

Image Segmentation Based on Local Spectral Histograms and Linear Regression

Jiangye Yuan, DeLiang Wang, and Rongxing Li

Abstract—We present a novel method for segmenting images with texture and nontexture regions. Local spectral histograms are feature vectors consisting of histograms of chosen filter responses, which capture both texture and nontexture information. Based on the observation that the local spectral histogram of a pixel location can be approximated through a linear combination of the representative features weighted by the area coverage of each feature, we formulate the segmentation problem as a multivariate linear regression, where the solution is obtained by least squares estimation. Moreover, we propose an algorithm to automatically identify representative features corresponding to different homogeneous regions, and show that the number of representative features can be determined by examining the effective rank of a feature matrix. We present segmentation results on different types of images, and our comparison with another spectral histogram based method shows that the proposed method gives more accurate results.

I. INTRODUCTION

THE goal of image segmentation is to partition an image into a number of regions so that each region should be as homogeneous as possible and neighboring ones should be as different as possible. It is a critical task for a wide range of image analysis problems. Although substantial progress has been made in this area, image segmentation remains an unsolved problem in computer vision.

An image segmentation method should work on different kinds of images including texture and nontexture images. It is widely recognized that a visual texture is very difficult to characterize. A large number of methods have been proposed to deal with texture images, and they are mainly focused on the following two aspects. One is on filtering, which typically uses filterbanks to decompose an image into a set of sub-bands. Filtering methods show impressive performance for texture segmentation [3] [5]. The other is on statistical modeling, which characterizes texture regions as resulting from some underlying probability distributions. These methods emphasize global appearance and are robust to noise [14] [15] [18].

Building on the above two themes, a local spectral histogram consists of marginal distributions of chosen filter responses in an image window [11]. It has been shown that

local spectral histograms provide a generic statistic feature for texture as well as nontexture regions. Using local spectral histograms as features, the segmentation problem could be solved by measuring the distance among the features. However, since the local spectral histograms computed over the windows straddling boundaries do not give distinctive features, such methods tend to have difficulty in accurately localizing region boundaries. It is thus important to find a segmentation approach that can leverage local spectral histograms and accurately localize boundaries.

In this paper, we propose a new segmentation method based on local spectral histograms and linear regression. The local spectral histogram representation is introduced in Section II. By regarding a pixel location as a linear combination of representative features, the segmentation problem can be formulated as a multivariate linear regression. Section III presents our segmentation algorithm in detail. In Section IV, we show experimental results and comparisons. Section V concludes the paper.

II. LOCAL SPECTRAL HISTOGRAMS

Motivated by perceptual observations, the spectral histogram model has been proposed to characterize texture appearance [11]. For a window \mathbf{W} in an input image, a set of filter responses is computed through convolving with a chosen bank of filters $\{F^{(\alpha)}, \alpha = 1, 2, \dots, K\}$. For a sub-band image $\mathbf{W}^{(\alpha)}$, a bin of its histogram can be written as

$$H_{\mathbf{W}}^{(\alpha)}(z_1, z_2) = \sum_{\bar{\mathbf{v}} \in \mathbf{W}} \int_{z_1}^{z_2} \delta(z - \mathbf{W}^{(\alpha)}(\bar{\mathbf{v}})) dz. \quad (1)$$

Here z_1 and z_2 specify the range of the bin. $\bar{\mathbf{v}}$ represents a pixel location, and δ denotes the Dirac delta function. In this paper, we use 11 equal-width bins for each filter response. The spectral histogram with respect to the chosen filters is then defined as [11].

$$H_{\mathbf{W}} = \frac{1}{|\mathbf{W}|} (H_{\mathbf{W}}^{(1)}, H_{\mathbf{W}}^{(2)}, \dots, H_{\mathbf{W}}^{(K)}) \quad (2)$$

where $|\cdot|$ denotes cardinality. The spectral histogram is a normalized feature statistic, which can compare image windows of different sizes. For each pixel location, the local spectral histogram is computed over the window centered at the pixel. The size of the window is called integration scale. When the filters are selected properly, the spectral histogram is sufficient to capture texture appearance [11].

