Extracting Texture Features from Arbitrary-shaped Regions for Image Retrieval

Ying Liu
School of ITEE,
The University of Queensland,
QLD, 4072, Australia
e_liuy@yahoo.com

Xiaofang Zhou
School of ITEE,
The University of Queensland,
QLD, 4072, Australia
zxf@itee.uq.edu.au

Wei-Ying Ma
Microsoft Research Asia,
5F, No. 49 Zhichun Road,
Beijing, 100080, China
wyma@microsoft.com

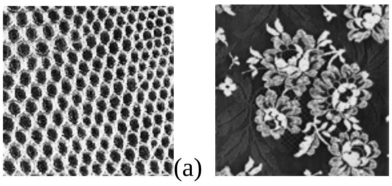
Abstract

Lots of work has been done in texture feature extraction for rectangular images, but not as much attention has been paid to the arbitrary-shaped regions available in region-based image retrieval (RBIR) systems. In this paper, we present a texture feature extraction algorithm based on Projection Onto Convex Sets (POCS) theory. POCS iteratively concentrates more and more energy into the selected coefficients from which texture feature of an arbitrary-shaped region can be extracted. Experimental results demonstrate the effectiveness of the proposed algorithm for image retrieval purposes.

1. Introduction

Texture plays an important role in human vision and is important in image classification. Lots of work has been done in efficient texture feature extraction from rectangular images. Texture features extracted using, for example, Discrete Wavelet Transform (DWT) [1], Gabor [2], have been proved to be efficient in texture description. However, it is not unusual that the query image or the database images are of arbitrary-shape, such as in a region-based image retrieval system. So far, there is no paper specifically focuses on texture featuring of arbitrary-shaped regions.

Textures can be structured with repetitive patterns, as in Figure 1(a), or non-structured as in Figure 1(b) with no obvious pattern. To efficiently describe such texture as in Figure 1(b), information from the entire region (or as large a region as possible) shall be considered. Hence, the intuitive way of finding an inner rectangular block (IRB) from a region, then applying traditional texture feature extraction algorithms, is not suitable. The reason is that it is difficult to find an IRB as large as possible due to the various shapes. In [4], DWT domain texture features of all the 4*4 blocks in a region are calculated and the mean of these features is used as the region feature (referred to as 'Ave'). The problem is that the average feature of small blocks can't sufficiently describe the texture property of an entire region.



(a) (b)

Figure 1. structured texture, less-homogeneous texture(Brodatz textures[3], d36 and d42)

To extract texture feature from the entire region, a direct way is to extend the region to a rectangular area by padding zero outside the boundary and then applying transformations such as DWT or Gabor. Texture feature can be calculated using the coefficients obtained. We refer to this method as 'RCT' (region coefficient taking). However, zero padding introduces spurious frequency components and the corresponding coefficients will degrade the retrieval performance of the texture feature obtained. To relieve this problem, this paper presents a texture feature extraction algorithm based on POCS [5] theory which can select a set of coefficients best describing a region from a superset of coefficients available. Experimental results prove it to be effective in describing arbitrarily-shaped regions for image retrieval.

The remainder of the paper is organized as follows. Section 2 describes the proposed texture feature extraction algorithm. Section 3 provides experimental results. Finally, Section 4 concludes this paper.

2. The proposed algorithm

2.1. Problem description

2D-DWT decomposes an image into different frequency subbands. Mean and variance of the coefficients in each subband can be calculated as texture feature [1]. Given an arbitrary-shaped region $f(x,y),(x,y)\in A$ containing M pixels, with A being the region interior and the boundary. To apply 2D-DWT, we first extend it into a rectangular block L of size N (>M) enclosing the region interior by padding some values outside the boundary. Then, the region can be approximated by a set of N 2D-DWT basis functions defined on L as

$$g(x,y)=\sum_{(k,l)\in K_v} c_{kl}\varphi_{kl}(x,y),\ (x,y)\in L \qquad (1)$$

where K_v denotes the set of basis function indices used in the expansion for $g(x,y)$, c_{kl} is the coefficient corresponding to basis function φ_{kl} .

As zero-padding in 'RCT' introduces spurious high frequency components, including such components into texture feature extraction will certainly degrade retrieval performance. The problem then is among the superset of N basis functions, how to select the basis best representing the region and obtain the corresponding coefficients from which texture feature can be extracted. POCS can be used to answer this question.

