SYMMETRY FEATURE IN CONTENT-BASED IMAGE RETRIEVAL*

Jingrui He[1], Mingjing Li[2], Hong-Jiang Zhang[2], Changshui Zhang[3]

[1,3]Department of Automation, Tsinghua University, Beijing 100084, China
[2]Microsoft Research Asia, 49 Zhichun Road, Beijing 100080, China
[1]hejingrui98@mails.tsinghua.edu.cn, [2]{mjli, hjzhang}@microsoft.com, [3]zcs@tsinghua.edu.cn

ABSTRACT

In this paper, we first apply the theory of wallpaper groups to natural images and extract a novel feature to depict the symmetry property of natural images. The original proposed algorithm takes autocorrelation and correlation as a preprocessing step, which is very time-consuming. Through further analysis, we develop a set of schemes to accelerate this algorithm. Experimental results demonstrate that in performing content-based image retrieval, the proposed symmetry feature outperforms wavelet feature, which is a widely accepted descriptor of texture, and water-filling feature. The accelerated version of the algorithm improves the processing speed by a large margin while it brings little degradation to retrieval performance.

1. INTRODUCTION

In the past few decades, content-based image retrieval (CBIR) has been an active research topic as a result of an explosively growing volume of digital images. One of the basic problems in CBIR is to extract proper features to represent the contents of the original images. Many kinds of features have been proposed and investigated, including color, texture, shape and structure. All of them contribute to the improvement of performance in CBIR systems. For example, to make use of color information, Swain and Ballard [10] proposed using color histograms as image representation; Markus et al [9] first introduced color moment feature for indexing. By incorporating spatial correlations of colors, correlogram feature proposed by Huang et al [4] tends to achieve better retrieval performance than the above two features. Color coherence vector proposed by Pass et al [7] is another efficient feature which combines color and spatial information. To depict texture property, Jacobs et al [5] and Wang et al [11] applied Haar wavelet and Daubechies wavelet transforms to the original images respectively and compared their corresponding coefficients as the similarity measure. In [13, 14], Zhou et al extract structural features through the water-filling algorithm, hoping to obtain salient edges and structure information.

As is mentioned in [6], symmetry is pervasive in both natural and man-made environments. It also plays a non-trivial role in human recognition. For example, Reisfeld et al [8] propose a robust facial feature detector based on a generalized symmetry interest operator; Gauch et al [1] present a shape-based image segmentation method using intensity axis of symmetry (IAS). According to the theory of wallpaper groups [2], the infinite variety of symmetry patterns can be well categorized into seventeen Crystallographic groups. This theory has already been applied in Chemistry, analyzing molecules and crystal, etc [12]. However, since strictly symmetrical patterns rarely appear in natural images, there is no research applying this theory to depict natural image contents as far as we know.

Based on the theory of wallpaper groups, in this paper, we first propose an algorithm to extract symmetry features on multi-scale for natural images. The original algorithm takes autocorrelation and correlation as a preprocessing step, thus it is very time-consuming. Through further analysis, we accelerate the algorithm and make it suitable for practical use. To demonstrate the effectiveness of the proposed symmetry feature, we compare it with wavelet and water-filling features on a general purpose image database including 5,000 Corel images. The experimental results are very promising.

The rest of the paper is organized as follows. In section 2, we propose the symmetry feature for natural images. In section 3, we discuss the acceleration scheme. Experimental results are presented in section 4. Finally, we conclude the paper in section 5.

2. THE SYMMETRY FEATURE FOR NATURAL IMAGES

2.1. Symmetry feature

* This work was performed at Microsoft Research Asia.

According to the theory of wallpaper groups [2], there are exactly seventeen different plane symmetry groups, which are characterized by four distinct kinds of planar symmetry, namely translation symmetry, rotation symmetry, reflection symmetry and glide reflection symmetry. The theory also guarantees that rotation symmetry can only be 2-fold, 3-fold, 4-fold and 6-fold, where k -fold means $360/k$ $(k=2,3,4,6)$ degrees rotation. To determine the symmetry type of a 2D repeated pattern, Liu et al [6] first apply the symmetry to be tested to the entire pattern, then check the similarity between the original and transformed images. In [3], we generalize this idea by comparing the correlation between the original and transformed images with the autocorrelation of the original image in terms of translation vectors. However, the above two pieces of work can only deal with man-made repeated patterns.