Manuscript received February 1, 2011. This work was supported in part by an NGA University Research Initiatives grant (HM 1582-07-1-2027).

J. Yuan and R. Li are with the Mapping and GIS Laboratory, Department of Civil and Environmental Engineering and Geodetic Science, The Ohio State University, Columbus, OH 43210 USA (e-mail: yuan.75@osu.edu; li.282@osu.edu).

D. L. Wang is with the Department of Computer Science & Engineering and the Center for Cognitive Science, The Ohio State University, Columbus, OH 43210 USA (e-mail: dwang@cse.ohio-state.edu).

In this paper, we use seven filters: the intensity filter, two LoG (Laplacian of Gaussian) filters with the scale values of 0.2 and 0.5, and four Gabor filter with the orientations of 0° , 45° , 90° , and 135° and the scale value of 3. The parameters of filters are not adjusted for individual images, but for a type of images a set of filters is chosen based on general image attributes.

In order to extract meaningful texture features, the integration scale is set to be relatively large, which makes computing local spectral histograms computationally expensive. A fast implementation method is therefore introduced in [12]. For an input image, an integral histogram image is defined as follows: at location (x, y) the integral histogram is calculated using the pixel values above and to the left of (x, y) (see Fig. 1(a)). The integral histogram image can be efficiently computed in one pass over the image. Given the integral histogram image, the histograms of arbitrary rectangular regions can be obtained with four references. As illustrated in Fig. 1(b), we can compute the histogram of region R using the following four references: $L_4 + L_1 - L_2 - L_3$. Hence, once the integral histogram image is computed, we only need three vector arithmetic operations to obtain any local spectral histogram regardless of window size. A detailed description of the fast implementation can be found in [12].

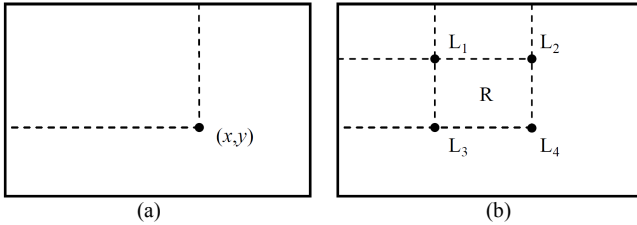


Fig. 1. Illustration of fast implementation for computing local spectral histograms. (a) The integral histogram value at location (x, y) is the histogram of the image window above and to the left of (x, y) . (b) The histogram of region R can be computed using four references: $L_4 + L_1 - L_2 - L_3$.

III. SEGMENTATION ALGORITHM

As discussed in Section I, although local spectral histograms provide an effective feature, segmentation methods using feature distance to measure region homogeneity tend to produce inaccurate boundaries caused by features extracted in image windows that cross multiple regions. Here, we describe a new segmentation algorithm based on linear regression, which can produce segmentation with high accuracy and great efficiency.

A. Segmentation using linear regression

Fig. 2(a) illustrates the difficulty of extracting features over a large window. Here, an image contains five homogenous regions. Given a pixel location A , the corresponding local spectral histogram is computed using a square window. Since this window straddles two different regions, the extracted feature is not discriminative. As a result, it is difficult to correctly classify the corresponding pixel by measuring feature similarity.

Let us define a window \mathbf{W} consisting of disjoint connected subregions $\{\mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_s\}$, and $H_{\mathbf{W}_i}^{(\alpha)}$ is the histogram

computed from region \mathbf{W}_i and filter α . Since $H_{\mathbf{W}}^{(\alpha)} = H_{\mathbf{W}_1}^{(\alpha)} + H_{\mathbf{W}_2}^{(\alpha)} + \dots + H_{\mathbf{W}_s}^{(\alpha)}$, we can rewrite the spectral histogram $H_{\mathbf{W}}$ as