2.2. POCS in texture description

H.H.Chen proposed an algorithm based on POCS [6], in which a given image segment is approximated using 2D-DCT basis functions. We extend it by using 2D-DWT (denoted as 'POCS-DWT') for the purpose of feature extraction from arbitrary-shaped regions. Other transforms such as Gabor can be applied as well, we choose DWT due to its efficiency in texture classification and fast implementation available.

'POCS-DWT' defines two convex sets. The set of images represented by a selected group of coefficients constitutes the first set, and is referred to as Selected Coefficient Set (SCS).

$$SCS = \{g \mid G(i,j)=0\},\ (i,j)\in I \qquad (2)$$

where G is the transform coefficients set and I the discarded group of transform coefficients. This can be obtained by performing the block transform, zeroing the coefficients in I and retaining the rest.

The projection of an arbitrarily-shaped region onto the second set can be obtained by replacing the interior pixels of $g(x,y)$ with the original interior pixel values of $f(x,y)$, and is referred to as the Region of Support Set (RSS).

$$RSS = \{g \mid g(m,n)=f(m,n)\},\ (m,n)\in A \qquad (3)$$

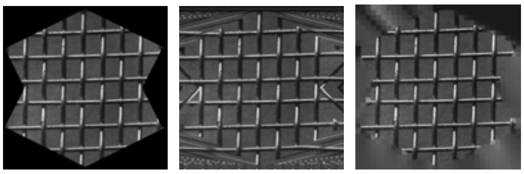
Given the two convex sets SCS and RSS, 'POCS-DWT' finds an image which lies in the intersection of the two sets. The algorithm comprises of two parts. The first part is 'initial padding'. The second part involves a POCS iteration loop. It projects the initial estimate onto either one of the two sets SCS and RSS, and iteratively mapping it back and forth to the other set. The iteration terminates when the difference between $g(x,y)$ obtained from different iteration is below a given threshold. The convergence of the iteration is guaranteed [5]. Finally, the reconstructed region is obtained

$$\hat{f}(m,n)=g(m,n),\ (m,n)\in A \qquad (4)$$

Different initial padding techniques can be used. Zero padding (ZR) is the simplest. However, it introduces discontinuities at the region boundary, yielding spurious high frequency coefficients that will degrade performance of the texture feature obtained. To relieve this problem, we investigate two different padding techniques: a simple mirroring (MR) padding, and an Object-based Padding (OBP) [7] which provides smooth extrapolation.

MR extends the region with its 'mirror image' outside the boundary. Usually, the support of the region is arbitrary with respect to the extended rectangle, and the mirroring may have to be done several times up to the rectangle boundary.

OBP has the following 4 steps: i)The region A is first extended over the rectangular block L by zero-padding. ii) L is divided into 8*8 or 16*16 blocks, and each block is classified into one of the 3 types: (1) all pixels are in A; (2) some pixels belong to A; (3) all pixels do not belong to A. iii) Only blocks in the above case (2) and (3) need padding. For case (2), pixels that do not belong to A are replaced with the mean value of the pixels that are in A. In case (3), the pixel values are computed by referring to the adjacent blocks around the current block. If no block that has already been padded has been found among the adjacent blocks, the current block is skipped and operation is moved to the next block. This process proceeds from top-left to bottom-right in L and is repeated until all blocks are padded. iv) Finally, the entire padded region is low-pass filtered to reduce discontinuities because there are some block discontinuities between padded blocks. The extrapolation result of OBP is smooth. Figure 2 gives an example.



(a) (b) (c)

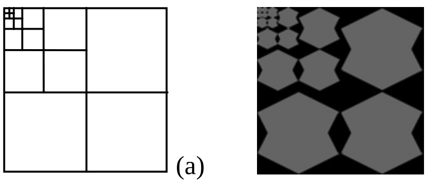
Figure 2. (a) ZR and (b) MR (C) OBP

In this paper, we refer to 'POCS' with 'ZR', 'MR' and 'OBP' as 'PZR', 'PMR' and 'POBP', respectively. The algorithm converges after about 3~5 iterations for texture description. It is shown that the region approximation performance of ‘POBP’, ‘PMR’ is better than ‘PZR’ and ‘RCT’ with higher PSNR. For example, for the region in Figure 2, the PSNR values of the reconstructed region are 36.55dB, 65.64dB, 68.56dB, 84.81dB respectively using ‘RCT’, ’PZR’, ‘PMR’ and ‘POBP’.

Other transformations such as Gabor can be applied to ‘POCS’ as well, we choose DWT due to its efficiency in texture classification and fast implementation available.

