IV. EXPERIMENTS

In this section, we evaluate the performance of the proposed method on one synthetic and two natural image databases. The describable textures database (DTD) [36] is selected as a synthetic image test, and the BSDS500 dataset [37] and the Weizmann database [38] are selected as natural image tests for practical image segmentations.

Besides, both subjective and objective evaluation methods are used in these evaluations. The subjective evaluation method is to compare and analyze the segmentation results of each segmentation method with reference to ground truth. As for objective evaluation, we select four popular and effective metrics [39], i.e., the probabilistic rand index (PRI) [40], the variation of information (VoI) [41], the global consistency error (GCE) [42], and the boundary displacement error (BDE) [43]. PRI represents the agreement between the obtained segmentation and the ground truth. VoI measures the distance between two segmentations in terms of their average conditional entropy, thereby roughly measuring the amount of randomness in one segmentation that cannot be explained by the other. GCE measures the extent to which one segmentation can be viewed as a refinement of the other. BDE measures the average displacement between the boundaries of two segmented images. We should notice that PRI ranges in [0, 1], and a larger value is better; VoI varies in $[0, +\infty)$, and a smaller value is better; GCE ranges in [0, 1], and a smaller value is better, and BDE varies in $[0, +\infty)$ in pixels, and a smaller value is better. Therefore, a segmentation is better if PRI is larger and the other three metrics are smaller when compared with the ground truths.

Our proposed method is compared with six other classical and advanced segmentation techniques, i.e., the unsupervised J-images segmentation (JSEG) [24] method, the multiscale