$$\begin{aligned} H_{\mathbf{W}} &= \frac{1}{|\mathbf{W}|} (H_{\mathbf{W}}^{(1)}, H_{\mathbf{W}}^{(2)}, \dots, H_{\mathbf{W}}^{(K)}) \\ &= \frac{1}{|\mathbf{W}|} \left(\sum_{i=1}^s H_{\mathbf{W}_i}^{(1)}, \sum_{i=1}^s H_{\mathbf{W}_i}^{(2)}, \dots, \sum_{i=1}^s H_{\mathbf{W}_i}^{(K)} \right) \\ &= \frac{|\mathbf{W}_1|}{|\mathbf{W}|} \left(\frac{1}{|\mathbf{W}_1|} (H_{\mathbf{W}_1}^{(1)}, H_{\mathbf{W}_1}^{(2)}, \dots, H_{\mathbf{W}_1}^{(K)}) \right) \\ &\quad + \frac{|\mathbf{W}_2|}{|\mathbf{W}|} \left(\frac{1}{|\mathbf{W}_2|} (H_{\mathbf{W}_2}^{(1)}, H_{\mathbf{W}_2}^{(2)}, \dots, H_{\mathbf{W}_2}^{(K)}) \right) + \dots \\ &\quad + \frac{|\mathbf{W}_s|}{|\mathbf{W}|} \left(\frac{1}{|\mathbf{W}_s|} (H_{\mathbf{W}_s}^{(1)}, H_{\mathbf{W}_s}^{(2)}, \dots, H_{\mathbf{W}_s}^{(K)}) \right). \end{aligned}$$

With the definition in (2), we have

$$H_{\mathbf{W}} = \frac{|\mathbf{W}_1|}{|\mathbf{W}|} H_{\mathbf{W}_1} + \frac{|\mathbf{W}_2|}{|\mathbf{W}|} H_{\mathbf{W}_2} + \dots + \frac{|\mathbf{W}_s|}{|\mathbf{W}|} H_{\mathbf{W}_s}. \quad (3)$$

Therefore, a spectral histogram of an image window can be linearly decomposed into spectral histograms of its subregions, where weights are proportional to region areas.

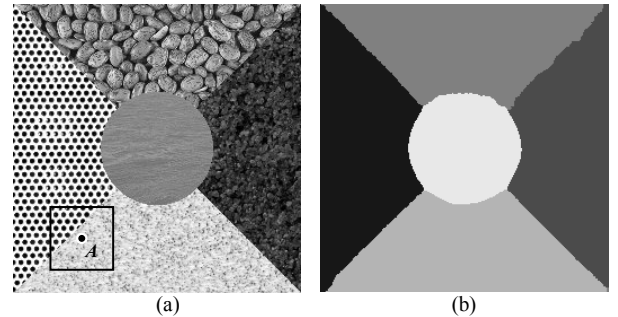


Fig. 2. Texture image segmentation via linear regression. (a) Texture image with size 256×256 . The local spectral histogram at location A is computed within the square window. (b) Segmentation result using linear regression. Each region is represented by a distinct gray value

Because spectral histograms can characterize image appearance, we assume that spectral histograms within a homogeneous region are approximately constant. This assumption along with (3) implies that the spectral histogram of a local window can be approximated by a weighted sum of the spectral histograms of regions overlapping that window, and the corresponding pixel can be classified into the region whose spectral histogram weighs the most. For instance, in Fig. 2(a), the local spectral histogram at location A can be decomposed into the spectral histograms of two neighboring regions, and A can be segmented according to the weights.

The above analysis is not valid in some scenarios: 1) subregions within an image window are too small to obtain meaningful histograms; and 2) filter scales are so large as to

cause distorted histograms in near-boundary subregions, making them different from the constant spectral histograms of the regions that include the subregions. However, both scenarios should have a minimal impact for the following reasons. For the first scenario, if a subregion within a window is small, it contributes little to the spectral histogram and thus can be neglected in feature decomposition. While it is possible that a window consists of many small subregions, such scenarios do not occur often in an image. For the second scenario, because the purpose of filtering in a spectral histogram is to capture a local spatial pattern, the chosen filters should not have large scales.

For each homogeneous region in an image, we define a representative feature, which should be equal to the constant spectral histogram of the region. By extending the above analysis, we regard the local spectral histogram of a pixel location as a linear combination of all representative features weighted by the area coverage of each feature. Consequently, we can use a multivariate linear regression model to associate each feature to the representative features. Given an image with N pixels, feature dimensionality of M , and L representative features, the model can be expressed as

$$\mathbf{Y} = \mathbf{Z}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \quad (4)$$

where \mathbf{Y} is an $M \times N$ matrix whose columns are feature vectors of all the pixel locations, \mathbf{Z} is an $M \times L$ matrix whose columns are representative features, and $\boldsymbol{\beta}$ is an $L \times N$ matrix whose columns are weight vectors. $\boldsymbol{\varepsilon}$ is an error term.