2.3. Texture features

A 5-scale 2D-DWT(Db4) produces 16 subbands as in Figure 3(a). The coefficients used to compute texture features are taken from a coefficient selection mask (CSMask) related to the region shape. The CSMask is obtained by iteratively down-sampling the mask of the original region at each decomposition scale. Figure 3(b) gives an example. Mean and variance are calculated from each subband to form the texture feature.



(a) (b)

Figure 3. (a) The 15 subbands in DWT domain (b) Coefficient selection mask example

3. Experimental results

In our experiments, three test data sets are used. The first contains 448 irregular-shaped textures created from the 112 Brodatz textures [3], referred to as ‘DB1’. Examples are given in Figure 4. The second is ‘DB2’, obtained by cutting texture tiles from each of the 112 Brodatz textures. In the third database ‘DB3’, half of the textures are rectangular textures from ‘DB2’ and the other half from ‘DB1’.

For each database, 112 queries are performed and the average recall and precision are computed to measure retrieval performance. Recall indicates the proportion of similar textures retrieved from the database for a query and Precision the proportion of the retrieved textures that are similar to the query. For each query region, there are 4 tiles in the database similar to it.

Firstly, we compare the texture retrieval performance of ‘Ave’, ’RCT’, ‘PZR’ and ‘POBP’ using ‘DB1’ (for clarity concern, results of ‘PMR’ which is very close to ‘POBP’ is not shown here). With ‘k’ being the number of textures retrieved, experimental results in Figure 5 show that, 1) the performance of ‘PZR’, ‘RCT’, ‘PMR’ and ‘POBP’ are much better than that of ‘Ave’. This proves that the average feature of small blocks can’t sufficiently describe the property of an entire region. 2) Performance of ‘PZR’ is close to that of ‘RCT’, while ‘POBP’ and ‘PMR’ perform better. For example, when k=4, recall of ‘RCT’, ‘PZR’, ‘PMR’ and ‘POBP’ are 0.682, 0.690, 0.73, and 0.736, respectively.

Secondly, we test the retrieval performance of ‘PZR’, ‘PMR’ and ‘POBP’ on rectangular textures in ‘DB2’. The performance improvement brought by ‘POBP’ and ‘PMR’ is significant as shown in Figure 6. For instance, when k=10, recall of ‘PZR’, ‘PMR’, ‘POBP’ are 0.576, 0.891 and 0.873 respectively.

Applied to ‘DB3’ which contains both rectangular and arbitrary-shaped textures, again it is shown that the retrieval performance of ‘POBP’ and ‘PMR’ is much better than that of ‘PZR’, as shown in Figure 7.

Experimental results prove that ‘POBP’ and ‘PMR’ can well describe the texture property of arbitrary-shaped regions, with the performance of ‘PMR’ marginally better. From Figure 5, Figure 6 and Figure 7, we also observed that with arbitrary-shaped region as query, the advantage of ‘POBP’ and ‘PMR’ over ‘PZR’ and ‘RCT’ is more obvious when they are applied to database including nice texture features extracted from rectangular textures.

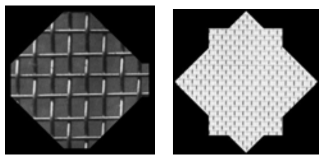


Figure 4. Examples of arbitrary-shaped textures

4. Conclusions

This paper presented a texture feature extraction algorithm based on POCS theory for arbitrary-shaped regions. Mirroring padding and an object-based initial padding technique providing smooth extrapolation are used to relieve the spurious high frequency components as introduced by zero padding. Experimental results prove the algorithm to be effective in describing arbitrary-shaped regions for image retrieval purposes.

5. References

[1] John R. Smith, Shih-Fu Chang, “Transform Features for Texture Classification and Discrimination in Large Image Databases,” Proc. IEEE Inter. Conf. on Image Processing, vol. 3, pp.407-411, 1994.

[2] B. S. Manjunath, W. Y. Ma,“Texture Features for Browsing and Retrieval of Large Image Data,” IEEE Trans. On Pattern Anal. and Mach. Intell. vol.18, no. 8, pp.837-842,1996.

[3] P. Brodatz, Textures: a Photographic Album for Artists & Designers, New York: Dover, New York, NY, 1966.