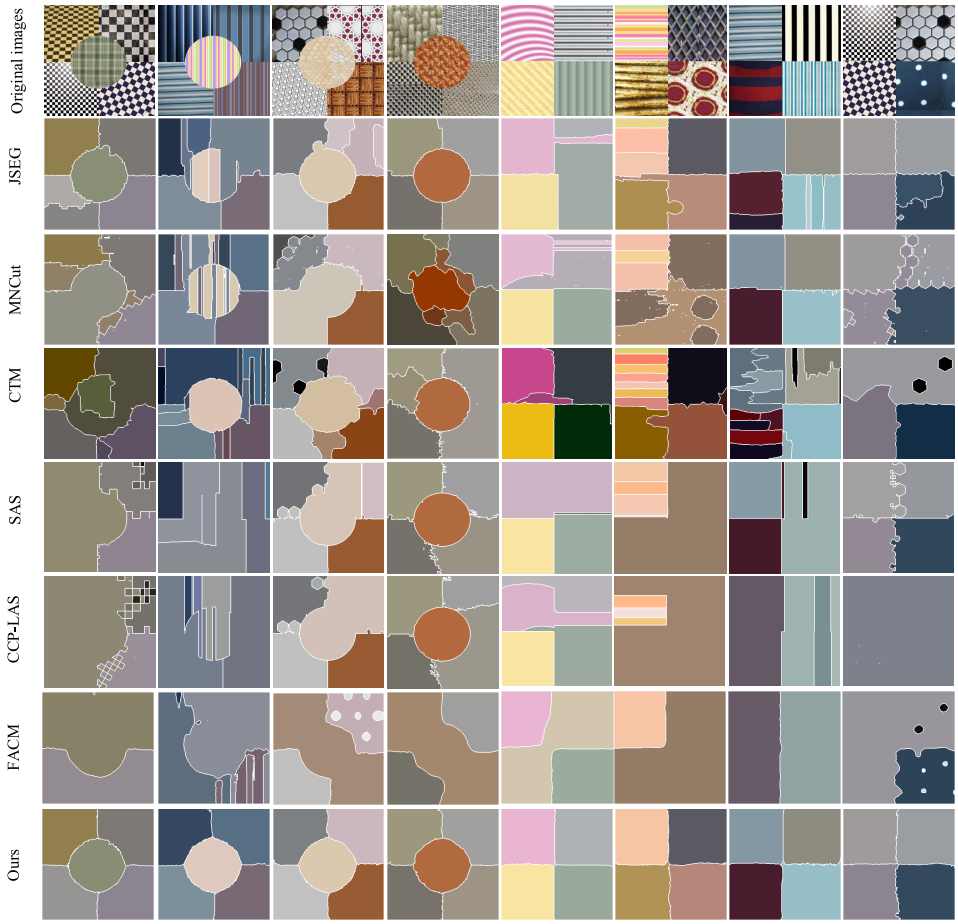



**FIGURE 14.** Visual comparison of the results obtained with the seven methods on the DTD database.

normalized cuts (MNCut) [26] method, the compression-based texture merging (CTM) [20] method, the segmentation by aggregating superpixels (SAS) [44] method, the contour-guided color palettes based (CCP-LAS) [45] method, and the factorization based active contour model for texture segmentation (FACM) [29] method. The test results will be given and analyzed in the following section.

### A. PARAMETER SETTING

There are a few parameters that can be set to influence the segmentation results for our method. Most of the parameters can be set to fixed values. The fixed parameters used in the LH descriptor are set as follows: The radius $r$ shown in (1) are from 5 to 14, and the step size for two adjacent filters is 1. The number of segmented regions on the input image is set to 100 to obtain an over segmentation result to maintain segmentation details.

For the purpose of fair comparisons, the main parameter settings for the six competing methods are given below. The JSEG method has three parameters that should be set, i.e., the color quantization threshold, the scale, and the merge threshold. Just as the authors suggested, these parameters can be set to 255, 1.0, and 0.4. The number of segmented regions for MNCut is set appropriately according to the image content. The CTM method is a simple agglomerative clustering algorithm derived from a lossy data compression approach, and the texture difference threshold is set to 0.15. For the SAS method, the affinity coefficient, the scale factor, and the neighbor numbers of super pixels are set to 0.001, 20, and 1 respectively. For the CCP-LAS method, we select the spectral BW parameter as 5, as the authors used in their paper. For the FACM method, the two balance coefficients are set to 1 and 0.04, respectively, the integration scale is set to 20, the number of bins of histogram is set to 12, and the initial contour divides the image into 9 identical $3 \times 3$ sub-images.

### B. EXPERIMENT ON SYNTHETIC IMAGES

The synthetic image segmentation task is tested firstly on some specifically designed synthetic images, which are from the DTD database. Fig. 12 shows the comparisons of the segmentation results among our method, benchmark JSEG, MNCut, CTM, SAS, CCP-LAS, and FACM.

The first row of Fig. 12(a) shows a challenging synthetic color-texture image in which the row sub-images differ in color and the column sub-images differ in texture. The test is set to evaluate the capability of our proposed method on