Since the feature matrix \mathbf{Y} and the representative feature set \mathbf{Z} are known, the segmentation problem boils down to estimating $\boldsymbol{\beta}$ that best models the relationship between the feature matrix and the representative features. The least squares estimate of $\boldsymbol{\beta}$ is given by [6]

$$\hat{\boldsymbol{\beta}} = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{Y}. \quad (5)$$

One can manually select windows at the centers of homogeneous regions to calculate representative features, and a segmentation result is provided by examining $\hat{\boldsymbol{\beta}}$ – each pixel is assigned to the class where the corresponding representative feature has the largest weight. Thus, the segmentation is completed through simple matrix operations. Fig. 2(b) shows the segmentation result of the image in Fig. 2(a), where different segments are indicated using distinct gray values. By counting the wrongly segmented pixels, the segmentation error rate in this case is only 1.1%.

Certain constraints should be imposed on least squares solutions. For example, the sum of the weight vector of each pixel should be equal to one, and the weight vector should be nonnegative. There are solutions for least squares estimates satisfying these constraints [8]. However, for the sake of simplicity and computational efficiency, we take the unconstrained solution in (5), which gives satisfactory results in our experiments.

B. Automatic representative feature selection

In the segmentation algorithm presented above, representative features are assumed to be manually given. We present an algorithm to automatically select representative features corresponding to homogeneous regions. The goal is to find such features that each of them is close to a subset of all the features and they are as different as possible from each other. A straightforward method is to perform k -means clustering on all the features and take cluster centers as representative features. However, the features near boundaries can form small clusters, which make the clustering result unstable. To address this problem, we compute image gradients based on local spectral histograms and exclude the features associated with large gradients from the clustering process. Because the remaining features are expected to be inside homogenous regions, they should yield a stable clustering result.

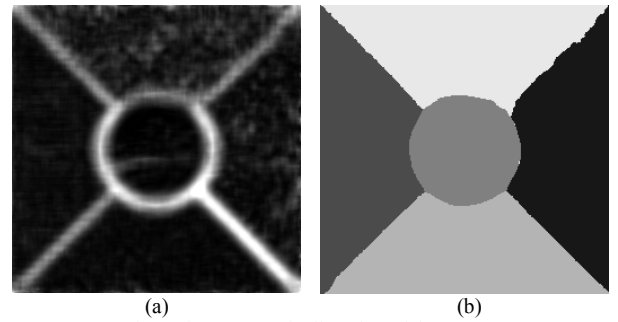


Fig. 3. Segmentation using automatically selected features. (a) Texture gradient of the image shown in Fig. 2(a). (b) Segmentation result with the representative features selected automatically.

At a pixel location (x, y) , we compute the feature distance between pixel locations $(x + d, y)$ and $(x - d, y)$ and that between $(x, y + d)$ and $(x, y - d)$, where d is half of side length of \mathbf{W} . The gradient is the sum of the two distances [11]. We employ χ^2 -statistics to measure the distance, which is commonly used with histograms,

$$\chi^2(H_{w_1}, H_{w_2}) = \frac{1}{|\mathbf{W}|} \sum_{\alpha=1}^K \sum_z \frac{(H_{w_1}^{(\alpha)}(z) - H_{w_2}^{(\alpha)}(z))^2}{H_{w_1}^{(\alpha)}(z) + H_{w_2}^{(\alpha)}(z)}. \quad (6)$$

This distance measure is also used in k -means clustering. It should be noted that the particular form of distance measure is not critical for spectral histograms. Other distance measures can also achieve comparable results.

Fig. 3(a) shows the gradient of the image in Fig. 2(a), where the gray value is proportional to the gradient. One can see that the pixels near boundaries have large gradients. By applying k -means clustering to the features with small gradients, we obtain cluster centers as the representative features. Fig. 3(b) shows the segmentation result using the resulting representative features, which is similar to the result in Fig. 2(b). Generally speaking, because automatically selected features pool over populations of features, they can be more representative and produce better segmentation results than manually selected features.

