Self-similarity and Points of Interest in Textured Images

Shripad Kondra[1,2,⋆] and Alfredo Petrosino[2]

[1] Neuroimaging Lab, National Brain Research Centre, India
[2] Department of Applied Science, University of Naples "Parthenope", Italy
alfredo.petrosino@uniparthenope.it

Abstract. We propose the application of symmetry for texture classification. First we propose a feature vector based on the distribution of local bilateral symmetry in textured images. This feature is more effective in classifying a uniform texture versus a non-uniform texture. The feature when used with a texton-based feature improves the classification rate and is tested on 4 texture datasets. Secondly, we also present a global clustering of texture based on symmetry.

1 Introduction

Symmetry is a prevalent perceptive feature for humans. For instance, the human face or body is approximately symmetric, a quality that is exploited to assist in face recognition and facial feature detection. Symmetry is said to be important in perception. Several psychological studies have been performed to show it. Psychologists of the Gestalt school have assigned a relevant role to symmetry in attentive mechanism both in visual and auditory systems [6]. For example, psycho-physical experiments show that infants (1-2 years old) tend to fixate symmetrically around an arbitrary single distinctive stimulus (i.e. corners of a triangle).

Where has symmetry been used? It has been used to find interest points, to determine symmetry axis [14,12,10,9,5]. Classification was done based on global symmetry of the image, but never on the distribution of the local symmetries. Detected symmetry has been used in finding lines in images [4]. Recently a global model based on affine symmetry between structural textures at a local level has been developed in a multiresolution framework for multiscale analysis, by which the self similarity of the image is exploited across space and scale [11].

What is meaningful in textured man-made structures is the property of being symmetrical, like correspondence in size, shape, and relative position of parts on opposite sides of a dividing line or median plane or about a center or axis. Figure 1 shows some examples of man-made structures which have symmetry.

Zavidovique et al. [14] have proven that, in any direction, the optimal symmetry axis corresponds to the maximal correlation of a pattern with its symmetric

⋆ This work was done when Shripad Kondra was at University of Naples.

M.K. Kundu et al. (Eds.): PerMIn 2012, LNCS 7143, pp. 306–313, 2012.

Fig. 1. Symmetry in famous Man-made structures

version. In particular, in this paper we deal with bilateral symmetry, a measure obtained by using correlation with the flipped image around a particular axis.

The paper consists of two parts: the first one about using symmetry for texture classification and the second one about using symmetry for clustering textures. Symmetry can be used to distinguish between texture which are uniform and textures which are complex. We will discuss first about single-scale symmetry distribution and extend it to multi-scale. We shall then mention about a new symmetry based feature vector. The symmetry is extracted from different patches over the image and the feature vector is formed using the symmetry, angle distribution and other patch properties. We used our symmetry based feature in combination with a texton–based method [7] for texture classification on four standard texture datasets. We also present a new feature vector based on symmetry. The symmetry is extracted from different patches over the image and the feature vector is formed using the symmetry, angle distribution and other patch properties.

2 Texture Classification Using Symmetry

2.1 Single Scale Symmetry

Symmetry of a patch around each pixel is found as follows. Any image is resized to have diagonal size of 256 pixels and the patch size is 12×12. The symmetry is calculated by the following steps shown in Algorithm 1.

Input: Set of points
Output: Symmetry distribution
foreach 12×12 *Patch around a point* i **do**
    foreach *Rotation theta* **do**
        1. Normalize the patch;
        2. Find the correlation between the rotated patch and its reflected patch around y–axis;
    end
    Find the maximum symmetry value and corresponding angle;
end

Algorithm 1: Calculation of single scale symmetry distribution

The range of the symmetry value is between 0 and 1, with 1 as maximum symmetry. In order to use symmetry to represent texture, it is necessary to see if the symmetry distribution for an image is rotation invariant. To see this we tested the symmetry distribution of some images with their rotated versions. This can be seen from the following figures. The symmetry is obtained as the maximum over 8 different patch rotations. Figure 2 shows the distribution of symmetry and the corresponding angle at which maximum symmetry is detected. The angle distribution is further sorted to make it rotation invariant.