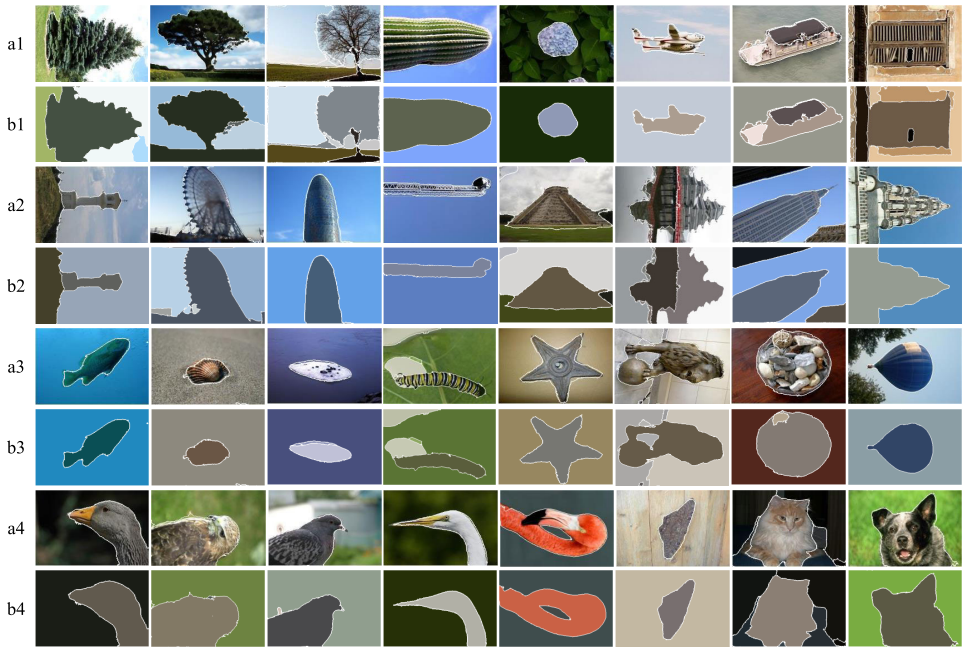



**FIGURE 15.** Thirty-two test images from the Weizmann database and results from our method. The rows a1-a4 are the original images overlaid with the boundaries of the segmentation results obtained by our proposed segmentation method, and the rows b1-b4 are the corresponding segmentation results.

discriminating different colors and textures. It can be seen that our proposed method and CCP-LAS can segment the image with different colors and textures accurately. However, both JSEG and MNCut fail in segmenting different texture areas as shown in the other rows of the first column. Although the other two methods, CTM and SAS, can also separate different colors and textures, the dividing boundaries are inaccurate. FACM can segment regions with different textures very well, but it fails in separating different colors. The first row of Fig. 12(b) shows a more complex image, which is synthesized from five sub-images with different colors and textures. On one diagonal, the three sub-images have similar color but different textures. On the other diagonal, the three sub-images have different colors but with the same texture. It can be seen that, except for the CTM and FACM methods, all other methods can distinguish the regions with the same texture but different colors. As for the regions with the same color but different textures, only the SAS, CCP-LAS, FACM, and our method can segment the image correctly, and our method performs best among these segmentation methods. Although FACM performs well in separating regions with different textures, it cannot distinguish colors well. The first row of Fig. 12(c) shows a synthesis image with two irregular-shaped sub-images, and the sub-images are with different textures but with similar colors. It can be seen that only FACM and our proposed method can separate the two sub-images with different color-texture features and locate the boundaries accurately. The segmentation results in these tests demonstrate that our approach has a much better performance than the other six methods, especially for distinguishing regions with different textures but with similar colors.

**TABLE 1.** The average values of PRI, VoI, GCE, and BDE for the seven algorithms on the synthetic images in Fig. 13.

| Algorithm | PRI | VoI | GCE | BDE |
|---|---|---|---|---|
| JSEG | 0.9246 | **0.9908** | 0.1148 | 20.9811 |
| MNCut | 0.8644 | 1.4405 | 0.2446 | 21.5578 |
| CTM | 0.8909 | 1.8960 | 0.1485 | 14.5395 |
| SAS | 0.7717 | 1.3052 | 0.1414 | 22.0476 |
| CCP-LAS | 0.7876 | 1.4361 | 0.1393 | 21.2063 |
| FACM | 0.6717 | 1.6854 | 0.1206 | 32.2371 |
| Ours | **0.9446** | 1.3756 | **0.0945** | **12.3513** |

Meanwhile, more synthetic color-texture images are chosen from the DTD dataset for testing. Compared with the images in Fig. 12, the color-texture features in these images are more random, irregular, and complex, as shown in Fig. 13. In this experiment, the numbers of segmented regions for MNCut, SAS, CCP-LAS, and our method are all set to 4 in rows 1-2 and 5 in rows 3-4 in Fig. 13. The parameters for the other three methods, JESG, CTM, and FACM, are adaptively adjusted by their algorithms. As shown in Fig. 14, our proposed method performs best in the comparison results, especially in accurately locating the region boundaries.

