SCALE AND ORIENTATION-INVARIANT TEXTURE MATCHING FOR IMAGE RETRIEVAL

WEE KHENG LEOW
School of Computing, National University of Singapore, Lower Kent Ridge Road, Singapore 119260
E-mail: leowwk@comp.nus.edu.sg

SEOW YONG LAI
DSO National Laboratories, 20 Science Park Drive, Singapore 118230
E-mail: lseowyon@dso.org.sg

Texture-based image retrieval is a very difficult task, especially for retrieving images of natural scene. Such images contain multiple texture patterns that may vary in intensity, scale, and orientation but still look the same to humans. Existing methods have been successful in retrieving images that contain single uniform texture but their performance often deteriorate when retrieving natural scene images. This article presents an invariant texture matching method that can retrieve images containing texture patterns that differ in intensity, scale, and orientation from the query texture. Experimental results show that the invariant method performs better than existing methods especially in retrieving natural scene images.

1 Introduction

Retrieval of images, especially images of natural scene, based on texture is a very difficult task. Such images contain a wide variety of natural texture such as bricks, wire mesh, sand, and grass patches. Non-uniform lighting conditions resulting from shadows, time of the day, etc. can alter the apparent brightness of the texture. In addition, texture patterns in natural scene images tend to vary smoothly in scale and orientation due to perspective distortion. Nevertheless, humans still perceive them as the same texture. Therefore, texture matching should be invariant to texture intensity, scale, and orientation.

Main texture classification techniques used for content-based image retrieval can be categorized into two approaches: statistical and spectral. Statistical methods characterize texture in terms of local statistical measures (e.g., coarseness, directionality, and contrast) [1], simultaneous autoregressive models (SAR) [2], or Markov random field models [3]. Variations of these methods have been shown to be rotation and scale-invariant in classifying Brodatz textures [4,5]. In general, these models are good at modeling random patterns such as sand and pebbles but not so suitable for modeling highly structured patterns such as roof tiles and bricks. Liu and Picard [6] showed that the

SAR method does not return satisfying results in retrieving images of structured patterns. Among these methods, Tamura's texture features [1] are used in IBM's QBIC [7] and Markov random field model is used in MIT's PhotoBook [6]. In addition, MIT's PhotoBook also uses Discrete Fourier Transform to extract texture features.

The spectral approach is based on the response of a set of band-pass filters, usually the 2D Gabor filters [8]. Each filter responds most strongly to a selected spatial-frequency and orientation bands. The advantage of this approach is that both scale and orientation information about the texture can be captured by the filters. This method has been used extensively for classifying plane textures such as Brodatz textures and aerial images [8,9,10,11]. It has also been adopted in the NeTra image retrieval system [12].

Existing texture matching methods, for example [1,6,7,12], can effectively handle images containing single texture patterns that are uniform in scale and orientation. However, their performance often deteriorate when they are used to match texture patterns that differ in scale and orientation. This article presents a texture matching method that is invariant to intensity, scale, and orientation of texture. Experimental results show that it performs better than existing methods especially in retrieving images of natural scenes.

2 Texture Feature Extraction

To perform texture matching, texture features are first extracted from the images, and the images are segmented into regions each containing a homogeneous texture pattern. Texture features are extracted from an image using a set of Gabor filters [8]:

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

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 maximum spatial frequency is set at 0.3 cycle per pixel because Gabor filters with spatial-frequency higher than 0.3 cycle per pixel cannot be accurately represented in digital form [8]. The spatial frequency bandwidth is set at 0.5 octave so that 6 frequency channels can cover the spatial frequency range of most texture patterns. The orientation bandwidth is set at 45°, which allows 8 orientation channels to cover all possible orientations. Hence, a total of 48 filters with 6 spatial frequencies and 8 orientations are used.

