NATURAL SCENE SEGMENTATION BASED ON A STOCHASTIC TEXTURE REGION MERGING APPROACH

R. S. Medeiros, J. Scharcanski

Universidade Federal do Rio Grande do Sul
Instituto de Informática
Caixa Postal 15064, Porto Alegre - RS, Brazil

A. Wong

University of Waterloo
Department of Systems Design Engineering
Waterloo, Canada N2L 3G1

ABSTRACT

This paper presents an approach for segmenting natural scenes based on the underlying texture characteristics using a stochastic region merging strategy. Texture region models are constructed from patch-based stochastic texture features using a texton dictionary learning approach. Finally, a stochastic region merging strategy performs the image segmentation based on texture region likelihood. Compared with other state-of-the-art texture segmentation methods, our experimental results suggest that our approach potentially can handle better highly textured regions commonly found in natural scenes, and also can be more robust to color and illumination variations.

***Index Terms*—** Image segmentation, texture segmentation, stochastic region merging, natural scenes.

1. INTRODUCTION

Image segmentation is specially challenging in natural scenes, where we may find a large color and illumination variability, and grouping similar textures can be very subjective. Therefore, many methods have been proposed for texture representation, such as wavelet and Gabor filtering[1, 2], image patches [3], just to name a few. Other approaches use texture features based on brightness, color or texture gradients [4, 5]. More recently, stochastic texture features have been proposed [6], which are extracted from image patches using random projections. This allows a significant reduction in feature dimensionality, while preserving texture information and discrimination capacity between different textures. An alternative approach involves the use of improved region merging strategies instead of improved texture features. These approaches iteratively merge adjacent regions (initialized with single pixels) after a deterministic [7] or stochastic [8] criteria, and have been shown to achieve strong results, even using simple features such as intensity or color.

This paper proposes to combine stochastic texture features with the stochastic region merging approach proposed in [8], and evaluate its viability for natural scene segmentation. The proposed approach handles highly textured regions based on stochastic texture models, and has the potential to not only better account for certainties in texture characteristics within a region, but also color and illumination variations across the image. Next, we present the stochastic texture features used to build the proposed texture model. Then we present the stochastic region merge strategy used to segment the image, based on the proposed texture similarity measurement.

2. REPRESENTING TEXTURES USING STOCHASTIC PATCH FEATURES AND MODELS

Before beginning the texture feature extraction, we map all color in the input image to the CIE L*a*b* color space, reducing correlation between the luminance and color channels. To obtain a robust local texture description, rotation-invariant stochastic texture representations (STR), based on image patches [3] and random projections (RP) [9], are computed at each channel independently. Here, let us define a vector of stochastic texture features (denoted as $v_c(p)$) within the vicinity of pixel p at channel c as:

$$v_c(p) = \Phi sort(\mathcal{N}_c(p)), \tag{1}$$

where $\Phi \in \mathbb{R}^{m \times n}$, $m < n$ denotes a projection matrix, $\mathcal{N}_c(p)$ denotes the neighborhood (image patch) around pixel p at channel c, and $sort$ denotes a sorting operation used to introduce rotational invariance since the sorted results are independent of the original positioning of elements.

Ideally, this projection matrix Φ is an information preserving transform, which can be ensured [9] if the projection matrix $\Phi = \{r_{ij}\}$ is defined in a random, stochastic manner at each pixel:

$$r_{ij} = \begin{cases} 0 \text{ with probability } \frac{1}{2}; \\ 1 \text{ with probability } \frac{1}{2}. \end{cases}$$

A texture is assumed to be a pattern, stochastic or periodic, that is repeated over some area. Although STR represents pixel-wise texture information, we need texture region representations for regions which may have many pixels. The textural information in a region is modeled by assembling the texture feature vectors in a texton dictionary [3, 6], which contains texture feature vector prototypes.

In this work, the set of all feature vectors at a particular channel c are used to learn a texton dictionary via k-means clustering. As such, each texton in the texton dictionary is a cluster centroid. This dictionary may have different number of textons per texture class, but we have determined experimentally that with a sufficient number of clusters the dictionary is able to reliably represent all textures in the image. Once the dictionary has been obtained, a texture appearance model T for a textured region R is represented by the texton occurrence probabilities for all channels:

$$T(R) = \{H_c(R) | 1 \leq c \leq 3\},$$

where $H_c(R)$ denotes the normalized histogram of texton occurrence of a textured region R for channel c. As such, the model of a textured region R consists of 3 normalized histograms, one for each color channel.