C. Segment number determination

A thorny issue in k -means clustering is how to determine the cluster number, i.e. the segment number in our case. Note that our method puts pixels with similar features into one segment without considering the connectivity of each segment, which, however, can be achieved via postprocessing. Here, we present a method to determine the segment number.

For a unique solution in (3) to exist, it requires that \mathbf{Z} have the full column rank. In other words, representative features have to be linearly independent in order to have a unique segmentation solution. Since each feature is considered to be a linear combination of representative features, the rank of the feature matrix \mathbf{Y} should equal the rank of \mathbf{Z} , which is exactly the number of representative features, or the segment number. Although in real cases the rank of \mathbf{Y} tends to be larger than the segment number due to image noise, we can estimate the segment number by determining the effective rank of \mathbf{Y} .

We employ Singular Value Decomposition (SVD) to determine the effective rank of a matrix [10]. The feature matrix can be decomposed into the following form

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

$$= \begin{pmatrix} u_{11} & \cdots & u_{1M} \\ \vdots & \ddots & \vdots \\ u_{M1} & \cdots & u_{MM} \end{pmatrix} \begin{pmatrix} \sigma_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{MN} \end{pmatrix} \begin{pmatrix} v_{11} & \cdots & v_{1N} \\ \vdots & \ddots & \vdots \\ v_{N1} & \cdots & v_{NN} \end{pmatrix}^T. \quad (7)$$

Here \mathbf{U} and \mathbf{V} are orthogonal matrices, where the columns of \mathbf{U} are the eigenvectors of the matrix $\mathbf{Y}\mathbf{Y}^T$, and the columns of \mathbf{V} are the eigenvectors of the matrix $\mathbf{Y}^T\mathbf{Y}$. The diagonal terms of the matrix $\mathbf{\Sigma}$ are called singular values, which are the square roots of the eigenvalues of the matrix $\mathbf{Y}\mathbf{Y}^T$, or $\mathbf{Y}^T\mathbf{Y}$. The effective rank can be determined as follows. First rank the singular values, and then discard those below a certain threshold. The number of remaining singular values is the effective rank.

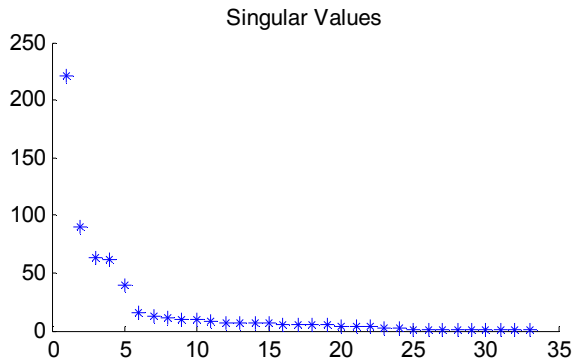


Fig. 4. Singular values computed from the feature matrix of the image in Fig. 2(a).

In Fig. 4, we plot the singular values for the image in Fig. 2(a). The singular values are arranged in a descending order. We can see that the first 5 values are noticeably larger than

the rest, which agrees with the number of regions in the original image.

A few comments should be made regarding this method. Based on the fact that $\mathbf{Y}\mathbf{Y}^T$ and $\mathbf{Y}^T\mathbf{Y}$ have the same nonzero eigenvalues, we choose $\mathbf{Y}\mathbf{Y}^T$ to compute the singular values, which is a much smaller matrix than $\mathbf{Y}^T\mathbf{Y}$. When the singular values are computed, we need to set a threshold to determine the effective rank. Although in some cases the singular values are not well separated, our experiments show that a proper threshold can work well for one image type.