An input image $I(x, y)$ is filtered by a set of Gabor filters with different frequencies f and orientations θ:

$$\begin{aligned} e_{c,f\theta}(x,y) &= I(x,y) * h_{c,f\theta}(x,y) \\ e_{s,f\theta}(x,y) &= I(x,y) * h_{s,f\theta}(x,y) \end{aligned} \tag{2}$$

where $h_{c,f\theta}(x,y)$ and $h_{s,f\theta}(x,y)$ are the real and imaginary components of the Gabor function. The Gabor channels' output energy is computed as:

$$E_{f\theta}(x,y) = e^2_{c,f\theta}(x,y) + e^2_{s,f\theta}(x,y) \ . \tag{3}$$

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 channels' outputs at each pixel location are further normalized to remove intensity variation by dividing the energy value of each channel by the largest value at that location:

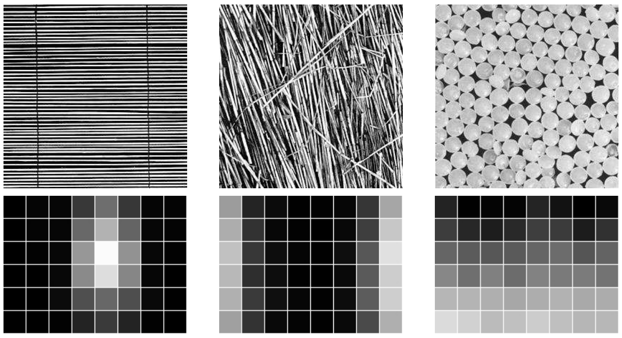
$$G_{f\theta}(x,y) = \frac{E_{f\theta}(x,y)}{\max\limits_{f,\theta} E_{f\theta}(x,y)} \ . \tag{4}$$

Since the energy of Gabor output is proportional to the (square of the) image intensity, the normalization process will remove variations in Gabor outputs due to differences in texture image intensity. The normalized Gabor outputs now form a multi-dimensional texture feature vector at each pixel location that is invariant to image intensity.

Once the feature vectors are extracted, a region segmentation algorithm is applied to segment the image into regions each containing a homogeneous texture. For the purpose of image retrieval, the region segmentation algorithm need not localize the region boundaries precisely. It only has to ensure that each region contains a single texture pattern. In the current implementation, segmentation is performed by computing the normalized vector dot product of the texture features at neighboring positions, and grouping positions with large product into the same region.

3 Invariant Texture Matching Method

The invariant texture matching method consists of two steps: (1) feature transformation and (2) feature matching. The first step transforms the texture feature vectors into a texture space that is invariant to scale and orientation. The second step performs matching both in the invariant texture space as well as using the original feature vectors.



(a) (b) (c)

Figure 1. The structures of different textures (top row) manifest as different patterns in the 2D representations of Gabor feature vectors (bottom row). In the 2D representations, spatial frequency increases from bottom up along the vertical axis, and orientation changes from left to right along the horizontal axis and wraps around. Each square represents the output of a Gabor channel, with light squares denoting large outputs.

3.1 Feature Transformation

Texture feature transformation is performed based on an important characteristic of the Gabor feature vectors, which can be represented in a 2-dimensional pattern with frequency increasing along the vertical axis and orientation along the horizontal axis (Fig. 1). Three distinct patterns are observed for three extreme types of textures. Highly structured texture with specific scale and orientation has localized peaks in the 2D representation (Fig. 1a). Oriented texture with a specific orientation and a range of scales has a column-shaped pattern in the 2D representation (Fig. 1b). Granular texture with a specific scale and a range of orientations has a row-shaped pattern (Fig. 1c). These distinct patterns correspond to three important texture characteristics: structuredness, orientedness, and granularity. They form three axes of the invariant texture space. A typical texture would contain a mixture of these three texture characteristics. Different texture patterns would have different 2D representations and would map into different points in the texture space. This method of representing various textures is known as *distributed representation* in the Neural Networks literature [13].

Interestingly, the texture categories defined above correspond closely to the texture characteristics used in the Wold texture model. However, the texture spaces defined in the two models are not identical. In particular, the Wold model defines periodicity, directionality, and randomness as the texture dimensions. The invariant texture space, on the other hand, defines structuredness (similar to periodicity), orientedness (similar to directionality), and granularity as the texture dimensions. Randomness of the Wold model corresponds to a region in the invariant space rather than a dimension.

Similar texture categories have been found to be important texture attributes in the study of Rao and Lohse on human's perception of texture similarity [14]. In the study, repetitiveness, directionality and complexity are the three texture attributes found to be most important to human's perception of texture and they correspond to the structuredness, orientedness and randomness of the invariant texture space.