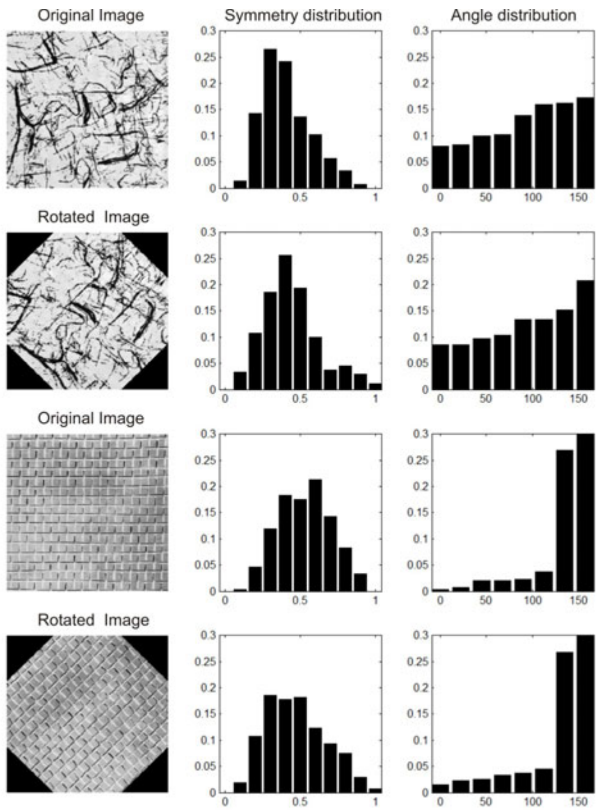


Fig. 2. Single scale symmetry distributions for some textures

2.2 The New Multiscale Feature Vector

Firstly, the image is whitened in the range $[0, 1]$ by removing the mean. Next for every point in the image, the following four different parameters are extracted from three different patch sizes (12×12, 24×24, 36×36).

1. Multiscale symmetry for the patch at three scales
2. The maximum direction for the symmetry along each of the three scales

3. The mean intensity of the patch for each scale
4. The entropy of the patch for each scale (calculated in terms of dit (log10))

Two feature vectors are then constructed:

1. One for the points having mean intensity <0 and
2. Another having mean intensity>=0.

Each feature vector consists of

1. Distribution of symmetry with 11 histogram bins in the range [0, 1], with bin width 0.1.
2. Sorted distribution of symmetry directions. (14 different directions are used)
3. Distribution of entropy with 5 bins in the range [0, 0.6], with bin width 0.15

Thus, the dimension of feature vector is $3 \times (11 + 14 + 5) \times 2 = 180$ (3 is the number of scales). The details of the feature vector are shown in figure 3. The parameters such as the histogram bins, number of directions and weight for each feature are selected from experiments on a small subset of Brodatz texture dataset.

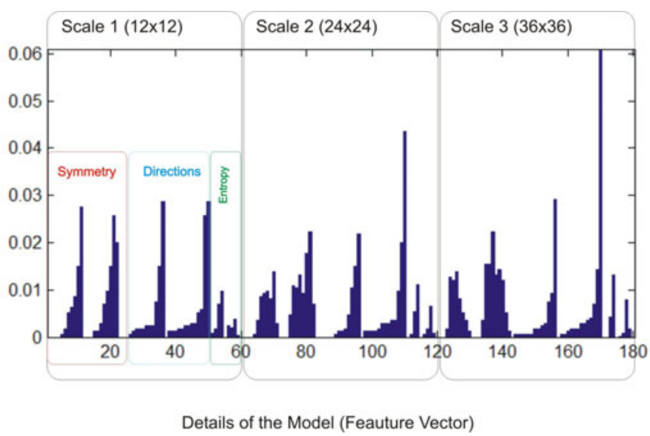


Fig. 3. Multiscale scale feature vector used for texture classification

2.3 Image Representation for Textons

For texton representation, we use the method of [7]. The method uses 3 circular filters. Images are first filtered using these filters, then thresholded and averaged over two small neighborhoods (3×3 and 6×6). Eighty universal textons are used for each neighborhood. The feature space is reduced in one neighborhood by grouping into 4 bins. Each image is thus represented by a 2D histogram giving a 320 (80×4) dimensional feature vector (indicated as *Model*).

2.4 Textons and Symmetry Combined

From the experiments done on the small subset of Brodatz texture dataset (see table 1), it was found that the symmetry based feature vector gives alone poor performance as can be seen from the recognition rate. But the performance improve when the feature vector it was combined with the texton distribution [7]. A combined feature vector is formed using the texton and symmetry distribution. For each image we have two representations: one for textons and another for symmetry. The combined model for different images is shown in Figure 4. In the first two cases texton distribution is different but the symmetry distribution is nearly the same. Whilst, in the last two images texton distribution is nearly the same but the symmetry distribution is different.