Based on the learnt texture models for the regions, we define a texture similarity metric between regions based on the comparison of their histograms, which will be used to compare texture representations in our region merging strategy. Since a normalized histogram provides a discrete estimate for a PDF, given two region appearance models, the Bhattacharyya distance (D_B) [10] between normalized histograms is used for computing the texture dissimilarity between two regions. Assigning a weight to each channel $W_g = [w_L, w_a, w_b]$, the texture dissimilarity d_T between two regions R_i and R_j can be defined as

$$d_T(R_i, R_j) = \frac{W_g \begin{bmatrix} D_B(H_L(R_i), H_L(R_j)) \\ D_B(H_a(R_i), H_a(R_j)) \\ D_B(H_b(R_i), H_b(R_j)) \end{bmatrix}}{\sum_{g=\{L,a,b\}} W_g}. \tag{2}$$

3. STOCHASTIC TEXTURE MERGING AND SEGMENTATION

Based on the texture model we introduced for representing textured regions, let us now describe the proposed stochastic texture merging method. First, an adjacency graph is constructed to model the interactions between neighboring pixels. The adjacency graph $G = (R, E)$ represents the status of the current configuration of the regions, where each vertex represents a region R, and each edge represents the local dissimilarities between two neighboring texture regions. For a given image I, with $N \times M$ pixels, no prior assumptions are made about the number of distinct texture regions in the image; hence we assign a unique texture region label $R = \{1, \cdots, N \times M\}$ to each pixel. Hence, each pixel is associated with a unique vertex, with each initially connected to four other vertexes (adjacent texture regions), representing the 4-neighborhood of that pixel.

A region R_i is merged with an adjacent region R_j with a probability of $\alpha(R_i, R_j)$, which denotes a novel texture likelihood function that extends upon the stochastic region merging criterion proposed by Wong et al. [8] to account for texture characteristics:

$$\alpha(R_i, R_j) = \exp\left[-\frac{d_T(R_i, R_j)}{\Lambda(R_i, R_j)}\right], \tag{3}$$

where $d_T(R_i, R_j)$ denotes the texture dissimilarity between R_i and R_j, defined in Eq. 2, and Λ denotes a statistical merging penalty based on the size of the regions:

$$\Lambda(R_i, R_j) = \frac{D_f^2}{2Q_k}\left[\frac{ln(\Psi(f)^2)}{\Psi(R_i)} + \frac{ln(\Psi(f)^2)}{\Psi(R_j)}\right], \tag{4}$$

where $\Psi(R)$ represents the number of elements (pixels) in the region R, $\Psi(f)$ represents the number of pixels in the image, D_f represents the range of possible values in f, and Q_k represents a regularization term at iteration k which controls the flexibility of the merging likelihood.

Two adjacent regions are merged the following predicate is satisfied:

$$\mathcal{P}(R_i, R_j) = \begin{cases} 1 & \text{if } u \leq \alpha(R_i, R_j), \\ 0 & \text{otherwise .} \end{cases} \tag{5}$$

where u denotes a random number from a uniform distribution between 0 and 1. To merge the regions consistently the evaluation order of the region merging process is important. Therefore, the adjacency graph edges are inserted in a priority queue in the decreasing order of their weights, used in all merging tests.

The proposed stochastic texture merging method (STM) is based on the statistical modeling of the texture region appearance, which may be challenging to obtain for small regions (less than a few pixels) due to the local variability of highly textured regions. To minimize this difficulty we initialize the edge weights with the gradients of the image as pixel-to-pixel texture differences. During this process, whenever a pair of regions is analyzed, they are removed from the queue. Every time a merge occurs, the adjacency graph is updated and the priority queue is modified to reflect these changes. Once the priority queue becomes empty the resulting adjacency graph will yield the initial segmentation result. To reduce the chance of over segmenting the image, we adopt an iterative segmentation refinement. Given the initial segmentation result, the stochastic texture region merging process is successively repeated taking the segmentation obtained in the previous step as the input.