We have proved mathematically for a sinusoidal texture that a change of its spatial frequency or orientation does not alter its corresponding 2D Gabor pattern but only shifts the pattern along the vertical or horizontal axis [15]. Empirical results show that this property is largely preserved for real images of natural textures (e.g., Fig. 2).

The amounts of texture characteristics possessed by a texture are computed using a template matching method. Three distinct masks are devised to detect the presence of local peaks, columns of peaks, and rows of peaks in the 2D representation:

$$c_i = \max_{f\theta}(\mathbf{G}_{f\theta} \star \mathbf{T}_i) \tag{5}$$

where c_i is the amount of texture characteristic of type i for $i = 1, 2, 3$, $\mathbf{G}_{f\theta}$ is a 2D representation of Gabor texture feature, $\mathbf{T}_i$ is the mask, and $\star$ denotes the normalized cross-correlation operation. The mask values are obtained using a training method similar to the "tuned" masks method [4].

Since a scale or orientation change in a texture results in only a shift of its 2D Gabor representation along the vertical or horizontal axis, roughly the same amounts of texture characteristics are contained in the 2D Gabor representations of the scaled and rotated versions of a texture. The template matching operation (Eq. 5) will detect the amounts of texture characteristics present in a texture regardless of the locations (i.e., spatial frequencies and orientations) of the characteristics patterns (Fig. 1) in the 2D Gabor representation. Therefore, it transforms the texture feature vectors, which contain frequency and orientation information, into vector points in the 3D texture space which is invariant to scale and orientation. Together with normalization of texture energy (Eq. 4), the vector points now correspond to intensity-, scale-, and orientation-invariant representations of the textures.

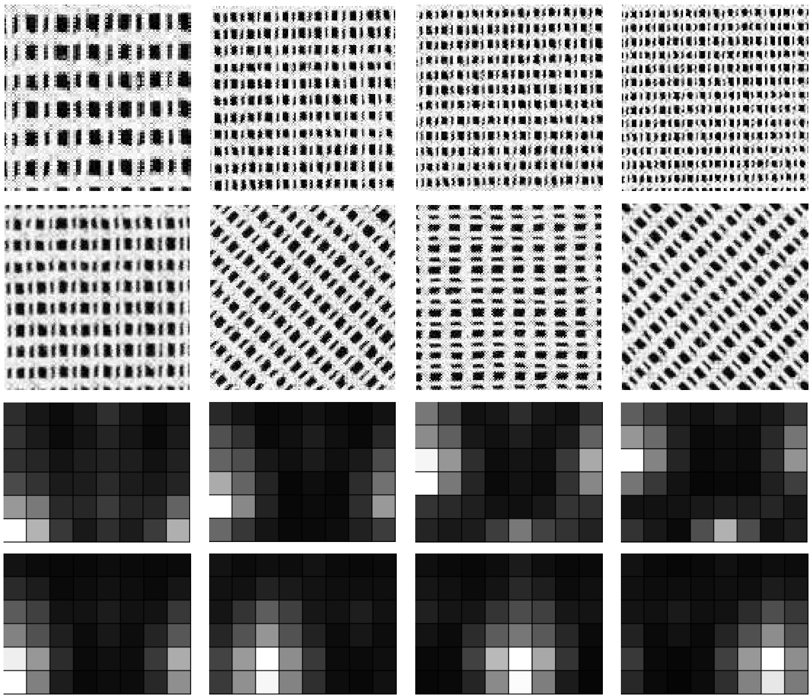


Figure 2. Feature pattern shifts according to changes in texture frequency and orientation. Top two rows show a simple texture in various scales and orientations. Last two rows show the feature patterns in the 2D representation, with frequency increasing from bottom up and orientation changing from left to right. As the frequency and orientation change, the feature pattern shifts along the vertical and horizontal axes.

3.2 Feature Matching

After feature transformation, the texture patterns can now be matched in the invariant texture space using simple Euclidean distance measure. The smaller the Euclidean distance between two vector points, the more similar are the corresponding texture patterns.