In addition, to objectively evaluate the segmentation results from our proposed method compared with other methods, the PRI, VoI, GCE, and BDE values are obtained and given in Table 1.

As shown in Table 1, the segmentation result of our method is the most accurate. JSEG segments well but it obtains a bad BDE result. MNCut has a bad result in GCE because it generates more errors. As a segmentation method based on texture, CTM achieves a good score in BDE, but it is not able

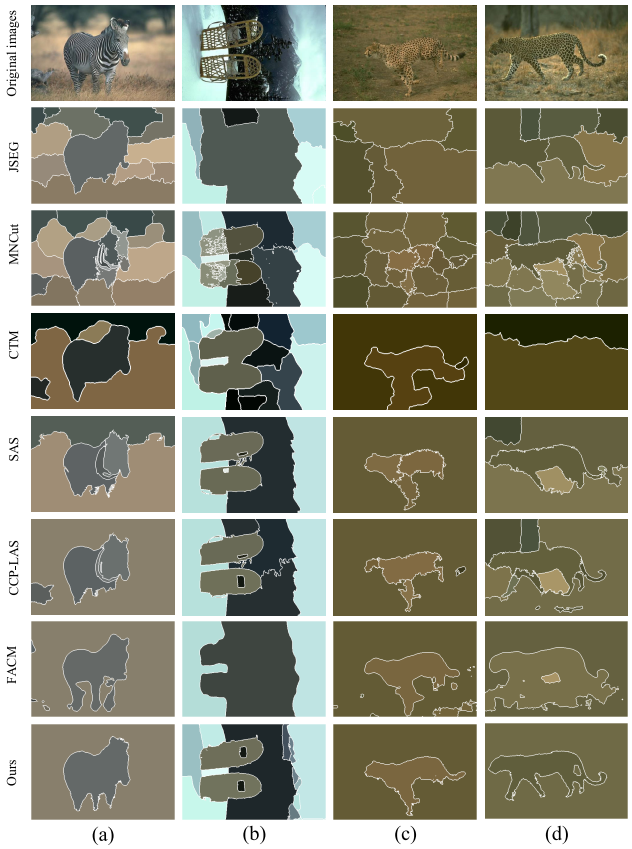



**FIGURE 16.** Comparison of segmentation results between our method and other six methods on four real-world images with rich color-texture information. (a) Zebra. (b) Snowshoes. (c-d) Leopard.

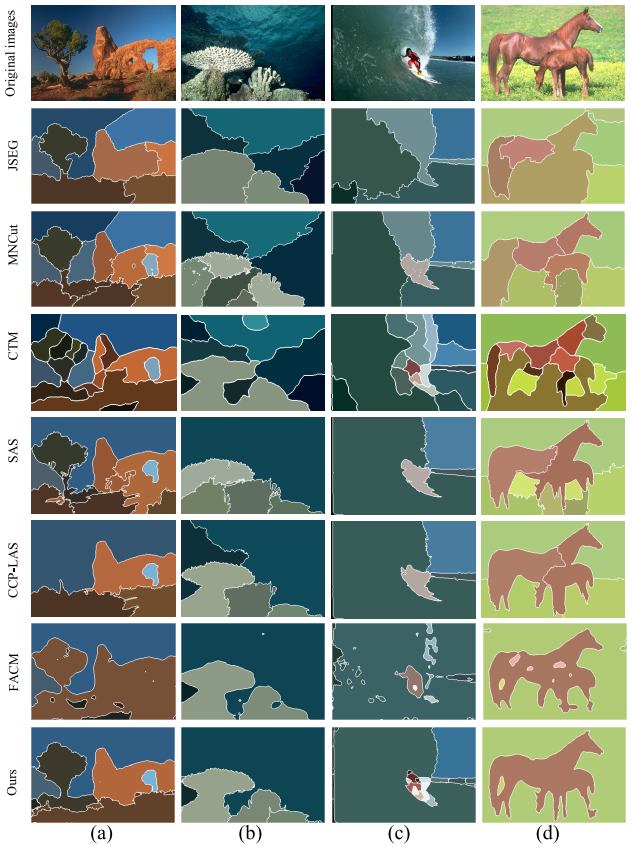



**FIGURE 17.** Comparison of segmentation results between our method and other six methods on four real-world images with gradually changed brightness. (a) Sky. (b) Sea. (c) Surfing. (d) Horses.