In this paper, for the first time we try to extend our idea in [3] to natural images. To make symmetry property explicit for natural images, we first divide the images into blocks and examine their symmetry property. When dealing with natural images, we only focus on rotation symmetry. The reason lies in the fact that reflection symmetry and glide reflection symmetry cannot be correctly identified if there is no information about the reflection axis, which is just the case when we deal with natural images. On the other hand, translation symmetry can only be identified when there are more than four identical repeated units in the neighborhood which rarely appears in natural images.

2.2. Feature extraction algorithm

To fully describe local details, we divide a given image into small blocks. Neighboring blocks have half of their area overlapping to avoid unreasonable segmentation. Like in [3], to determine the presence of a certain kind of rotation symmetry, we first rotate the original block and compute its correlation with the transformed block, then compare the result with the autocorrelation of the original block. However, the comparison cannot be done based on translation vectors due to the lack of repeated units. In this case, we compare the maximum values of correlation and autocorrelation instead, which can be interpreted intuitively as follows. If a block has a certain kind of rotation symmetry, the rotated block will exhibit similar appearance as the original one, thus the maximum value of correlation, which corresponds to a good match between the two blocks, approximates that of autocorrelation. To be specific, let a_m and c_m denote the maximum values of autocorrelation and correlation respectively, i.e. $a_m = \max_{i,j} a(i,j)$, $c_m = \max_{i,j} c(i,j)$, where $a(i,j)$ and $c(i,j)$ denote autocorrelation and correlation values at the point (i,j) . Furthermore, to obviate the influence of different area involved in computing autocorrelation and correlation, we normalize a_m and c_m in the following way: $\hat{a}_m = a_m / area_0$, $\hat{c}_m = c_m / area_c$, where $area_0 = h \cdot w$ is the size of the original block with height h and width w , and $area_c$ is the overlapping area between the two blocks with their centers in superposition. Thus the symmetry measure of a block can be defined as the ratio between the two maximum values:

$$s_k = \frac{\hat{c}_m}{\hat{a}_m} \tag{1}$$

where k can be 2, 3, 4, or 6, corresponding to k -fold rotation symmetry. The larger s_k is, the more likely this block has k -fold rotation symmetry.

Once we have obtained all four symmetry measures for each block, some statistics can be computed to represent the whole image. In our work, we only calculate the first and second order moments for each kind of rotation symmetry, thus get an 8-dimensional feature for the given image.

One key factor of the above feature extraction scheme is the size of each block, which should balance between symmetry formation and locality. It is unwise to fix this parameter due to the unpredicted variation in scale. Therefore, we vary the block size and calculate symmetry measure on multi-scale. In this paper, we vary the block size in $T=3$ levels, i.e. 16×16 , 32×32 and 64×64 , where the block size in the higher level is 4 times as large as that in the lower level. Let $v(t)$ denote the 8-dimensional feature at the tth level $(t=1,2,3)$, the symmetry feature used to represent the image can be written as: $v=[v^T(1), v^T(2), v^T(3)]^T$, and it has 24 dimensions altogether.

3. ACCELERATION SCHEME

The feature extraction algorithm described in section 2 takes autocorrelation and correlation as a preprocessing step, which is very time-consuming, especially when the size of blocks is large. In this section, we have developed several schemes to accelerate the original algorithm, including: 1) the maximum value of autocorrelation can be directly calculated using gray values; 2) correlation between the original block and the rotated one can be accelerated by means of Fast Fourier Transform (FFT); 3) the maximum values of both autocorrelation and correlation at the higher level can be approximated using those at the lower level.

Firstly, it is easily understood that the maximum value of autocorrelation must appear at the central point of a given block, which can be written in the following form:

$$a_m = a_{h/2,w/2} = \sum_{i,j} (p(i,j) \cdot p(i,j)) \tag{2}$$

where $p(i,j)$ is the gray value at the point (i,j) in the original block. If we stack each column of the points within the block into an $(h \cdot w)$ -dimensional vector, $a_{h/2,w/2}$ can be viewed as the inner product between this vector and itself. Meanwhile, $a_{i,j}$ where $i \neq h/2$ or $j \neq w/2$ can be viewed as the inner product between the block vector and another one with the same points but different arrangement. Note that the two vectors share equal module. Thus $a_{h/2,w/2} = \max_{i,j} a(i,j)$. Based on this simple rule, we can obtain a_m using Eq.2. instead of calculating the autocorrelation at every point.