Ideally, texture patterns that differ only in scales and orientations should map to a single point in the texture space. In practice, however, due to limited resolution and the presence of image noise, these patterns actually map into a region in the texture space, and neighboring regions corresponding to similar looking textures may overlap. Consequently, ranking based on the Euclidean distance in the texture space alone may not accurately reflect the similarity between the texture patterns.

To improve the ranking of the textures, a rank refinement process is performed as follows. A texture t's original Gabor feature vector (Eq. 4) is compared with those of its nearest N neighbors down the ranked list. The comparison is performed using normalized vector dot product. Neighbors that are more similar to texture t are moved up the list whereas those that are less similar are moved down. This procedure is repeated for every texture in the ranked list.

After rank refinement, the ranked list of texture patterns is mapped into a ranked list of images by identifying the images that contain the texture patterns. An image may appear more than once in the ranked list if it contains multiple texture regions that match the query texture. The image instance that is at the top of the list is retained whereas the others are discarded. The final ranked list of images is the result of the invariant texture matching method.

4 Experimental Results

Extensive experiments have been conducted to compare the performance of the invariant texture matching method and existing methods. Two existing methods were considered:

1. matching method based on the texture model of Tamura et al. [1] because it is used in many existing image retrieval systems [7], and
2. matching method based on the Wold model because it has been reported to outperform other existing methods for retrieving Brodatz texture images [6]. It would be reasonable to regard the Wold method as a representative of the state-of-the-art texture-based image retrieval method.

The source codes of these methods were downloaded from the FTP sites vismod.www.media.mit.edu/pub/fliu/wold and /pub/tpminka/MRSAR.

The three texture matching methods were tested on retrieving both Brodatz textures and natural scene images. For Brodatz texture retrieval tests, each image in the Brodatz album provides nine 256×256 patches. A total of 1008 patches were cropped from the 112 images in the album. Texture patches that are perceptually identical are classified into the same group based on the work done by Ma and Manjunath [16]. A total of 38 texture groups were formed each containing 1 to 7 types of texture.

In a retrieval test, one of the texture patches served as the query texture and the three texture matching methods were executed to rank the other 1007

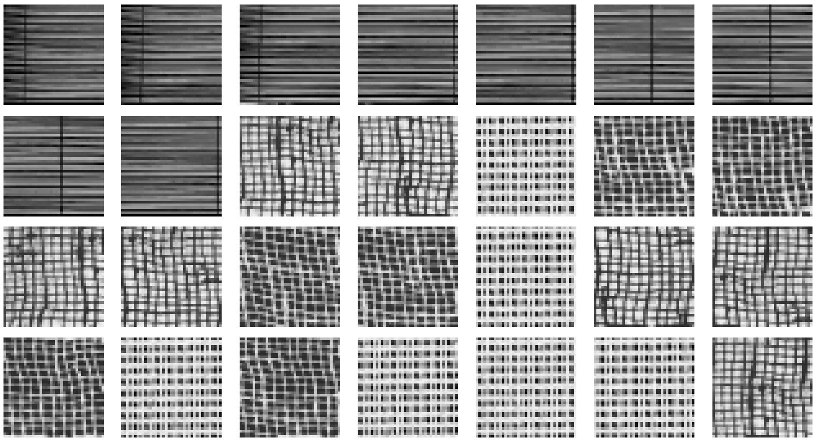


Figure 3. Sample Brodatz image patches retrieved using Tamura et al.'s texture features in decreasing order of similarity. The first patch is the query texture.

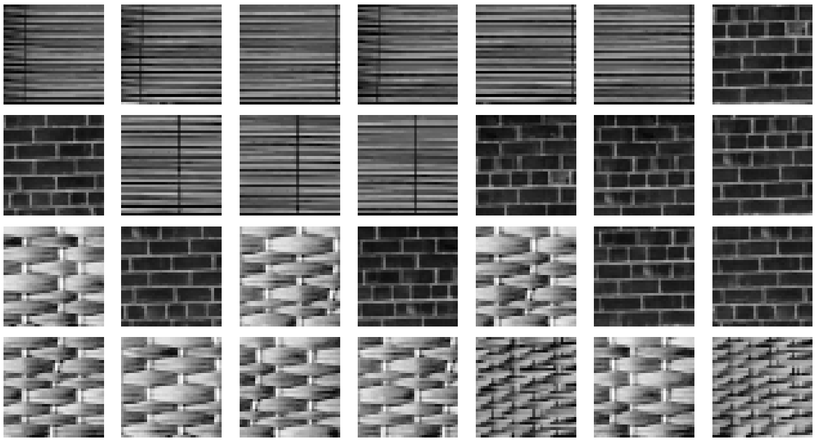


Figure 4. Sample Brodatz image patches retrieved using the Wold texture features in decreasing order of similarity. The first patch is the query texture.