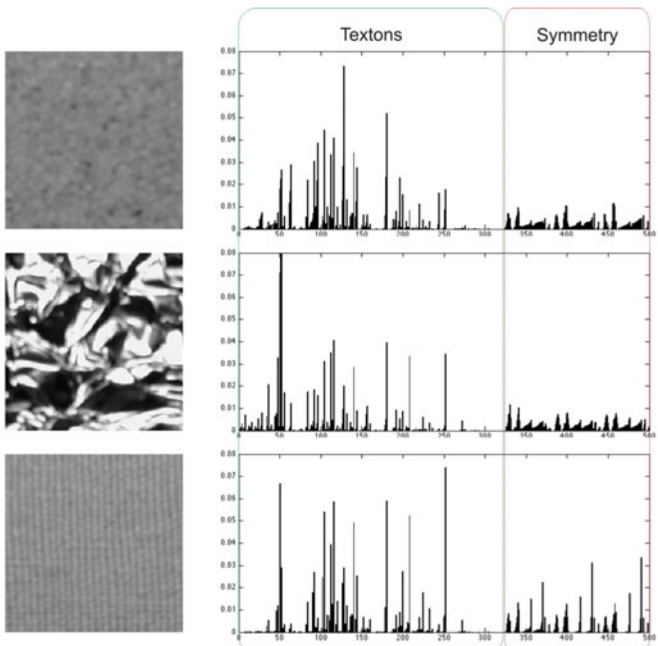


Fig. 4. Combined feature vector Textons plus Symmetry

2.5 Texture Dataset Benchmarking

The experimental setup is the same as mentioned in [7]. The model is trained with SVM using chi-square kernel. The weight of symmetry vector was chosen as 0.4 for this experiment.

The results from Table 1 indicate that symmetry alone is not a good feature for classification. Symmetry can be used to help the classification results between uniform and non-uniform textures. Using the parameters tuned from this small dataset, the recognition was done on four different datasets giving the following

Table 1. Parameter tuning on a small subset of Brodatz dataset

Feature	**Recognition rate (%) on subset of Brodatz dataset**
Symmetry	72.98 ±1.8
Textons [8]	95.97 ± 0.72
Textons + Symmetry	98.27 ± 1.4

results. For the experimental evaluation, four texture datasets were used. The texture datasets are UIUCTex [8], KTH-TIPS [3], Brodatz [1], and CUReT [2].

Table 2. Comparison with other state of art methods

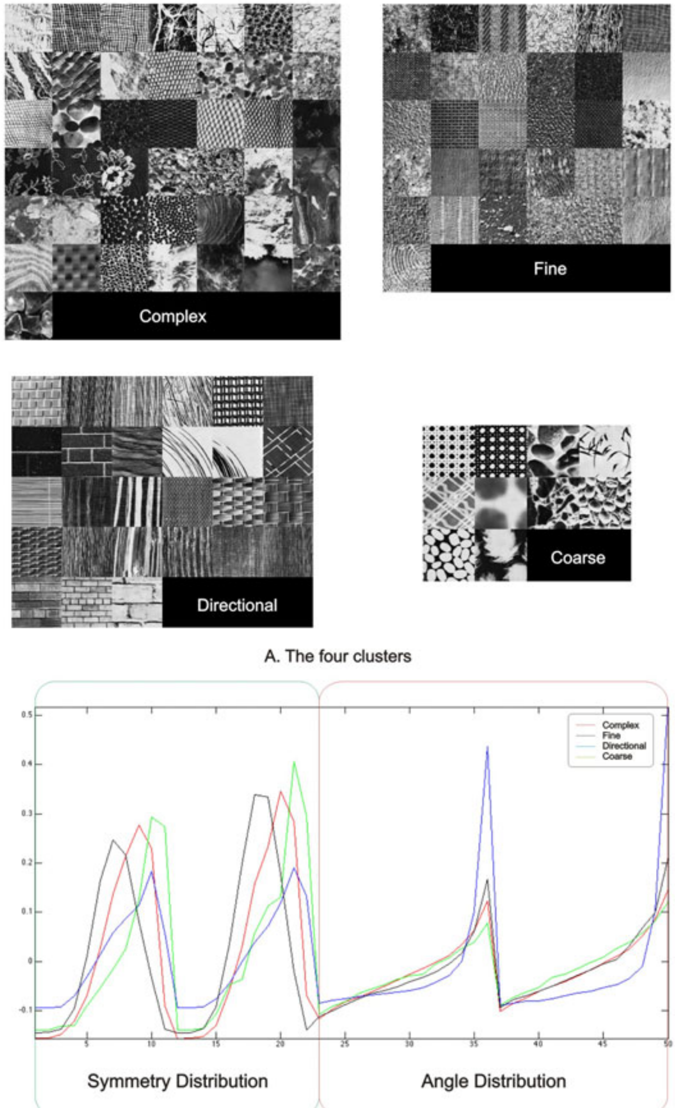
Database	UIUCTex	KTH-TIPS	Brodatz	CUReT
Proposed	96.9 ±0.8	**98.1 ±1.1**	94.0 ±0.9	98.5 ±0.2
Kondra [7]	92.9 ±1.2	97.7 ±0.8	92.3 ±1.0	97.0 ±0.4
Zhang [15]	**98.3 ±0.5**	95.5 ±1.3	**95.4 ±0.3**	95.3 ±0.4
Hayman [3]	92.0 ±1.3	94.8 ±1.2	95.0 ±0.8	**98.6 ±0.2**
VZ-joint [13]	78.4 ±0.9	92.4 ±1.4	92.9 ±1.0	96.0 ±0.7
Lazebnik [8]	96.4 ±2.0	91.3 ±2.1	89.8 ±0.8	72.5 ±0.4
G. Gabor	65.2 ±2.0	90.0 ±2.0	87.9 ±1.0	92.4 ±0.5