When we calculate c_m , there is no such rule that guarantees its location at the central point. To speed up the calculation of correlation, we may take the advantage of FFT. When we put the pixel gray values in an $(h \times w)$ matrix, and get two matrices M_1, M_2 representing the two blocks, the correlation between these two blocks is equal to the convolution of M_1 and $\hat{M}_2$ up to a translation factor, where $\hat{M}_2(i,j) = M_2(h-i, w-j)$. Therefore, we first perform FFT to an image block and its rotated counterpart to get two spectral representations: $F_0(\omega_1, \omega_2)$ and $F_r(\omega_1, \omega_2)$. The correlation result can be obtained by transforming $F = F_0 \cdot F_r^*$ back to space domain, where F_r^* is the complex conjugate of F_r .

Let $s_k^{(t)}$ denote k-fold symmetry measure at the *t*th scale, with the first scale corresponding to block size of 16×16 . Notice that one block $B^{(t)}(i,j)$ in the *t*th scale is composed of four neighboring blocks $B^{(t-1)}(2i+l, 2j+k)$, $(l,k = 0,2)$ in the $(t-1)$th scale. Once we have calculated the maximum values of autocorrelation and correlation for each block at the $(t-1)$th scale, we can approximate the maximum values at the *t*th scale using those at the $(t-1)$th scale. Let $a_m^{(t-1)}(2i+l, 2j+k)$ and $c_m^{(t-1)}(2i+l, 2j+k)$ $(l,k = 0,1)$ denote the maximum values of $B^{(t-1)}(2i+l, 2j+k)$, and $a_m^{(t)}(i,j)$ and $c_m^{(t)}(i,j)$ denote those of $B^{(t)}(i,j)$, then we have:

$$
\begin{aligned}
a_m^{(t)}(i,j) &= \sum_{i,j} (p(i,j) \cdot p(i,j)) \\
&= \sum_{l,k} a_m^{(t-1)}(2i+l, 2j+k); \\
c_m^{(t)}(i,j) &= \sum_{i,j} (p(i,j) \cdot q(i,j)) \\
&\approx \sum_{l,k} c_m^{(t-1)}(2i+l, 2j+k)
\end{aligned}
\tag{3}
$$

where $q(i,j)$ is the gray scale at point (i,j) in the rotated block; $l,k = 0,2$. The first equation is strictly satisfied, while the second one is approximately satisfied due to the rotational effect.

Based on the above preparation, we can summarize the algorithm as follows:

Symmetry feature extraction algorithm

1. Divide the image into 16×16 overlapping blocks; calculate a_m^1 for each block using Eq.2.; calculate c_m^1 by means of FFT;
2. Calculate $v(1)$ using Eq.1.;
3. Repeat for $t = 2,3$:
   - Calculate a_m^t and c_m^t using Eq.3.;
   - Calculate $v(t)$ using Eq.1.;
4. Output symmetry feature for the given image:

$$v = [v^T(1), v^T(2), v^T(3)]^T$$

4. EXPERIMENTAL RESULTS

All our experiments are performed on a general purpose image database. It consists of 5,000 images taken from Corel image database, and can be categorized into 50 groups, each of which contains 100 images. Some of the concept groups are: balloon, dog, model, ship, wolf, etc. Two images belonging to the same group are considered relevant, and vice versa.

4.1. Processing time

The main operation in the original algorithm is autocorrelation and correlation. By means of the three schemes proposed in the previous section, the accelerated version greatly relieves the computation load. To test the effectiveness of the acceleration scheme, we perform both the original algorithm and the accelerated version on a 384×256 image, and list their processing time in table1.

Algorithm	Original	Accelerated
Processing time	13.92 seconds	2.734 seconds

Table 1. Processing time comparison (Intel(R) 1.79GHz, 512M RAM)

4.2. Systematic retrieval results

We build three kinds of features on our image database, i.e. symmetry feature, wavelet feature [11] and water-filling feature [13, 14], and use them for image retrieval respectively. All these features are based on the gray value of pixels, and no color information is utilized. In [11], Wang et al proposed applying Daubechies wavelet transform to the images, and using the coefficients as feature vectors. However, the dimensionality of their feature is very high, especially for large images. In our implementation, we first perform 3-level Daubechies wavelet transform, and then extract the first and second order moments of coefficients at each level, thus get an 18-dimensional representation. Water-filling feature has 18 dimensions, consisting of MaxFillingTime and the associated ForkCount (MFT and FC), MaxForkCount and the associated FillingTime (MFC and FT), FillingTime Histogram (7 bins), and ForkCount Histogram (7 bins).