The regions obtained in the previous iteration are used to build a new adjacency graph, its edges are inserted in the priority queue, and sorted in ascending order of region dissimilarity. As the regions on initialization map are progressively larger after the previous merging steps, the queue is sorted using the region-wise texture dissimilarity of Eq. 2. The merging process is then repeated with a lower regularization term,

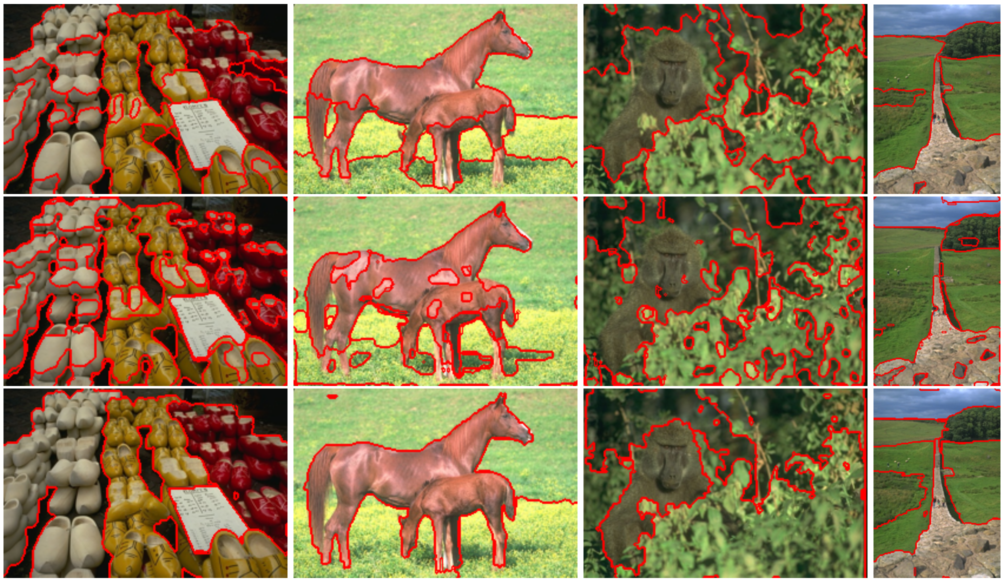

Fig. 1. Visual results of the proposed segmentation method (bottom row) versus JSEG (top) and SRM (Middle). Red contours indicate the boundaries of the segmented regions.

and Q_k is exponentially decreased at each iteration. Therefore, the value of Q used in the k^{th} iteration of the region merging process (denoted as Q_k) can be expressed as:

$$Q_k = (Q_1 - Q_{min}) * \exp(1-k) + Q_{min} \quad (6)$$

where $k \geq 1$ denotes the current iteration, Q_1 represents the value of the regularization term in the first iteration, and $Q_{min} = 200$ represents the minimum value for Q used in our experiments.

At each iteration one Q value is used and the number of regions is expected to decrease. If $Q_i - Q_{i-1}$ is too small, the number of regions remains unaltered, and convergence is reached. In these iterations we can avoid under-segmentation by setting a higher value for Q, and lowering the risk of over-segmentation.

4. EXPERIMENTAL RESULTS

To evaluate the quality of our method segmentation results, we conducted tests on the BSDS300 dataset [11]. This database is publicly available and consists of 300 natural color images of 481×321 pixels each, divided in 200 images for training and 100 images for testing. As our approach have no training stage, we chose to run our tests on the 200 images used for testing, which contains many outdoor scene images. These images were rescaled to 241×161 to reduce computational complexity. Since natural image segmentation can be very subjective, this dataset provides up to 10 handmade segmentations as ground truths for each image.

To extract the texture features, we used image patches of $W = 5$ pixels, and the RP matrix reduces these patches to $M = 10$ vector elements. The texton dictionary was set to have $K = 30$ centroids. In the stochastic region merging stage we set the regularization term Q to $\{100, 200, 400\}$, and the weight vector $W_g^c \in [1, 2]$. The vector W_g weights the contributions of the color and luminance aspects to the similarity measurement. The regularization term Q, on the other hand, affects the final number of segmented texture regions. Increasing Q tends to decrease α in Eq. 3, the merging probability drops, and we obtain more segmented regions. Moreover, decreasing Q with the iterations of the stochastic texture region merging method helps the segmentation convergence (see Eq. 6).

