IMAGE SEGMENTATION USING LOCAL SPECTRAL HISTOGRAMS

Xiuwen Liu
Dept. of Computer Science
Florida State University
Tallahassee, FL 32306-4530
liux@cs.fsu.edu

DeLiang L. Wang
Dept. of Comp. & Info. Science
The Ohio State University
Columbus, OH 43210-1277
dwang@cis.ohio-state.edu

Anuj Srivastava
Dept. of Statistics
Florida State Univeristy
Tallahasse, FL 32306
anuj@stat.fsu.edu

ABSTRACT

We propose a new algorithm for image segmentation. We use spectral histogram, which is a vector consisting of marginal distributions of responses from chosen filters as a generic feature for texture as well as intensity images. Motivated by a new segmentation energy functional, we derive an iterative and deterministic approximation algorithm for segmentation. Based on the relationships between different scales and neighboring windows, we also develop an algorithm which can automatically detect homogeneous regions in an input image, which may consist of texture regions. To reduce the boundary uncertainty due to the large spatial window used for spectral histograms, we propose a novel local feature by building precise probability models based on current segmentation results. We have applied our algorithm to intensity, texture, and natural images and obtained good results with accurate texture boundaries.

1. INTRODUCTION

Segmentation can be defined as a constrained partition problem. Each partitioned region should be as homogeneous as possible and neighboring regions should be as different as possible. To define a perceptually meaningful homogeneity measure, we use a local spectral histogram, defined as a vector of marginal distributions of responses of chosen filters as a general feature statistic for intensity as well as texture images. The spectral histogram has been successfully used to capture the appearance of texture as well as intensity images [2],[7],[3]. We then derive an efficient algorithm by extending Mumford-Shah energy functional for segmentation [4]. Because the solution space for segmentation must be piece-wise constant by the definition, the Mumford-Shah energy functional becomes [4]:

$$E(\Gamma) = \sum_i \int\int_{R_i} (g(x,y) - \mu_{R_i}(g))^2 dxdy + \nu|\Gamma| \quad (1)$$

where

$$\mu_{R_i}(g) = \frac{1}{|R_i|}\int\int_{R_i} g(x,y)dxdy$$

Here $|R_i|$ is the area of region R_i. There are several limitations of the energy functional (1). The feature to be used is limited to the mean value of a region, which is not sufficient for characterizing texture regions and also may give undesirable solutions when the mean values of two regions are very close. Another problem is that a result obtained by minimizing the energy functional is unpredictable in regions which cannot be described by their mean values. Some of problems are resolved in [6].

In this paper, we extend the model (1) using the spectral histogram and the associated distance measure. We develop an algorithm which couples the feature detection and segmentation steps together by extracting features based on the currently available segmentation result. We also develop an algorithm which identifies regional features in homogeneous texture regions automatically and one for boundary localization.

2. SEGMENTATION ALGORITHM

Following the notations used in Mumford and Shah [4], let R be a grid defined on a planar domain and R_i, $i = 1, \cdots, n$ be a disjoint subset of R, Γ_i be the piecewise smooth boundary of R_i, and Γ be the union of Γ_i, $i = 1, \cdots, n$. A feature $\mathcal{F}_i$ is associated with each region R_i, $i = 1, \cdots, n$. We also define R_0, which is called *background* [5], as $R_0 = R - (R_1 \cup \cdots R_n)$.

Based on the energy functional by Mumford and Shah [4], given an input image I, we define an energy functional for segmentation as

$$\begin{aligned} E(R_i, n) \;=\; & \lambda_R \textstyle\sum_{i=1}^n \sum_{(x,y)\in R_i} D(\mathcal{F}_{R_i}(x,y), \mathcal{F}_i) - \\ & \lambda_{\mathcal{F}} \textstyle\sum_{i=1}^n \sum_{j=1}^n D(\mathcal{F}_i, \mathcal{F}_j) + \\ & \lambda_{\mathcal{L}} \textstyle\sum_{i=1}^n |\Gamma_i| - \sum_{i=1}^n |R_i| \end{aligned} \quad (2)$$

0-7803-6725-1/01/$10.00 ©2001 IEEE

Here D is a distance measure between a feature at a pixel location and the feature vector of the region, λ_R, $\lambda_{\mathcal{F}}$, and $\lambda_{\mathcal{L}}$ are weights that control the relative contributions of the corresponding terms.

