Perceptually Consistent Segmentation of Texture Using Multiple Channel Filter*

Nan Zhang and Wee Kheng Leow

Department of Information Systems and Computer Science
National University of Singapore, Lower Kent Ridge Road, Singapore 119260
email: zhangnan, leowwk@iscs.nus.edu.sg

Abstract. Texture segmentation aims at dividing an image into perceptually uniform regions each containing a distinct texture. In images of natural scene, texture in a region can change gradually in scale and orientation due to perspective distortion. A naive segmentation method may erroneously group image patches with the same texture but slowly varying scales and orientations into distinct regions. This paper describes a novel segmentation method which takes into account the rate of change of texture scale and orientation. The method extracts scale and orientation information from the outputs of a set of Gabor filters, and use them to group image patches into perceptually uniform texture regions.

1 Introduction

Texture segmentation aims at dividing an image into perceptually uniform regions each containing a distinct texture. It is a very difficult task for images of natural scene. In these images, texture in a perceptually uniform region can change gradually in scale and orientation due to perspective distortion. Take Fig. 1 for example. This image contains several perceptually uniform regions each covered by one type of material: bricks or pebbles. Texture in the image decreases gradually in scale with increasing viewing distance from the bottom of the image to the top. Orientation of the brick texture also changes gradually from the left of the image to the right. Texture boundaries are perceived at locations where the type of texture differs (e.g., between bricks and pebbles) or when the texture scale or orientation changes abruptly (e.g., between the top and the bottom brick patterns).

A naive segmentation method may erroneously group image patches with the same texture but slowly varying scales and orientations into distinct regions. On the other hand, it may erroneously group neighboring patches with the same texture but sharp changes in scales or orientations into the same region. To correctly segment an image, a texture segmentation algorithm has to take into account the rate of change of texture scale and orientation. So far, existing works on image segmentation have not considered this issue. This paper presents a novel method that performs perceptually consistent segmentation of images containing natural texture.

---

* This research is supported by NUS Academic Research Grant RP950656 and NUS Research Scholarship HD950345.

Appears in *Proceedings of Asian Conference on Computer Vision*, 1998.

Fig. 1. Image of floor containing several perceptually uniform regions each covered by a different type of texture: bricks or pebbles. Texture in a region can vary gradually in scale and orientation.

2 Related Works

Texture segmentation of images is performed based on texture features which can be divided into 4 categories: Markov random field, local statistics, Fourier spectrum, and Gabor filter magnitudes.

The Markov random field (MRF) model assumes that texture is formed by a stochastic process such that a pixel's intensity is given by a weighted sum of the intensities of neighboring pixels and a noise term [7, 10]. It is very difficult for MRF model to be scale- and orientation-invariant since it would require a different set of weights for every possible scales and orientations.

Local statistics such as the means, variances, and 3rd-order moments computed from the eigenvalues of covariance matrices have also been used as texture features [8, 9]. Although some of the features are orientation-invariant, they lack the information needed to determine whether the change of scale and orientation is gradual or abrupt.

Fourier spectrum can capture texture's frequency and orientation [4, 7]. Unfortunately, it is not localized in the spatial domain. It is impossible to extract a textured region's Fourier spectrum unless the region has already been segmented.

Gabor filters have the advantage of being optimally localized simultaneously in the spatial and the spatial frequency domains. Existing segmentation methods based on Gabor filters [1, 2, 3, 6] use only some of the filter channels, typically those with large output magnitudes. The invariant segmentation method described in this paper, however, uses all the filter channels so as to better estimate the rate of change of texture frequency and orientation. Existing methods handle only images containing *uniform texture*, i.e., texture that does not vary in scale and orientation within a perceptually uniform region. Typically, these images contain juxtaposed plane texture taken from the Brodatz album. Since these methods do not take into consideration gradual variation of texture scale and orientation, they cannot correctly segment images containing non-uniform texture. In contrast, the invariant method can segment images of natural scene that contain multiple non-uniform texture.