The combination of textons and symmetry thus improves the results as can be seen from Table 2. As it can be evidenced (rows 1 and 2 in the table), for UIUCTex dataset the recognition rate increased by 4% considering that UIUCTex dataset has more non-uniform textures.

3 Texture Clustering Using Symmetry

Using only the symmetry and direction vector (single scale of the new feature vector), we clustered the 111 Images in the Brodatz dataset into 4 separate clusters using K–means algorithm with correlation based distance. We named the clusters as Coarse, Directional, Fine, Complex as shown in figure 5. This type of clustering was never done before to our knowledge. The visual meaning captured from the clusters is that symmetry is indeed a human visual phenomenon and there must be a local symmetry evaluation inside the visual system.

Taking into account the mean feature vector of the clusters, it can be seen that the directional textures have high symmetry and the angle distribution has one prominent angle. The other clusters have a wide angle distribution. Fine textures have less symmetry. Complex textures have medium symmetry, while Coarse textures have the highest symmetry.



B. The mean normalized feature of the above four clusters

Fig. 5. Texture clusters

4 Discussion

In this work we have proposed a feature vector based on symmetry. The conducted experiments fix that symmetry is more effective in classifying a uniform texture versus a non-uniform texture; when used with a texton-based feature, symmetry improves the classification rate. We tested the classification performance on four texture datasets and achieved increase in performance over the previous texton based approaches. We also present a global clustering of texture based on the symmetry. The textures are clustered into more meaningful sets, which shows the importance of symmetry in humans. The method may be more suitable for biomedical images where objects of interest are mostly symmetric and images are usually captured under restricted geometric pose. Ongoing work consists into constructing an affine invariant symmetry descriptor. The algorithm reported here can be tweaked to detect multiple reflection and rotation symmetries.

References

1. Brodatz, P.: Textures: A Photographic Album for Artists and Designers. Dover, New York (1996)
2. Dana, K.J., Ginneken, B.V., Nayar, S.K., Koenderink, J.J.: Reflectance and texture of real-world surfaces. ACM Transactions on Graphics (TOG) 18(1), 1–34 (1999)
3. Hayman, E., Caputo, B., Fritz, M., Eklundh, J.-O.: On the Significance of Real-World Conditions for Material Classification. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 253–266. Springer, Heidelberg (2004)
4. Huebner, K.: A Symmetry Operator and Its Application to the RoboCup. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020, pp. 274–283. Springer, Heidelberg (2004)
5. Keller, Y., Shkolnisky, Y.: An algebraic approach to symmetry detection. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, vol. 3 (2004)
6. Koehler, W., Wallach, H.: Figural after-effects: an investigation of visual processes. Proc. Amer. Phil. Soc. 88, 269–357 (1944)
7. Kondra, S., Torre, V.: Texture classification using three circular filters. In: Sixth Indian Conference on Computer Vision, Graphics & Image Processing, ICVGIP 2008, pp. 429–434 (2008)
8. Lazebnik, S., Schmid, C., Ponce, J.: A sparse texture representation using local affine regions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1265–1278 (2005)
9. Loy, G., Eklundh, J.O.: Detecting Symmetry and Symmetric Constellations of Features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 508–521. Springer, Heidelberg (2006)
10. Manmatha, R., Sawhney, H.: Finding symmetry in intensity images. Tech. rep., University of Massachusetts Amherst MA USA (1997)
11. Park, H., Martin, G.R., Bhalerao, A.: An affine symmetric image model and its applications. IEEE Transactions on Image Processing 19(7), 1695–1705 (2010)
12. Stentiford, F.: Attention Based Facial Symmetry Detection. In: Singh, S., Singh, M., Apte, C., Perner, P. (eds.) ICAPR 2005. LNCS, vol. 3687, pp. 112–119. Springer, Heidelberg (2005)
13. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. International Journal of Computer Vision 62(1), 61–81 (2005)
14. Zavidovique, B., Ges, V.D.: The s-kernel: A measure of symmetry of objects. Pattern Recognition 40(3), 839–852 (2007)
15. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. In: 2006 Conference on Computer Vision and Pattern Recognition Workshop, p. 13 (2006)