patches. Sample retrieval results are shown in Figs. 3–5. The results show that the methods based on the Tamura features and the Wold model can retrieve texture patches that appear to be quite different from the query texture. On the other hand, the texture patches retrieved by the invariant matching method are more perceptually similar to the query texture, even when their

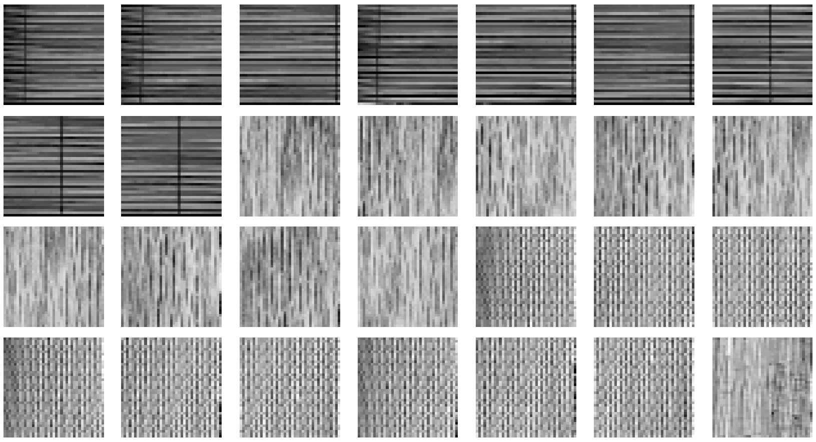


Figure 5. Sample Brodatz image patches retrieved by the invariant texture matching method in decreasing order of similarity. The first patch is the query texture.

Table 1. Performance of texture matching methods in retrieving Brodatz texture images. P_n and R_n are the normalized precision and recall rates, and diff is the improvement of the invariant method over the other methods.

	Tamura		Wold		Invariant	
query	P_n	R_n	P_n	R_n	P_n	R_n
D15	1.00	1.00	1.00	1.00	0.98	1.00
D31	0.65	0.85	0.70	0.81	0.58	0.75
D40	0.47	0.81	0.58	0.81	0.67	0.85
D49	1.00	1.00	0.99	1.00	1.00	1.00
D53	0.62	0.82	0.55	0.78	0.95	0.99
D66	0.43	0.61	0.85	0.96	0.62	0.86
D82	0.51	0.70	0.89	0.97	0.76	0.94
D96	0.70	0.87	0.72	0.93	0.84	0.94
D100	0.83	0.98	0.84	0.99	0.96	0.99
D112	0.87	0.97	0.92	0.98	0.83	0.97
mean	0.71	0.86	0.80	0.92	0.82	0.93
diff (%)	15.7	7.9	1.9	0.7	–	–

orientations differ from that of the query texture. Ten retrievals were performed using representatives of structured, oriented, granular, and random texture patterns as the query texture. Table 1 summarizes the performance of the three methods measured in terms of normalized precision and recall. The Wold method, with very high precision and recall, is indeed a very good

Table 2. Performance of the texture matching methods in retrieving natural scene images. P_n and R_n are the normalized precision and recall rates, and diff is the improvement of the invariant method over the other methods.

	Tamura		Wold		Invariant	
query	P_n	R_n	P_n	R_n	P_n	R_n
1	0.62	0.81	0.18	0.36	0.82	0.89
2	0.32	0.50	0.17	0.33	0.55	0.62
3	0.57	0.68	0.30	0.48	0.64	0.72
4	0.32	0.61	0.38	0.52	0.59	0.62
5	0.33	0.70	0.33	0.67	0.72	0.86
6	0.33	0.70	0.39	0.70	0.82	0.94
7	0.50	0.71	0.23	0.50	0.77	0.86
8	0.58	0.72	0.45	0.67	0.62	0.73
9	0.16	0.38	0.42	0.69	0.63	0.69
10	0.31	0.60	0.16	0.42	0.55	0.70
mean	0.40	0.64	0.30	0.53	0.67	0.76
diff (%)	66.1	19.0	122.9	42.9	–	–

method for retrieving Brodatz images [6]. Nevertheless, the invariant method can still perform slightly better than the Wold method.