3 Multi-channel Texture Feature Extraction

The Gabor function $h(x,y)$ at image position (x,y) is a complex sinusoidal grating modulated by an oriented Gaussian function $g(x',y')$ [2]:

$$\begin{aligned} h(x,y) &= g(x',y')\exp(2\pi j f x') \\ g(x',y') &= \frac{1}{2\pi\lambda\sigma^2}\exp\left[-\frac{(x'/\lambda)^2+y'^2}{2\sigma^2}\right] \end{aligned} \tag{1}$$

where $(x',y') = (x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta)$ are rotated coordinates oriented at angle θ from the x-axis, λ is the aspect ratio, and σ is the scale parameter. The Gabor function has radial frequency f and orientation θ:

$$f = \sqrt{U^2+V^2} \qquad \theta = \tan^{-1}\left(\frac{V}{U}\right) \tag{2}$$

where U and V are spatial frequencies along the x- and the y-directions. Its frequency (octave) bandwidth B and orientation (radian) bandwidth Ω at half-peak are:

$$B = \log_2\left[\frac{\pi f\lambda\sigma+\alpha}{\pi f\lambda\sigma-\alpha}\right] \qquad \Omega = 2\tan^{-1}\left(\frac{\alpha}{\pi f\sigma}\right) \tag{3}$$

where $\alpha = \sqrt{(\ln 2)/2}$. The range of spatial frequencies within the frequency and orientation bandwidths is called the *half-peak support*.

A multi-channel approach is adopted to represent texture. An input image $I(x,y)$ is filtered by a set of Gabor filters with different frequency f and orientation θ:

$$\begin{aligned} k_{\mathrm{c},f\theta}(x,y) &= h_{\mathrm{c},f\theta}(x,y) * I(x,y) \\ k_{\mathrm{s},f\theta}(x,y) &= h_{\mathrm{s},f\theta}(x,y) * I(x,y) \end{aligned} \tag{4}$$

where $h_{\mathrm{c},f\theta}(x,y)$ and $h_{\mathrm{s},f\theta}(x,y)$ are the real and the imaginary components of Gabor function (Eq. 1). The magnitudes of the filters' outputs are given by:

$$k_{f\theta}(x,y) = \sqrt{k^2_{\mathrm{c},f\theta}(x,y) + k^2_{\mathrm{s},f\theta}(x,y)}\,. \tag{5}$$

After Gabor filtering, the channels' outputs are smoothed by Gaussian filters to remove local variations introduced by the sinusoidal terms in the Gabor functions. The Gaussians have the same orientations and aspect ratios as the Gabors. Their scale parameters σ are set at four times those of the corresponding Gabors. After smoothing, the Gabor output magnitudes at each pixel location form the *texture feature vector* $\mathbf{k}(x,y)$ in the f-θ space.

Unlike existing methods, the Gabor filters' frequencies f_i and orientations θ_i are chosen such that there are some overlaps in the filters' half-peak supports:

$$f_i = \frac{f_{\mathrm{m}}}{2^{i\beta B}} \qquad \theta_i = i\beta\Omega \tag{6}$$

where f_{m} is the maximum spatial frequency and β determines the amount of overlap in the filters' supports. The amount of overlap is minimum when $\beta = 1$

and increases with decreasing β. In the current implementation, $f_{\mathrm{m}} = 0.3$, $B = 0.75$ octave, $\Omega = 45°$, and $\beta = 0.5$. A total of 48 filters are used, with 6 spatial frequencies and 8 orientations.

In contrast to existing methods which use only some of the Gabor channels' outputs, the invariant segmentation method uses the *pattern* of outputs from all the channels as the texture feature. This method of representing texture feature is known as *distributed representation* in the Neural Networks literatures [5]. Distributed representation has the advantage of representing a large number of different texture types using a small number of channels' outputs. It also ensures that the output pattern will not be severely altered by slight changes in texture frequency and orientation, thereby facilitating a closer match between neighboring texture patches.

4 Invariant Texture Segmentation

After feature extraction, the texture feature vectors at each pixel location are grouped into regions in 2 stages: (1) region grouping, and (2) region merging.