D. Smoothing effect

The proposed method has a smoothing effect on region boundaries. To illustrate such an effect, we show an image containing jagged boundaries in Fig. 5, where a square window is used to compute the local feature. According to the coverage of two regions within the window, the proposed method will segment the corresponding pixel into the bright region. With the integration scale sufficiently large, we would obtain a segmentation result with a straight boundary. Although the smoothing effect may cause deviation from the true boundary as illustrated in Fig. 5, it is interesting to note that the effect tends to reduce the total curvature of boundaries. In other words, the smoothing effect amounts to a form of regularization often formulated as an objective for image segmentation [1] [13]. This smoothing effect is apparent in our experiments with real images in Section IV.

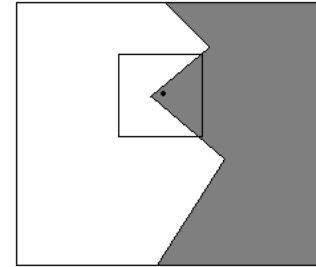


Fig. 5. Illustration of smoothing effect

Smoothness is controlled by integration scales. Ideally, one should choose an integration scale that exhibits some boundary smoothing but does not lead to over-smoothing. Different techniques could be considered, e.g., decomposition stability [4], inter-cluster versus intra-cluster variability [7], or multiscale analysis. For instance, at each location in multiscale analysis one can compute spectral histograms at multiple integration scales. By analyzing feature variations and relationships across scales, an optimal local integration scale or a combination of multiple scales could be identified. Such analysis will be addressed in future work.

IV. EXPERIMENTAL RESULTS AND COMPARISONS

A. Experimental results

We have applied our algorithm to different types of images. Fig. 6 shows the segmentation results for a set of texture

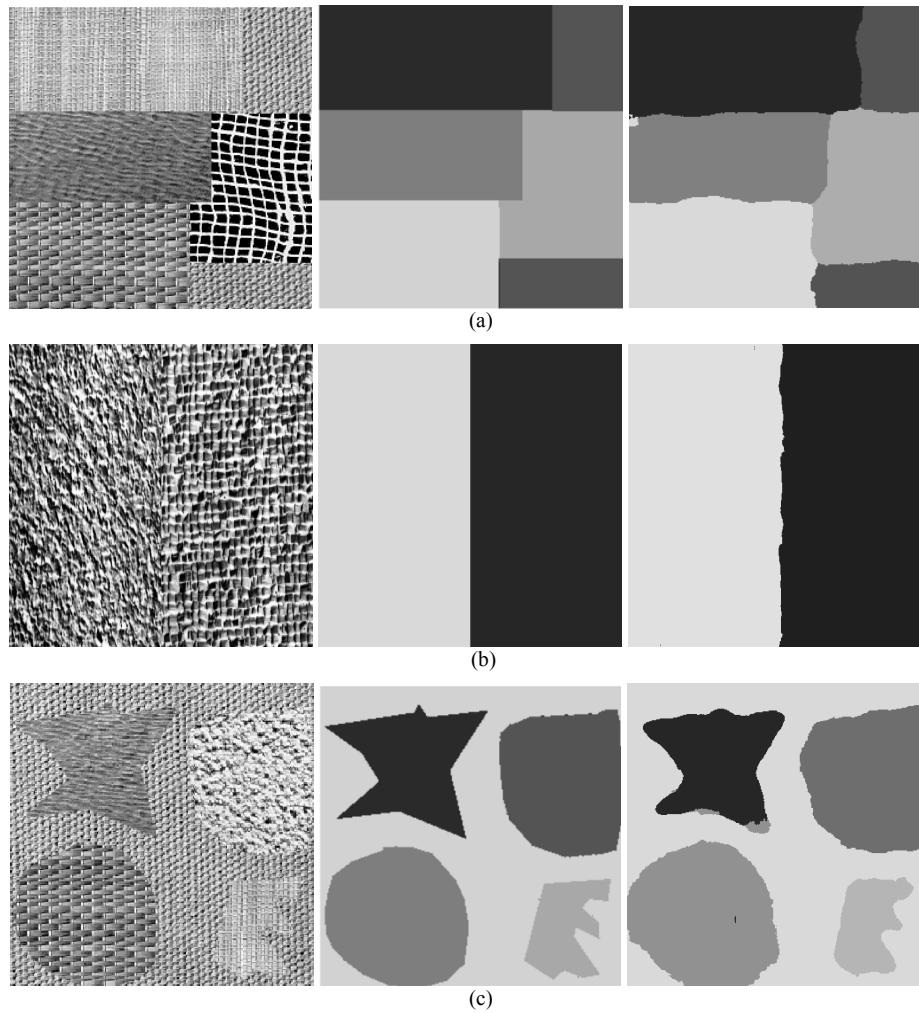


Fig. 6. Texture image segmentation. In each row, the left is the original image, the middle is the ground true, and the right is the segmentation result using our method. The integration scale used here is 25×25 .