to segment the texture images into correct regions, so it has a bad VoI score. SAS and CCP-LAS are based on boundaries and they are not good at texture image segmentation. FACM only achieves a good score in GCE because many of its results are under-segmented.

As described earlier in the comparative experiments, for the qualitative comparisons in Figs. 12-14 or the quantitative comparisons in Table 1, our proposed method has a superior performance compared with the other color segmentation methods in the aspects of accuracy, robustness, and color-texture discriminating capability.

### C. EXPERIMENT ON NATURAL IMAGES

In this subsection, we validate the effectiveness of our proposed method on two natural image datasets, i.e., the Weizmann database and the Berkeley dataset. Both of them include a set of real-world images with diverse, complex, and irregular color-texture patterns. The experiments are conducted to obtain quantitative and qualitative evaluations on the performances of our proposed method. Meanwhile, the JSEG, MNCut, CTM, SAS, CCP-LAS, and FACM algorithms are also chosen for comparisons.

We first begin validating our proposed method on 32 color-texture images from the Weizmann dataset, and the segmentation results from our method are given in Fig. 15, which indicates that our method extracts the objects with accurate and smooth boundaries. There are some test images which contain rich and complex textures, such as plant, buildings, and animals, and there are some test images which contain gradual brightness changes, such as the sky behind the ferris wheel and water surrounding an island. These experimental results show that our method is not only robust against brightness changes, but also can discriminate regions with rich color textures.

We next test some more complex color-texture images from the well-known Berkeley segmentation database (BSDS500). The images considered in this section are characterized by nonuniform textures, fuzzy borders, and low image contrast, and known color-texture segmentation methods often obtain undesirable results.

In order to demonstrate the effectiveness of our method in handling images containing rich and complex textures, four images (as shown in Fig. 16) are selected from the BSDS500 database. The comparisons of our proposed method and the other six segmentation methods are given in Fig. 16. In Fig. 16(a), a zebra image is tested, which contains different-scale strip textures. For instance, the texture in the waist region is very dense, while the scale of the strip texture


**FIGURE 18.** Forty-eight test images from the Berkeley segmentation database. The rows a1-a6 are original images overlaid with the boundaries of the segmentation results obtained by our proposed segmentation method, and the rows b1-b6 are corresponding segmentation results.

in the hip and neck regions is relatively large. Therefore, it is very suitable to demonstrate the superior scale discriminating power of our proposed method. It can be seen from Fig. 16(a) that, benefiting from the multiscale extraction of the LH descriptor, our method can separate the zebra as an entire object. However, the other six segmentation methods can only segment parts of the zebra, especially in regions with low color contrast and fine details, e.g., the zebra fore legs which are blurred by grass in the background. Besides, both SAS and CCP-LAS produce over-segmentation results on the neck of the zebra due to its relatively larger scale. Moreover, in Fig. 16(b), our method can not only discriminate the large-scale texture as a single area, but also preserve more details. Hence, the two rectangular holes are separate in the pair of snowshoes. In Fig. 16(c-d), our method can extract camouflaged leopards as an entire object from their surroundings. As for the other six compared methods, JSEG, MNCut, and CTM totally fail in these tests, although SAS, CCP-LAS, and FACM can segment parts of the leopards, and their segmentation results appear to be seriously over-segmented and wrongly segmented to some extent.