4.1 Region Grouping

In the region grouping stage, similar texture feature vectors at neighboring locations are first grouped into *seed regions*. This operation reduces the number of feature vectors that will be involved in the next stage. Each seed region R_i is characterized by a *peak* vector $\mathbf{P}_i$ and a *mean* vector $\mathbf{M}_i$. Initially, one seed region R_1 is created which contains the feature vector $\mathbf{k}(0,0)$ at location $(0,0)$, and $\mathbf{P}_1$ and $\mathbf{M}_1$ are set to $\mathbf{k}(0,0)$. Subsequently, a feature vector $\mathbf{k}(x,y)$ is grouped into region R_i if

- it is near enough: there is a vector $\mathbf{k}'(x',y')$ in R_i that is a 4-neighbor of $\mathbf{k}(x,y)$ in the x-y space,
- it is similar enough: similarity $C(\mathbf{k},\mathbf{P}_i) > \Gamma_{\mathrm{P}}$, and $C(\mathbf{k},\mathbf{M}_i) > \Gamma_{\mathrm{M}}$,

where $\Gamma_{\mathrm{P}} = \cos 5°$ and $\Gamma_{\mathrm{M}} = \cos 3°$ are constant thresholds, and C is the cosine similarity:

$$C(\mathbf{k}_1,\mathbf{k}_2) = \frac{\mathbf{k}_1 \cdot \mathbf{k}_2}{||\mathbf{k}_1||\ ||\mathbf{k}_2||}\ . \tag{7}$$

If a vector can be grouped into more than one region, then one of the regions is arbitrarily chosen to contain the vector. After grouping $\mathbf{k}$ into region R_i, the peak vector $\mathbf{P}_i$ and mean vector $\mathbf{M}_i$ are updated as follows:

$$\mathbf{P}_i = \max_{\mathbf{k}_j \in R_i} \mathbf{k}_j \qquad \mathbf{M}_i = \frac{1}{n} \sum_{\mathbf{k}_j \in R_i} \mathbf{k}_j \tag{8}$$

where n is the number of vectors in R_i.

4.2 Region Merging

After region grouping, seed regions are merged in the merging stage into larger regions that are more consistent with human's perception of the input image. Two regions that share a common boundary are merged if their feature vectors are similar enough. The similarity measure used in this stage is slightly more complex because it has to take into account gradual variations of texture scale and orientation. It can be shown that an isotropic scaling or a rotation of a texture image results in only a shift of the Gabor output pattern in the f-θ space. In an image of natural scene, however, changes in scale may not be isotropic due to perspective distortion. As a result, some parts of the Gabor output pattern may shift more than other parts. Therefore, to compute the similarity between two feature vectors, it is necessary to consider non-uniform shifting of Gabor output pattern.

The similarity measure between feature vectors are defined as follows. Suppose that feature vector $\mathbf{k}$ has a peak at (f, θ). Form a *local peak vector* by taking the feature vector components $k_{f'\theta'}$ in the 3×3 neighborhood of (f, θ). Now, consider two feature vectors $\mathbf{k}_1$ and $\mathbf{k}_2$ each having a set of local peak vectors $\{\mathbf{p}_i\}$ and $\{\mathbf{q}_j\}$. The *invariant similarity* $S(\mathbf{k}_1, \mathbf{k}_2)$ between $\mathbf{k}_1$ and $\mathbf{k}_2$ is defined as

$$S(\mathbf{k}_1, \mathbf{k}_2) = \frac{1}{2}[\zeta(\mathbf{k}_1, \mathbf{k}_2) + \zeta(\mathbf{k}_2, \mathbf{k}_1)] \tag{9}$$

where $\zeta(\mathbf{k}_1, \mathbf{k}_2)$ (and, similarly, $\zeta(\mathbf{k}_2, \mathbf{k}_1)$) is given as follows:

For each $\mathbf{p}_i$ of $\mathbf{k}_1$, find the matching $\hat{\mathbf{q}}_i$ such that

$$C_{\mathrm{d}}(\mathbf{p}_i, \hat{\mathbf{q}}_i) = \max_l C_{\mathrm{d}}(\mathbf{p}_i, \mathbf{q}_l)\ . \tag{10}$$