To evaluate each segmented image, the similarity to its ground truth is measured. Let S be the segmentation map of the input image I, and G the ground truth, both formed by a set of non-overlapping regions labeled as $S_i \in \{1, \cdots, k\}$, and $G_j \in \{1, \cdots, n\}$, respectively. We measure segmentation error by counting the overlapping pixels in S and G. We first associate each texture G_i to a region in the segmentation map by:

$$\hat{S}_j = \arg\max_{S_i} |G_j \cap S_i|, \quad (7)$$

where $\hat{S}_j$ is the region in the same spatial position as the texture region G_j. We define the segmentation accuracy given the region map $\{S\}$, and the ground truth set $\{G\}$,

$$\mathrm{ACC}_G^S = \frac{1}{\Phi(I)} \sum_{i=1}^{n} |G_i \cap \hat{S}_i|, \quad (8)$$

where, ACC_G^S indicates the proportion of image pixels that where assigned correctly to the label indicated in $\{G\}$. If one segment is assigned to more than one texture region in G, we consider only the segment with the texture label that has more

Table 1. Performance on the BDSD300 dataset.

Method	Average Acc. (%)	σ (%)
JSEG	76.38	14.10
SRM	77.41	12.56
Proposed ($Q = 100$)	**88.94**	**13.82**
Proposed ($Q = 200$)	83.86	12.02
Proposed ($Q = 400$)	75.00	14.09

pixels classified correctly (which may decrease the accuracy of incorrect segmentations since regions in G may be partially matched).

We compared our results with other state of art methods, like JSEG [12] and stochastic region merging (SRM) [8]. Table 1 shows the comparative results, and indicates that in average the proposed texture segmentation method is more accurate than the other tested methods, and has a similar accuracy standard deviation (σ). In all comparisons, we used the parameters specified in the respective literature, we used $W_g = [1, 1.5, 1.5]$ in the tests with our proposed method (which was determined experimentally). Figure 1 shows a visual comparison of the obtained results. The proposed method appears to be more robust to local variations in the texture regions and detect region boundaries more accurately, while the other methods tend to over-segment the image. Using textures features instead of just colors (as in SRM [8]) appears to better represent the texture regions, while dealing efficiently with local variations inside the texture regions.

5. CONCLUSION

In this paper, we have introduced a stochastic patch-based approach for texture modeling with a texton dictionary to improve the segmentation of texture regions. This approach extends the known stochastic region merging method [8], using a new texture region likelihood criterion. Experiments using the BDSD300 dataset have shown that the proposed method provides a more accurate segmentation than other tested methods proposed in the literature. Visually, this work has shown to be more robust to variations inside the textures, and able to find the boundaries between similar textures, avoiding over segmentation in highly textured regions. Future work include investigating the effectiveness of the proposed technique for recognizing specific textures, as in skin detection and material recognition.

6. REFERENCES

[1] M. A. Hoang, J. M. Geusebroek, and A. W. M. Smeulders, "Color texture measurement and segmentation," *Signal Processing*, vol. 85, no. 2, pp. 265–275, 2005.

[2] Y. Deng and B.S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 8, pp. 800 –810, aug 2001.

[3] M. Varma and A. Zisserman, "A statistical approach to material classification using image patch exemplars," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 11, pp. 2032 –2047, nov. 2009.

[4] C. Fowlkes, D. Martin, and J. Malik, "Learning affinity functions for image segmentation: combining patch-based and gradient-based approaches," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, june 2003, vol. 2, pp. II – 54–61 vol.2.

[5] D.R. Martin, C.C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 5, pp. 530 –549, may 2004.

[6] Li Liu and P. Fieguth, "Texture classification from random features," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 3, pp. 574 –586, march 2012.

[7] R. Nock and F. Nielsen, "Statistical region merging," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 11, pp. 1452 –1458, nov. 2004.

[8] A. Wong, J. Scharcanski, and P. Fieguth, "Automatic skin lesion segmentation via iterative stochastic region merging," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 15, no. 6, pp. 929 –936, nov. 2011.

[9] Li Liu, Paul Fieguth, and Gangyao Kuang, "Compressed sensing for robust texture classification," in *Computer Vision ACCV 2010*, Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto, Eds., vol. 6492 of *Lecture Notes in Computer Science*, pp. 383–396. Springer Berlin / Heidelberg, 2011.

[10] Frank J. Aherne, Neil A. Thacker, and Peter I. Rockett, "The bhattacharyya metric as an absolute similarity measure for frequency coded data," *Kybernetika*, vol. 32, pp. 1–7, 1998.

[11] 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. 8th Int'l Conf. Computer Vision*, July 2001, vol. 2, pp. 416–423.

[12] Yining Deng, B.S. Manjunath, and H. Shin, "Color image segmentation," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 1999, vol. 2, pp. 2 vol. (xxiii+637+663).