Another problem in image segmentation is that it is easy to produce over-segmentation in regions with gradually changed brightness by many color image segmentation methods. In Fig. 17, four images are tested, which contains rich color-texture information, and the brightness is gradually changing in some regions. Based on the boundary-guided region merging, our method is robust against brightness variations. For the textures with gradually changed brightness, for example, the sky and sea regions as shown in Fig. 17(a-c), our method

**TABLE 2. The average values of PRI, VoI, GCE, and BDE for the seven algorithms on the Berkeley segmentation database.**

| Algorithm | PRI | VoI | GCE | BDE |
|---|---|---|---|---|
| JSEG | 0.7092 | 2.2498 | 0.2700 | 19.8015 |
| MNCut-20 | 0.7174 | 2.6240 | 0.1930 | 12.4682 |
| MNCut-10 | 0.7607 | 2.3547 | 0.2494 | 13.7414 |
| CTM | 0.7220 | 2.3487 | 0.2392 | 18.2995 |
| SAS-20 | 0.7917 | 2.1410 | 0.1955 | 12.4949 |
| SAS-15 | 0.7884 | 2.0367 | 0.2064 | 12.6391 |
| SAS-10 | 0.7814 | 1.9247 | 0.2141 | 13.7763 |
| CCP-LAS | 0.7449 | 1.8902 | 0.2025 | 13.2134 |
| FACM | 0.6671 | 1.9559 | 0.1898 | 23.0955 |
| Ours | **0.8108** | **1.4620** | **0.1330** | **11.4320** |

avoids over-segmented results because of the region merging guided by boundaries. The segmentation results from the three comparison methods JSEG, MNCut, and CTM appear to be over-segmented. FACM performs well as shown in Fig. 17(b), but it appears to be under-segmented as shown in Fig. 17(a, c). Moreover, in Fig. 17(d), only our method can retain the horses as an entire object with accurate boundaries, because there is uniformity brightness on the horse back.

In order to illustrate the effectiveness of our method in handling different images, 500 images from the Berkeley database are used for testing. Part of segmentation results are shown in Fig. 18.

The experimental results on the Berkeley database are given in Table 2 which shows that the proposed method achieves a much better performance compared with the other algorithms based on the PRI, VoI, GCE, and BDE measures.

## V. CONCLUSION

In this paper, we propose a color-texture image segmentation method. To improve the color-texture discriminating capability, a new color-texture descriptor based on local histograms is designed to model the color-texture distribution. Meanwhile, the color-texture descriptor can also obtain the scale information of the textures. Afterwards, a variant of the local multiscale probability of a boundary detector is designed to extract texture boundaries. Then the texture boundaries are used for region merging. A large number of qualitative and quantitative experiment results show that our method outperforms other methods for color-texture segmentation, especially for images with rich color textures and gradual brightness changes.