The functional given in (2) extends a special case of the functional by Mumford and Shah [4], as shown in (1). In (2), the first term encodes the homogeneity requirement in each region R_i and the second term requires that the features of the regions should be as different as possible. The third term requires that boundaries of regions should be as short as possible, or as smooth as possible. The last term is motivated by the fact that some regions may not be described well by the selected features and should be treated as background, which can be viewed as grouping through inhomogeneity.

We use spectral histograms as features vectors [3], i.e., $\mathcal{F}_{R_i}(x,y) = H_{W^{(s)}_{(x,y)}}$, where $W^{(s)}_{(x,y)}$ is a local neighborhood, the size and shape of which are given by integration scale $W^{(s)}$ for segmentation. $W^{(s)}$ is a predefined neighborhood. A local spectral histogram is defined as the marginal distributions of filter responses. The size of the input image window is called *integration scale.* Because marginal distribution of each filter response is a distribution, a similarity measure is defined as χ^2-statistic.

Given the energy functional defined in (2) and local spectral histogram model, we use an iterative but deterministic algorithm. We assume that the feature vectors for regions, which may be given manually or detected automatically, are close to the true region vectors. For a given pixel (x, y) to be updated, we first estimate the spectral histogram using a window around the pixel and the local updating rule is given by

$$\begin{aligned}\pi_i(x,y) \;=\; & (1-\lambda_\Gamma)P(\chi^2(H_{W^{(s)}_{(x,y)}}, H_i))+ \\ & \lambda_\Gamma \textstyle\sum_{(x_1,y_1)\in N(x,y)} \frac{L^i_{N(x_1,y_1)}(x_1,y_1)}{|N(x,y)|}.\end{aligned} \qquad (3)$$

Here $L^i_{N(x,y)}(x,y)$ is the number of pixels in $N(x,y)$ whose current labels are i and $N(x,y)$ is a user-defined neighborhood, and λ_Γ is a parameter that controls the relative contributions from the region and boundary terms. The new label of (x, y) is assigned as the one that gives the maximum $\pi_i(x,y)$. A special case of (3) is for pixels along boundaries between the background region and a given region because we do not assume any model for the background region. For pixel $(x,y) \in R_0$, which is adjacent to region R_i, $i \neq 0$, if

$$\chi^2(H_{W^{(s)}_{(x,y)}}, H_i) < \lambda_B * T_i,$$

we assign label i to (x, y). Here T_i is a threshold for region R_i, which is determined automatically and λ_B is a parameter which determines relative penalty for unsegmented pixels. To etimate the probality model and parameter T_i, first feature vectors $\mathcal{F}_i$ are extracted from windows centered at given or detected pixel locations, the size of which is specified by integration scale $W^{(s)}$. Then the image is classified using feature vectors $\mathcal{F}_i$, and the classification result is used as the initial segmentation. We compute the histograms of the χ^2-statistic between the computed and given spectral histograms. Parameter T_i is determined by the first trough after the first peak from its histogram. Based on the assumption that feature vectors $\mathcal{F}_i$ are close to the true feature vectors, we derive a probability model by assigning zero probability for values larger than T_i. Then the initial segmentation result is refined through an iterative procedure similar to region growing but with fixed region features. This results in a fast convergence speed. Because spectral histograms characterize texture properties well, we obtain good experimental results even with this simple procedure. Fig. 1(a) shows an image consisting of two images with similar mean values but different variances. Fig. 1(b) shows the initial classification result and the segmentation result is shown in Fig. 1(c), where all the pixels are segmented correctly.

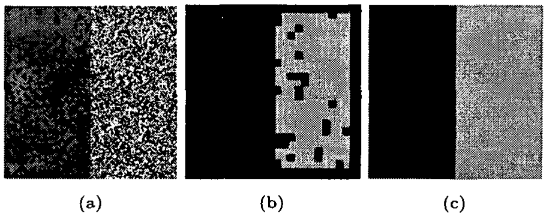


Fig. 1. Gray-level image segmentation using spectral histograms. $W^{(s)}$ is 15×15, $\lambda_\Gamma = 0.2$, and $\lambda_B = 3$. Two features are given at (32, 64) and (96, 64). (a) A synthetic image with size 128×128. The image is generated by adding zero-mean Gaussian noise with different σ's at left and right regions. (b) Initial classification result. (c) Segmentation result. Each region is represented by a manually assigned grey value. All the pixels are perfectly segmented.