images, including an image consisting of regions with different sizes, an image containing very similar textures, and an image with irregular-shaped regions. We use all seven filters to calculate local spectral histograms. The left column shows the original images, the middle column ground truth segmentation, and the right column our segmentation results. As we can see, the regions with different textures are separated successfully, and the boundaries are localized well due to feature decomposition. The inaccuracy of the boundaries is mostly caused by similar texture appearances.

Next, we show the results of applying our method to natural images. In general, orientation is not a strong cue for segmenting natural images. Hence, we only use the intensity filters and two LoG filters with different scales. Fig. 7 presents our segmentation results on two images, which are frequently used in image segmentation literature. It can be seen that for both images our algorithm produces reasonable segmentation. In Fig. 7(a), the resulting segment by and large agrees with the true shape of the zebra. In Fig. 7(b), the main components of the image, including the eyes, the skin, and the shirt, are successfully segmented. Without involving any object-specific models or human intervention, our results are comparable with the best results available [9] [14].

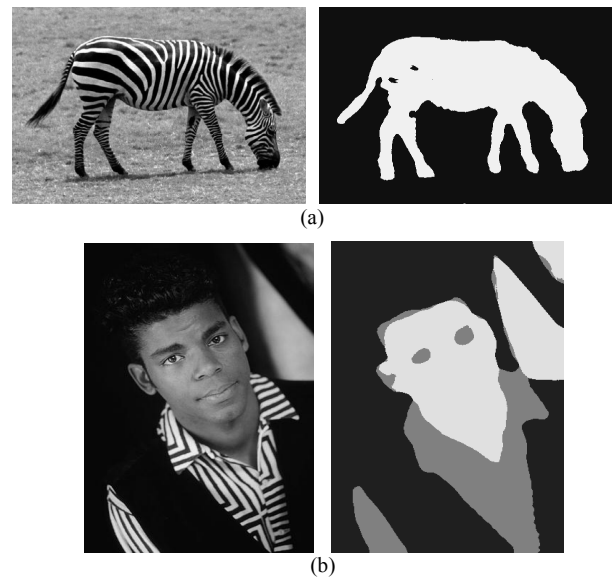


Fig. 7. Natural image segmentation. In each row, the left is the original image, and the right is the segmentation result using our method. The integration scale used here is 15×15 .

B. Systematic Evaluation and Comparisons

We compare the proposed method with another spectral histogram based segmentation algorithm proposed by Liu and Wang [12]. They build probability models based on local spectral histograms and iteratively update segmentation using the model, and the final result is obtained by a separate boundary localization algorithm. We refer to this method as the LW method. Fig. 8 shows the comparison results on two natural images. To show the accuracy of segmentation results, the boundaries of the salient objects are embedded in the original images. Both methods successfully segment the objects with well localized boundaries. A closer comparison, however, indicates that our results have much smoother boundaries, which should be more desirable in practical uses.

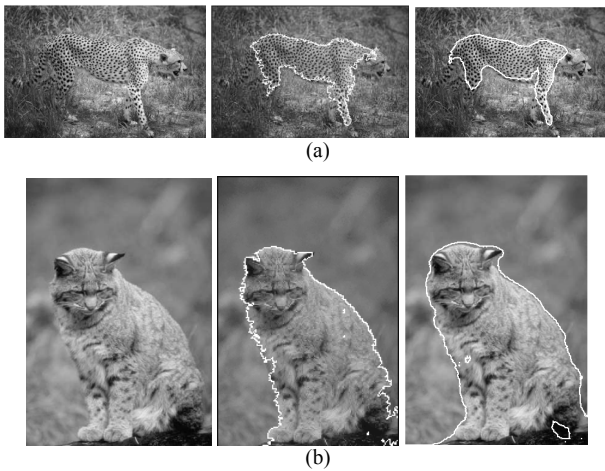


Fig. 8. Comparison between the LW method and proposed method on two natural images. In each panel, the left is the original image, the middle is the segmentation results from the LW method, and the right is the segmentation results from our method. The region boundaries are marked in white.