In the natural image retrieval tests, 539 texture regions were extracted from 100 images of natural scene. In each retrieval, one of these texture patterns or a Brodatz texture patch was used as the query texture and the three texture matching methods were executed to rank the natural scene images. A sample retrieval result is shown in Fig. 6.

Ten retrievals were performed using representatives of structured, oriented, granular, and random texture patterns as the query texture. Table 2 summarizes the performance of the three methods in retrieving natural scene images. Surprisingly, the Wold method did not perform as well as the Tamura method in retrieving natural scene images. A possible explanation is that the Wold model uses Discrete Fourier Transform (DFT) to extract texture features. Unfortunately, DFT can be drastically affected by image inhomogeneity [6] such as scale and orientation variations due to perspective distortion. In contrast, the invariant method performs significantly better than the other two methods precisely because it is designed to handle variations in texture scale and orientation.

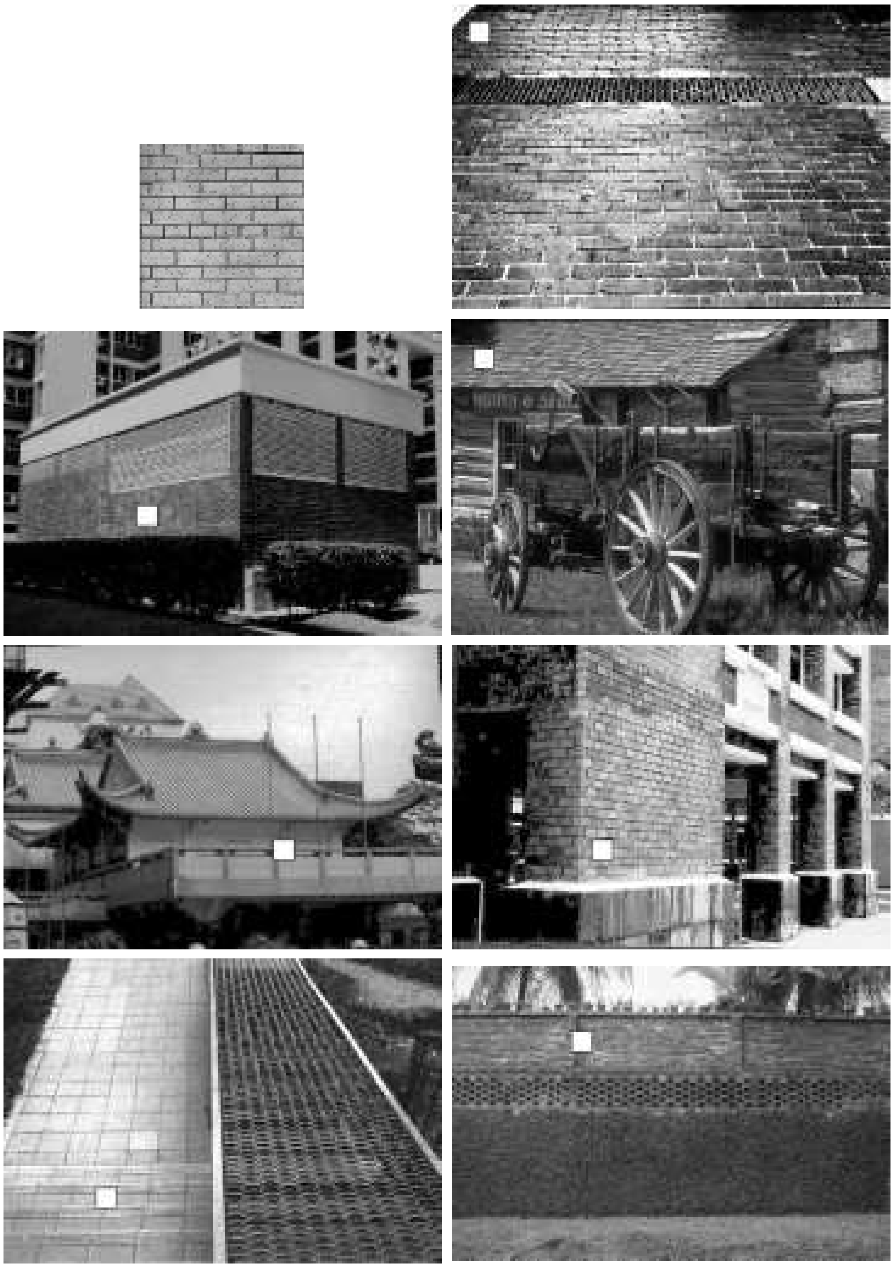


Figure 6. Sample natural scene images retrieved by the invariant matching method in decreasing order of similarity. The first image is the query texture. The white boxes in the images denote the regions that best match the query texture.

5 Conclusion

Texture matching for content-based image retrieval is a very difficult task, especially for retrieving natural scene images. These images can contain multiple texture patterns that vary in intensity due to different lighting conditions, and differ in scale and orientation due to perspective distortion. This article presented a texture matching method that is invariant to texture intensity, scale, and orientation. It extracts texture feature vectors using Gabor filters, transforms the feature vectors into an intensity-, scale-, and orientation-invariant texture space, compares texture similarity in the texture space, and refines texture ranking using the original Gabor feature vectors. Experimental results show that the invariant texture matching method outperforms existing methods in retrieving both Brodatz images and, especially, natural scene images.

Acknowledgement

This research is supported by NUS research grant RP950656.

References