3. AUTOMATED SEED SELECTION

In this section, we attempt to develop a solution for identifying seed points automatically based on the spectral histogram. The basic idea of the proposed method is to identify homogeneous texture regions within a given image. The spectral histogram can be defined on image patches with different sizes and shapes and those spectral histograms defined on different patches can be compared using the similarity/dissimilarity measure as spectral histograms are naturally normalized.

We identify homogeneous texture regions based on the divergence between two integration scales. Let $W^{(a)}$ be an integration scale larger than $W^{(s)}$, the integration scale for segmentation. We define the distance between the two scales centered at pixel (x, y)

$$\psi^{(s,a)}(x,y) = D(H_{W^{(s)}(x,y)}, H_{W^{(a)}(x,y)}). \qquad (4)$$

Within a homogeneously defined texture region, $\psi^{(s,a)}$ should be small because $H^{W^{(s)}}(x,y)$ and $H^{W^{(a)}}(x,y)$ should be similar. We also define a distance measure between different windows at scale $W^{(s)}$ within the window given by $W^{(a)}$,

$$\psi^{(s,s)}(x,y) = \max_{(x_1,y_1)} D(H_{W^{(s)}(x,y)}, H_{W^{(s)}(x_1,y_1)}). \qquad (5)$$

Equation (5) is approximated in implementation using four corner windows within $W^{(a)}$. Finally, we want to choose features that are as different as possible from those already chosen. Suppose we choose n features already, where, $\mathcal{F}_i = H_{W^{(s)}(x,y)}$, for $i = 1, \ldots, n$, we define

$$\psi^{(c)}(x,y) = \max_{1 \leq i \leq n} D(H_{W^{(s)}(x,y)}, \mathcal{F}_i). \qquad (6)$$

We have the following saliency measure

$$\begin{aligned} \psi(x,y) = {} & (1 - \lambda_C)(\lambda_A \times \psi^{(s,a)}(x,y) + \\ & (1 - \lambda_A) \times \psi^{(s,s)}(x,y)) - \\ & \lambda_C \times \psi^{(c)}(x,y) \end{aligned} \qquad (7)$$

Here λ_A and λ_C are parameters to determine the relative contribution of each term.

To save computation, we compute $\psi(x, y)$ on a coarser grid. Feature vectors are chosen according to the value of $\psi(x, y)$ until

$$\lambda_A \times \psi^{(s,a)}(x,y) + (1 - \lambda_A) \times \psi^{(s,s)}(x,y) < T_A,$$

where T_A is a threshold.

4. EXPERIMENTAL RESULTS

Fig. 2 shows the segmentation results for texture images. First the feature vectors are identified automatically and initial result is then obtained using a minimum distance classifier using the found region features. As shown in these examples, the texture boundaries are localized well and all the homogeneous texture regions are identified by the seed selection algorithm.

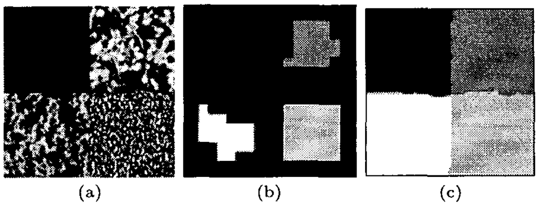


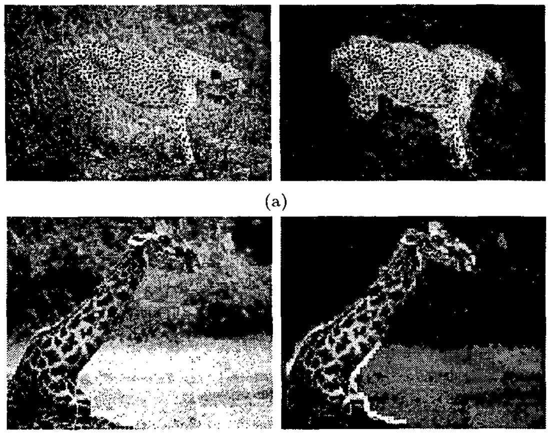
Fig. 2. Texture image segmentation. $W^{(s)}$ is 29 × 29, $W^{(a)}$ is 43 × 43, $\lambda_C = 0.1$, $\lambda_A = 0.2$, $\lambda_B = 5.0$, $\lambda_\Gamma = 0.4$, and $T_A = 0.20$. (a) Input texture image. Here the representative pixels are detected automatically. (c) Segmentation result.