Since different components in the same kind of feature usually have different ranges of value, we normalize them to [0, 1] to ensure that they have the same impact on the calculation of similarity between images. Throughout the experiments, Euclidian distance is used to measure the similarity. The retrieval results are plotted in Fig.1.

From Fig.1., we can see that symmetry feature outperforms wavelet feature, which is a widely accepted descriptor of texture, and water-filling feature. Furthermore, the symmetry feature obtained from the accelerated algorithm performs almost as well as that obtained from the original one, which demonstrates the effectiveness of our accelerated algorithm.

5. CONCUSION

In this paper, we have proposed a novel feature which depicts the symmetrical property of natural images based on the theory of wallpaper groups. When applied in the field of image retrieval, it outperforms wavelet and water-filling features. Furthermore, we have also developed several schemes to accelerate the algorithm for practical use, which brings almost no degradation to retrieval performance. Future work includes incorporating color information into the proposed symmetry feature to further improve its performance.

6. ACKNOWLEDGEMENTS

This work was supported by National High Technology Research and Development Program of China (863 Program) under contract No.2001AA114190.

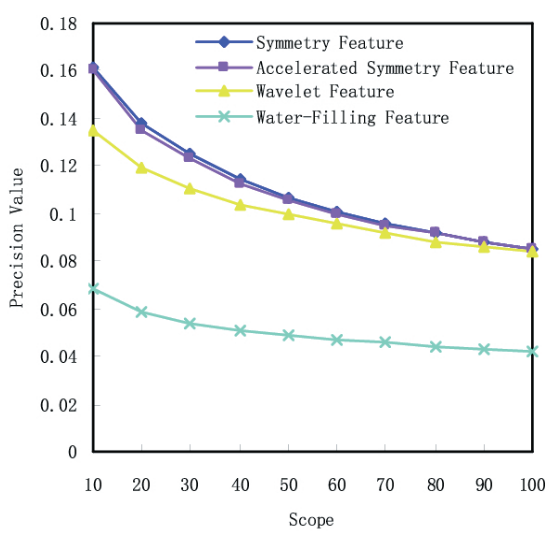

Fig.1. Retrieval performance comparison

7. REFERENCE

[1] J.M. Gauch, and S.M. Pizer, "The intensity axis of symmetry and its application to image segmentation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15:753-770, 1993.

[2] B. Grünbaum, and G.C. Shephard, *Tilings and patterns*, W.H. Freeman and Company, 1987.

[3] J.R. He, M.J. Li, H.J. Zhang, and C.S. Zhang, "Automatic peak number detection in image symmetry analysis", *submitted to PCM'2004*, 2004.

[4] J. Huang, S.R. Kumar, M. Mitra, W.J. Zhu, and R. Zabih, "Image indexing using color correlograms", *CVPR*, 762-768, 1997.

[5] C.E. Jacobs, A. Finkelstein, and D.H. Salesin, "Fast multiresolution image querying", *Proc. SIGGAPH'95*, 277-286, 1995.

[6] Y. Liu, and R. Collins, "A computational model for repeated pattern perception using Frieze and Wallpaper groups", *CVPR*, 537-544, 2000.

[7] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors", *ACM Multimedia*, 65-73, 1996.

[8] D. Reisfeld, and Y. Yeshurun, "Robust detection of facial features by generalized symmetry", *Proc. IEEE Int. Conf. on Pattern Recognition,* 117-120, 1992.

[9] M. Stricker, and M. Orengo, "Similarity of color images", *Storage and Retrieval for Image and Video Databases*, 381-392, 1995.

[10] M.J. Swain, and D.H. Ballard, "Indexing via color histograms", *Proc. Third Int. Conf. on Computer Vision*, 390-393, 1990.

[11] J.Z. Wang, G. Wiederhold, O. Firschein, and S.X. Wei, "Content-based image indexing and searching using Daubechies' wavelets", *Int. J. on Digital Libraries*, 1:311-328, 1997.

[12] H. Zabrodsky, S. Peleg, and D. Avnir, "Continuous symmetry measures", *J. American Chem.* Soc., 115:8278-8289, 1992.

[13] X.S. Zhou, and T.S. Huang, "Image retrieval: feature primitives, feature representation, and relevance feedback", *CBAIVL'2000, in conjunction with IEEE CVPR'2000*, 2000.

[14] X.S. Zhou, Y. Rui, and T.S. Huang, "Water-Filling: a novel way for image structural feature extraction", *ICIP'99*, 1999.