1. H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Trans. on SMC*, 8(6):460–473, 1978.
2. J.C. Mao and A.K. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25:173–188, 1992.
3. G.R. Cross and A.K. Jain. Markov random field texture models. *IEEE Trans. on PAMI*, 5:25–39, 1983.
4. J. You and H. A. Cohen. Classification and segmentation of rotated and scaled textured images using texture "tuned" masks. *Pattern Recognition*, 26(2):245–258, 1993.
5. F. S. Cohen, Z. Fan, and M. A. Patel. Classification of rotated and scaled textured images using Gaussian Markov random field models. *IEEE Trans. on PAMI*, 13(2):192–202, 1991.
6. F. Liu and R.W. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Trans. on PAMI*, 18(7):722–733, July, 1996.
7. J. Ashley, R. Barber, M. Flickner, J.L. Hafner, D. Lee, W. Niblack, and D. Petkovic. Automatic and semiautomatic methods for image annotation and retrieval in QBIC. In *Proceedings of SPIE Storage and Retrieval*

for Image and Video Databases III, pages 24–35, Feb 1995.
8. A.C. Bovik, M. Clark, and W.S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Trans. on PAMI*, 12(1):55–73, 1990.
9. J. Bigün and J. M. H. Du Buf. *N*-folded symmetries by complex moments in Gabor space and their application to unsupervised texture segmentation. *IEEE Trans. on PAMI*, 16(1):80–87, 1994.
10. D. Dunn, W. E. Higgins, and J. Wakeley. Texture segmentation using 2-D Gabor elementary functions. *IEEE Trans. on PAMI*, 16(2):130–149, 1994.
11. A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.
12. W.Y. Ma and B.S. Manjunath. NeTra: A toolbox for navigating large image databases. In *IEEE Conference on Image Processing*, volume 1, pages 568–571, Oct 1997.
13. G.E. Hinton, J.L. McClelland, and D.E. Rumelhart. Distributed representation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing*. MIT Press, Cambridge, Massachusetts, 1986.
14. A.R. Rao and G.L. Lohse. Towards a texture naming system: Identifying relevant dimensions of texture. *IEEE Conference on Visualization*, pages 220–227, Oct 1993.
15. N. Zhang. Invariant texture segmentation for images of natural scene. Master's thesis, Dept. of Information Systems and Computer Science, National University of Singapore, 1997.
16. W.Y. Ma and B.S. Manjunath. Texture features and learning similarity. In *Proc. of IEEE CVPR*, pages 1160–1169, June, 1996.