Natural images in general consist of many regions that are not homogeneous texture regions and we are interested in some meaningful regions, called *region of interest*. This is achieved in our system by identifying only a few region features. We apply the same algorithm but with one region identified for the two examples shown in Fig. 3. Fig. 3(a) shows a cheetah image and the segmentation result is shown in Fig. 3(b). To show the accuracy of segmented boundary, the result is embedded into the original image by reducing the intensity of unsegmented regions. Fig. 3(c) shows a natural image of a giraffe and Fig. 3(d) shows the segmentation result. Given that our system is generic and there is no image specific training and filter selection, our results are comparable with the best available results.

5. LOCALIZATION OF TEXTURE BOUNDARIES

Because textures need to be characterized by spatial relationships among pixels, relatively large integration windows are needed in order to extract meaningful features. The large integration scale we use however results in large errors along texture boundaries due to the uncertainty introduced by large windows [1]. For arbitrary texture boundaries, the errors along boundaries can be large even when the overall segmentation performance is good. For example, Fig. 4(b) shows a segmentation result using spectral histograms. While the segmentation error is only 6.55%, visually the segmentation result is intolerable due to large errors along texture boundaries.

In order to reduce the uncertainties along boundaries, we first build a probability model for given m pixels from a texture region. To capture the spatial relationship, we choose for each texture region a window as a template. In our case, the template is the same window from which the region feature $\mathcal{F}$ is extracted. For the selected m pixels, we define the distance between those pixels and a texture region as the mini-

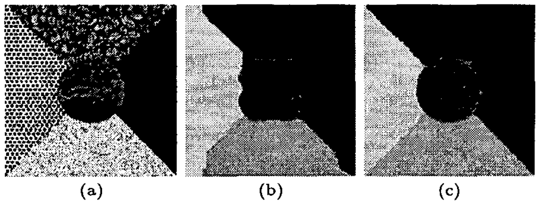

(a)

(b)

Fig. 3. (a) A cheetah image with size 324 × 486. (b) The segmentation result for (a). (c) A giraffe image with size 300 × 240. (d) The segmentation result for (c). To demonstrate the accuracy of the results, the segmentation results are embedded into the original image by lowering the intensity values of the background region.

mum mean square distance between those pixels and the template. Based on the obtained result, we build a probability model for each texture region with respect to the proposed distance measure. Intuitively, if the m pixels belong to a texture region, it should match the spatial relationship among pixels when the m pixels are aligned with the texture structure.

After the probability model is derived, we use the local updating equation given in (3) by replacing $W^{(s)}_{(x,y)}$ by the m pixels in a texture region along its boundary and $\chi^2(H_{W^{(s)}_{(x,y)}}, H_i)$ by the new distance measure. Fig. 4(c) shows the refined segmentation result with $m = 11$ pixels. The segmentation error is reduced to 0.95% and visually the segmentation result is improved significantly.


(a)

(b)

(c)

Fig. 4. (a) A texture image with size 256×256. (b) The segmentation result using spectral histograms. The segmentation error is 6.55%. (c) Refined segmentation result. The segmentation error is 0.95%.

6. CONCLUSION

We have proposed a novel algorithm for segmenting natural images. By using local spectral histograms, our algorithm can handle texture as well as non-texture images in a unified way. One distinctive advantage of our approach is that our algorithm also provides an explicit feature for regions which can be used for classification and recognition. Our boundary location algorithm improves the boundary accuracy significantly.

7. REFERENCES

[1] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), pp. 679–698, 1986.

[2] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis," In *Proceedings of SIGGRAPHS*, pp. 229–238, 1995.

[3] X. Liu, Computational Investigation of Feature Extraction and Image Organization, Ph.D. Dissertation, The Ohio State University, Columbus, Ohio,USA, 1999 (available at http://www.cis.ohio-state.edu/~liux).

[4] D. Mumford and J. Shah, "Optimal approximations of piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, XLII(4), pp. 577–685, 1989.

[5] D. L. Wang and D. Terman, "Image segmentation based on oscillatory correlation," *Neural Computation*, 9, pp. 805–836, 1997.

[6] 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*, 18, pp. 884–900, 1996.

[7] S. C. Zhu, X. Liu, and Y. Wu, "Exploring Julesz Ensembles by Efficient Markov Chain Monte Carlo," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 22, pp. 554-569, 2000.