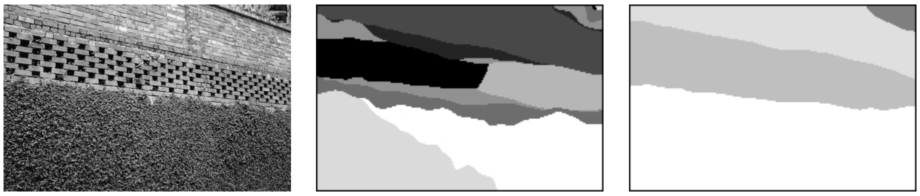
The similarity measure C_{d} is a distance-weighted vector dot-product:

$$C_{\mathrm{d}}(\mathbf{p}, \mathbf{q}) = \frac{w(\mathbf{p}, \mathbf{q})\, \mathbf{p} \cdot \mathbf{q}}{||\mathbf{p}||\ ||\mathbf{q}||} \tag{11}$$

where $w(\mathbf{p}, \mathbf{q})$ is a constant weighting factor that decreases with increasing Euclidean distance between $\mathbf{p}$ and $\mathbf{q}$ in the f-θ space. In other words, Eq. 10 finds the $\hat{\mathbf{q}}_i$ that is most similar and nearest (in f-θ space) to $\mathbf{p}_i$. Then, $\zeta(\mathbf{k}_1, \mathbf{k}_2)$ is computed as a normalized, weighted vector dot-product of all the local peak vectors $\mathbf{p}_i$ of $\mathbf{k}_1$ and their matching $\hat{\mathbf{q}}_i$ of $\mathbf{k}_2$:

$$\zeta(\mathbf{k}_1, \mathbf{k}_2) = \frac{\displaystyle\sum_i w(\mathbf{p}_i, \hat{\mathbf{q}}_i)\, \mathbf{p}_i \cdot \hat{\mathbf{q}}_i}{\sqrt{\displaystyle\sum_i ||\mathbf{p}_i||^2 \sum_i ||\hat{\mathbf{q}}_i||^2}}\ . \tag{12}$$

In the beginning of the merging process, each (seed) region R_i is characterized by its mean feature vector $\mathbf{M}_i$. As two regions are merged into one, both their mean vectors are collected in the merged region. It is necessary to collect the mean vectors instead of averaging them because, as a region gets larger in size, the mean vectors at its extreme ends may differ significantly even though they vary gradually over the entire region (e.g., Fig. 1). Two neighboring regions $\mathbf{R}_i$ and $\mathbf{R}_j$ can be merged if



(a) (b) (c)

Fig. 2. (a) Image of a wall covered with leaves and two types of brick texture. (b) Non-invariant segmentation method splits a perceptually uniform region into several fragments (painted in different shades of gray). (c) Invariant segmentation method produces regions that are more consistent with human perception.

- they share common boundaries, and
- they are similar enough: $S(\mathbf{M}_i, \mathbf{M}_j) > \Gamma_S$

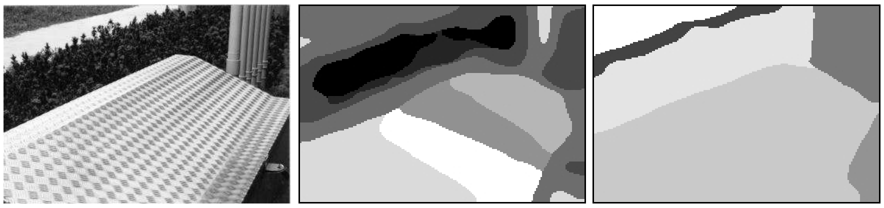
where $\mathbf{M}_i$ and $\mathbf{M}_j$ are the regions' mean vectors that are nearest to each other in the x-y space, and Γ_S is a constant threshold currently set at $\cos 23°$. At the end of the merging stage, the result is cleaned-up by merging very small regions into neighboring larger regions that are most similar in texture.

5 Test Results

This section illustrates some results of applying the texture segmentation method described in Section 4 on images of natural scenes. Figure 2 compares the performance of the invariant segment method with a typical non-invariant one. The non-invariant method used here is derived from the invariant method by replacing the invariant similarity measure S by a simpler cosine similarity C. The non-invariant method tends to split a perceptually uniform region into several fragments when there are significant differences between the scales or orientations of the texture patches (Fig. 2b). In contrast, the invariant method produces results that are more consistent with human perception (Fig. 2c).