## REFERENCES

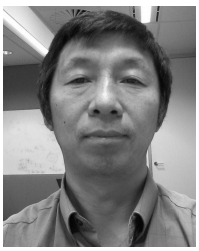
[1] S. Brahnam, L. C. Jain, A. Lumini, and L. Nanni, Eds., *Local Binary Patterns: New Variants and Applications*. Berlin, Germany: Springer, 2014.
[2] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognit.*, vol. 24, no. 12, pp. 1167–1186, 1991.
[3] Y. Yang, S. Han, T. Wang, W. Tao, and X.-C. Tai, "Multilayer graph cuts based unsupervised color–texture image segmentation using multivariate mixed student's $t$-distribution and regional credibility merging," *Pattern Recognit.*, vol. 46, no. 4, pp. 1101–1124, 2013.
[4] J.-S. Kim and K.-S. Hong, "Color–texture segmentation using unsupervised graph cuts," *Pattern Recognit.*, vol. 42, no. 5, pp. 735–750, 2009.
[5] H. Wang, H. Zhang, and N. Ray, "Adaptive shape prior in graph cut image segmentation," *Pattern Recognit.*, vol. 46, no. 5, pp. 1409–1414, 2013.
[6] H. Zhou, J. Zheng, and L. Wei, "Texture aware image segmentation using graph cuts and active contours," *Pattern Recognit.*, vol. 46, no. 6, pp. 1719–1733, Jun. 2013.
[7] Y. Yang, L. Guo, T. Wang, W. Tao, G. Shao, and Q. Feng, "Unsupervised multiphase color–texture image segmentation based on variational formulation and multilayer graph," *Image Vis. Comput.*, vol. 32, no. 2, pp. 87–106, 2014.
[8] Q. Cai, H. Liu, S. Zhou, J. Sun, and J. Li, "An adaptive-scale active contour model for inhomogeneous image segmentation and bias field estimation," *Pattern Recognit.*, vol. 82, pp. 79–93, Oct. 2018.
[9] B. Han and Y. Wu, "Active contours driven by global and local weighted signed pressure force for image segmentation," *Pattern Recognit.*, vol. 88, pp. 715–728, Apr. 2019.
[10] J. Sigut, F. Fumero, O. Nuñez, and M. Sigut, "Automatic marker generation for watershed segmentation of natural images," *Electron. Lett.*, vol. 50, no. 18, pp. 1281–1283, Aug. 2014.
[11] C. Zhang, C. Sun, R. Su, and T. D. Pham, "Segmentation of clustered nuclei based on curvature weighting," in *Proc. 27th Conf. Image Vis. Comput.*, Dunedin, New Zealand, Nov. 2012, pp. 49–54.
[12] Y. Liu and Y. Yu, "Interactive image segmentation based on level sets of probabilities," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 2, pp. 202–213, Feb. 2012.
[13] H. Min, L. Xia, J. Han, X. Wang, Q. Pan, H. Fu, H. Wang, S. T. Wong, and H. Li, "A multi-scale level set method based on local features for segmentation of images with intensity inhomogeneity," *Pattern Recognit.*, vol. 91, pp. 69–85, Jul. 2019.
[14] J. Tang, "A color image segmentation algorithm based on region growing," in *Proc. 2nd Int. Conf. Comput. Eng. Technol.*, vol. 6, Apr. 2010, pp. V6-634–V6-637.
[15] Y. Q. Zhao, X. H. Wang, X. F. Wang, and F. Y. Shih, "Retinal vessels segmentation based on level set and region growing," *Pattern Recognit.*, vol. 47, no. 7, pp. 2437–2446, 2014.
[16] H. Cho, S.-J. Kang, and Y. H. Kim, "Image segmentation using linked mean-shift vectors and global/local attributes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 10, pp. 2132–2140, Oct. 2017.
[17] M. S. Brown, L. S. Wilson, B. D. Doust, R. W. Gill, and C. Sun, "Knowledge-based method for segmentation and analysis of lung boundaries in chest X-ray images," *Computerized Med. Imag. Graph.*, vol. 22, no. 6, pp. 463–477, 1998.
[18] Z. Liu, C. Cao, S. Ding, Z. Liu, T. Han, and S. Liu, "Towards clinical diagnosis: Automated stroke lesion segmentation on multi-spectral MR image using convolutional neural network," *IEEE Access*, vol. 6, pp. 57006–57016, 2018.
[19] M. A. Hoang, J.-M. Geusebroek, and A. W. M. Smeulders, "Color texture measurement and segmentation," *Signal Process.*, vol. 85, no. 2, pp. 265–275, Feb. 2005.
[20] A. Y. Yang, J. Wright, Y. Ma, and S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Comput. Vis. Image Understand.*, vol. 110, no. 2, pp. 212–225, May 2008.
[21] S. Han, W. Xu, W. Tao, and Y. Chen, "Color-texture cosegmentation based on nonlinear compact multi-scale structure tensor and TV-flow," *Signal Process.*, vol. 131, pp. 456–471, Feb. 2017.
[22] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. KDD*, 1996, vol. 96, no. 34, pp. 226–231.
[23] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 530–549, May 2004.
[24] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 800–810, Aug. 2001.
[25] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[26] T. Cour, F. Benezit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2005, pp. 1124–1131.
[27] H. Min, W. Jia, X.-F. Wang, Y. Zhao, R.-X. Hu, Y.-T. Luo, F. Xue, and J.-T. Lu, "An intensity-texture model based level set method for image segmentation," *Pattern Recognit.*, vol. 48, no. 4, pp. 1547–1562, Apr. 2015.
[28] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, Feb. 2001.