To provide a quantitative assessment, we apply the LW and proposed method to the images shown in Fig. 6 and calculate the segmentation error rate by comparing with the ground truth. Since in this case a segmentation result is mostly consistent with the ground truth, we label each segment in the ground truth and assign the same label to the corresponding segment in the segmentation result. The error rate is the percentage of pixels that have different labels between the two segmentation results. The evaluation results are summarized in Table I. The comparison shows that our method produces significantly smaller segmentation errors in all three cases.

TABLE I
SEGMENTATION ERROR RATE (%) FOR THE LW AND THE PROPOSED METHODS

Image	LW method	Proposed method
Fig. 6(a)	5.8	2.5
Fig. 6(b)	2.7	1.9
Fig. 6(c)	3.7	2.8

V. CONCLUSION

In this paper, we have developed a novel segmentation method for images consisting of texture and nontexture regions. We use local spectral histograms as features. Based on the observation that each feature can be approximated by linearly combining several representative features, we formulate the segmentation problem as a multivariate linear regression, where the solution is given by least squares estimation. We have also proposed algorithms to automatically select representative features and determine their numbers. Experiments show that our method gives more accurate segmentation than other methods.

Although in the current version we assume that filterbanks and integration scales are given for a type of images, fully autonomous segmentation requires techniques for filter and integration scale selection that are task-dependent. Adaptive filters and integration scales may be adopted, which should be suited to the spatial structure of an image. We are currently investigating these issues along with a multiscale treatment of the smoothing effect.

REFERENCES

- [1] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, pp. 61–79, 1997.
- [2] T. Cour, F. Benezit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
- [3] D. Dunn, W. E. Higgins, "Optimal Gabor filters for texture segmentation," *IEEE Transactions on Image Processing*, vol. 4, pp. 947–964, 1995.
- [4] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. Academic Press, 1990.
- [5] A. Jain and R. Farrokhia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 24, pp. 1167–1186, 1991.
- [6] R. A. Johnson, D. W. Wichern, *Applied Multivariate Statistical Analysis*, 6th ed. Pearson Prentice Hall, 2007.
- [7] L. Kauffman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. J. Wiley & Sons, 1990.
- [8] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice Hall, 1993.
- [9] L. Kokkinos, G. Evangelopoulos, and P. Maragos, "Texture analysis and segmentation using modulation features, generative models, and weighted curve evolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 142–157, 2009.
- [10] K. Konstantinides and K. Yao, "Statistical analysis of effective singular values in matrix rank determination," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 757–763, 1988.
- [11] X. Liu and D. L. Wang, "A spectral histogram model for texton modeling and texture discrimination," *Vision Research*, vol. 42, pp. 2617–2637, 2002.
- [12] X. Liu and D. L. Wang, "Image and texture segmentation using local spectral histograms," *IEEE Transactions on Image Processing*, vol. 15, pp. 3066–3077, 2006.
- [13] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 21, pp. 577–685, 1989.
- [14] O. Michailovich, Y. Rathi, and A. Tannenbaum, "Image segmentation using active contours driven by the Bhattacharyya gradient flow," *IEEE Transactions on Image Processing*, vol. 16, pp. 2787–2801, 2007.
- [15] N. Paragios and R. Deriche, "Geodesic active regions and level set methods for supervised texture segmentation," *International Journal of Computer Vision*, vol. 46, pp. 223–247, 2002.
- [16] D. L. Wang, "The time dimension for scene analysis," *IEEE Transactions on Neural Network*, vol. 16, pp. 1401–1426, 2005.

- [17] D. L. Wang and D. Terman, "Image segmentation based on oscillatory correlation," *Neural Computation*, vol. 9, pp. 805–836, 1997.
- [18] L. Zhang and Q. Ji, "Image segmentation with a unified graphical model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1406–1425, 2010.