Figure 3 illustrates another comparison between invariant and non-invariant methods. Due to perspective distortion, the texture on the metal cover in Fig. 3(a) changes gradually from very coarse scales to very fine scales. The non-invariant method segments the metal cover into several fragments each containing a texture of approximately the same scale (Fig. 3b). In contrast, the invariant method segments the cover into a single region (Fig. 3c). It also segments the other regions in the image well.

Figure 4 shows the result of segmenting Fig. 1. The two brick patterns at the top and the bottom are of the same type except for a rotation of 90°. The invariant method correctly identifies the texture boundary indicated by the abrupt change in orientation. It also groups image patches with the same texture but gradually varying scales and orientations into the same region. The final example illustrates the segmentation of boundaries formed by irregular texture, such as



(a) (b) (c)

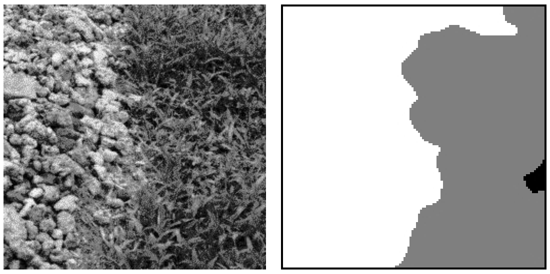
Fig. 3. (a) An image of a complex scene. (b) Non-invariant segmentation method splits the metal cover into several fragments according to the scale of the texture. (c) Invariant segmentation method identifies the patterns over the entire cover as the same texture despite gradual change in scale.

Fig. 4. Segmentation of Fig. 1. Invariant segmentation method correctly identifies the boundaries between the top and the bottom brick patterns that differ in orientations. It also segmented the pebble and the brick texture.

grass and rocks, that has rather ill-defined orientation. Figure 5 shows that the invariant segmentation method can identify the curvilinear boundary between the grass lawn and the rocks.

6 Conclusions

This paper has described a novel texture segmentation method that is invariant to gradual changes in texture scale and orientation. Two key features make the algorithm scale- and orientation-invariant. First, the method uses the pattern of outputs from all the Gabor channels as a distributed representation of texture feature. The channels have overlapping half-peak supports that reduce the variation of output pattern due to slight changes in texture scale and orientation. Second, the invariant similarity measure takes into account the non-uniform shifts of feature pattern due to perspective distortion, and performs individual matching of the local peaks in the feature pattern. The invariant segmentation method has been shown to perform well on images of natural scene containing multiple non-uniform texture patterns. In comparison with existing methods, it produces segmentation results that are more consistent with human perception.



(a) (b)

Fig. 5. (a) Image of ground covered with grass and rocks. (b) Invariant segmentation method can identify the curvilinear boundary between the grass lawn and the rocks.

References

1. J. Bigün and J. M. H. du Buf. N-folded symmetries by complex moments in Gabor space and their application to unspervised texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):80–87, 1994.
2. A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, 1990.
3. D. F. Dunn, W. E. Higgins, and J. Wakeley. Texture segmentation using 2-D Gabor elementary functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):130–149, 1994.
4. J. M. Francos, A. Zvi Meiri, and B. Porat. A unified texture model based on a 2-D Wold like decomposition. *IEEE Transactions on Signal Processing*, pages 2665–2678, Aug. 1993.
5. G. E. Hinton, J. L. McClelland, and D. E. Rumelhart. Distributed representation. In David E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing*. MIT Press, Cambridge, Massachusetts, 1986.
6. A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.
7. F. Liu and R. W. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):722–733, 1996.
8. S. V. R. Madiraju, T. M. Caelli, and C.-C. Liu. On the covariance technique for robust and rotation invariant texture processing. In *ACCV'93 Asian Conference on Computer Vision*, pages 171–174, 1993.
9. S. V. R. Madiraju and C.-C. Liu. Rotation invariant texture classification using covariance. In *Proceedings of International Conference on Image Processing*, volume 2, pages 655–659, 1994.
10. D. K. Panjwani and G. Healey. Markov random field models for unsupervised segmentation of texture color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):939–954, 1995.