[4] Jia Li, James Z. Wang and Gio Wiederhold, “IRM: Integrated Region Matching for Image Retrieval,” Proc. of 8th ACM Inter. Conf. on Multimedia :147-156, Oct. 2000.
[5] D.C. Youla, Mathematical Theory of Image Restoration by the Method of Convex Projections, H.Stark, editor, Image Recovery: Theory and Applications. Academic Press, 1987.
[6] H.H.Chen, M.R.Civanlar, and B.G.Haskell, “A Block Transform Coder for Arbitrarily Shaped Image Segments,” Proc. Int. Conf. Image Processing, Austin, Texas, pp85-89, 1994.
[7]Raj Talluri, et.al., “A Robust,Scalable, Object-Based Video Compression Technique for Very Low Bit-Rate Coding”. IEEE Trans. on Circuits and Systems for Video Technology, Vol 7, No.1, pp221-233, Feb. 1997.

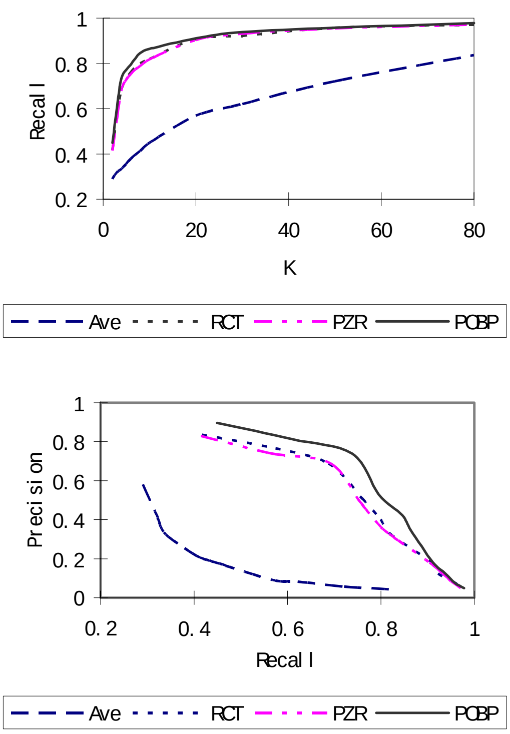
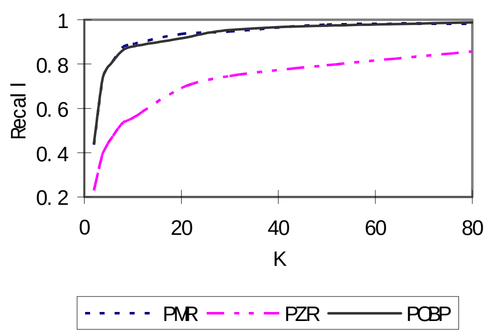


Figure 5. Performance comparison using ‘DB1’



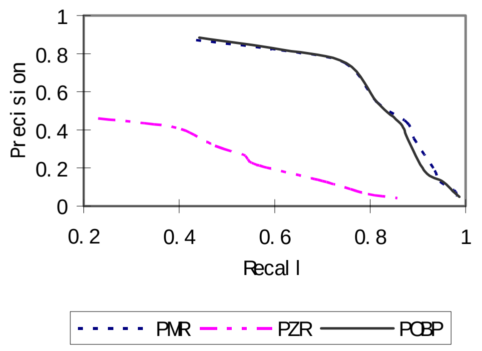


Figure 6. Performance comparison using ‘DB2’

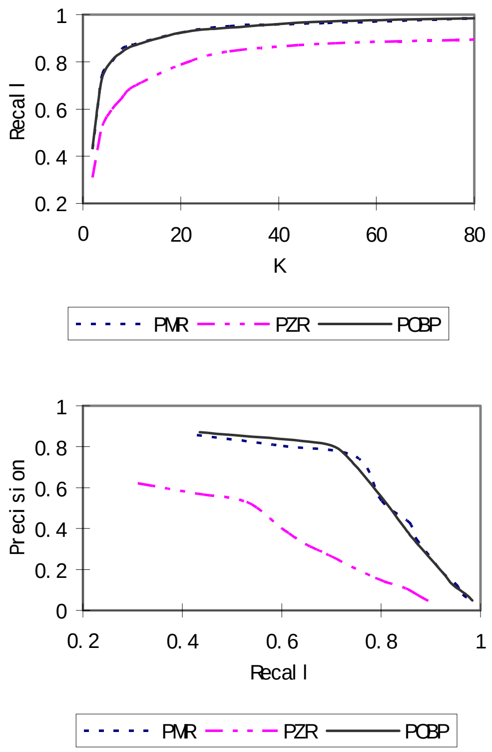


Figure 7. Performance comparison using ‘DB3’