[29] M. Gao, H. Chen, S. Zheng, and B. Fang, "A factorization based active contour model for texture segmentation," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2016, pp. 4309–4313.
[30] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.
[31] X. Xie, G. Xie, X. Xu, L. Cui, and J. Ren, "Automatic image segmentation with superpixels and image-level labels," *IEEE Access*, vol. 7, pp. 10999–11009, 2019.
[32] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1529–1537.
[33] G. Xu, X. Li, B. Lei, and K. Lv, "Unsupervised color image segmentation with color-alone feature using region growing pulse coupled neural network," *Neurocomputing*, vol. 306, pp. 1–16, Sep. 2018.
[34] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[35] C. C. Reyes-Aldasoro and A. Bhalerao, "The Bhattacharyya space for feature selection and its application to texture segmentation," *Pattern Recognit.*, vol. 39, no. 5, pp. 812–826, May 2006.
[36] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3606–3613.
[37] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2010-17, 2010.
[38] A. Sharon, G. Meirav, B. Achi, and B. Ronen, "Image segmentation by probabilistic bottom-up aggregation and cue integration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 315–327, Feb. 2012.
[39] B. Peng, L. Zhang, X. Mou, and M. H. Yang, "Evaluation of segmentation quality via adaptive composition of reference segmentations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 10, pp. 1929–1941, Oct. 2017.
[40] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 929–944, Jun. 2007.
[41] M. Meilă, "Comparing clusterings: An axiomatic view," in *Proc. 22nd Int. Conf. Mach. Learn.*, Aug. 2005, pp. 577–584.
[42] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, Jul. 2001, pp. 416–423.
[43] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí, "Yet another survey on image segmentation: Region and boundary information integration," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2002, pp. 408–422.
[44] Z. Li, X.-M. Wu, and S.-F. Chang, "Segmentation using superpixels: A bipartite graph partitioning approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 789–796.
[45] X. Fu, C.-Y. Wang, C. Chen, C. Wang, and C.-C. J. Kuo, "Robust image segmentation using contour-guided color palettes," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1618–1625.

**YANG LIU** received the B.S. degree (Hons.) from the China University of Geosciences, in 2013. He is currently pursuing the Ph.D. degree with Jilin University. His research interests include stereo vision, 3D reconstruction, and image segmentation.

**GUANGDA LIU** received the Ph.D. degree in optical engineering from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China, in 2000. He then joined Jilin University, China, where he is currently a Professor carrying out research and working on biomedical signal processing and medical instrument development. His current research interests include human blood flow parameters detection and hepatic functional reserves assessment.

**CHANGYING LIU** received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2006. He then joined Jilin University, China, where he is currently a Professor carrying out research and working on the visual measurement of railway vehicles. His current research interests include large-scale 3D precise visual measurement and 3D reconstruction.



**CHANGMING SUN** received the Ph.D. degree in computer vision from Imperial College London, London, U.K. in 1992. He then joined CSIRO, Sydney, NSW, Australia, where he is currently a Principal Research Scientist carrying out research and working on applied projects. His current research interests include computer vision, image analysis, and pattern recognition. He has served on the program/organizing committees of various international conferences. He is an Associate Editor of the *EURASIP Journal on Image and Video Processing*.