BINARY GABOR PATTERN: AN EFFICIENT AND ROBUST DESCRIPTOR FOR TEXTURE CLASSIFICATION

Lin Zhang, Zhiqiang Zhou, and Hongyu Li[*]
School of Software Engineering, Tongji University, Shanghai, China

ABSTRACT

In this paper, we present a simple yet efficient and effective multi-resolution approach to gray-scale and rotation invariant texture classification. Given a texture image, we at first convolve it with *J* Gabor filters sharing the same parameters except the parameter of orientation. Then by binarizing the obtained responses, we can get *J* bits at each location. Then, each location can be assigned a unique integer, namely "rotation invariant binary Gabor pattern (BGP_{ri})", formed from *J* bits associated with it using some rule. The classification is based on the image's histogram of its BGP_{ri}s at multiple scales. Using BGP_{ri}, there is no need for a pre-training step to learn a texton dictionary, as required in methods based on clustering such as MR8. Extensive experiments conducted on the CUReT database demonstrate the overall superiority of BGP_{ri} over the other state-of-the-art texture representation methods evaluated. The Matlab source codes are publicly available at http://sse.tongji.edu.cn/linzhang/IQA/BGP/BGP.htm

Index Terms— texture classification, Gabor filter

1. INTRODUCTION

Recent years have witnessed a growing interest in designing effective schemes for texture classification. Most of the methods [1-7] in this field adopt a common two-stage structure. In the first stage, texture images are represented as histograms over a discrete dictionary. In the second stage, the sample texture image will be assigned a class label based on the matching results between its histogram and the model histograms. From such a two-stage structure, we can know that the key issue in texture classification is how to represent texture images properly. Several representative works in this area will be reviewed in the following.

In [1], Ojala *et al*. proposed the "Local Binary Pattern (LBP)". In that method, the dictionary is a set of pre-defined uniform local binary patterns. Given a texture image, each of its location can be assigned with an integer representing a specific local binary pattern. Then the occurrence histogram of the uniform LBPs can be constructed for the image. Recently, Zhang *et al*. [4] extended the traditional LBP to Monogenic-LBP by incorporating other two rotation invariant measures, the local phase and the local surface type extracted using Riesz transforms. In [5], Crosier and Griffin proposed to use basic image features for texture representation. In their method, the dictionary is a set of pre-defined Basic Image Features (BIFs) [8], each corresponding to a qualitatively different type of local geometric structure. It needs to be noted that in methods [1, 4, and 5] there is no need for a pre-training step to learn a dictionary. Varma and Zisserman [6] proposed a statistical learning based algorithm, namely Maximal Response 8 (MR8), using a group of filter banks, where a rotation invariant texton dictionary is build first from a training set and then an unknown texture image is classified according to its histogram of texton frequencies. Later, under the same framework, Varma and Zisserman [7] proposed a new statistical learning based algorithm, in which, instead of responses of filter banks, compact image patches were used directly to represent local patterns.

In this paper, we propose a novel training-free rotation invariant texture representation scheme. Here training-free means that in our method there is no need for a pre-training step to learn a texton dictionary as MR8 does. Our idea is inspired by the success of LBP, such a simple yet powerful texture descriptor. From the definition of LBP it can be known that LBP for a central pixel is totally decided by the signs of differences between it and its neighboring pixels. But, each sign used in LBP is binarized from the difference of two single pixels so it may be sensitive to noise. To improve it, we can use difference between regions to replace difference between two single pixels, which will be more robust intuitively. Gabor filter [9] is an ideal tool to this end, which can calculate the difference between regions covered by its support. In our method, the dictionary is a set of pre-defined rotation invariant binary patterns called as "rotation invariant binary Gabor patterns (BGP_{ri}s)". The occurrence histogram of BGP_{ri}s can be formed to a given image. Then, the classification is based on the matching results between the sample histogram and the model histograms. Experiments are conducted on CUReT database [10, 11] to show the superiority of the proposed BGP_{ri} texture feature extractor.

The rest of this paper is organized as follows. Section 2 discusses the extraction of BGP_{ri}s. Section 3 presents the texture classification scheme using BGP_{ri}s. Section 4 reports the experimental results and Section 5 concludes the paper.

[*] Corresponding author. Email: hyli@tongji.edu.cn.

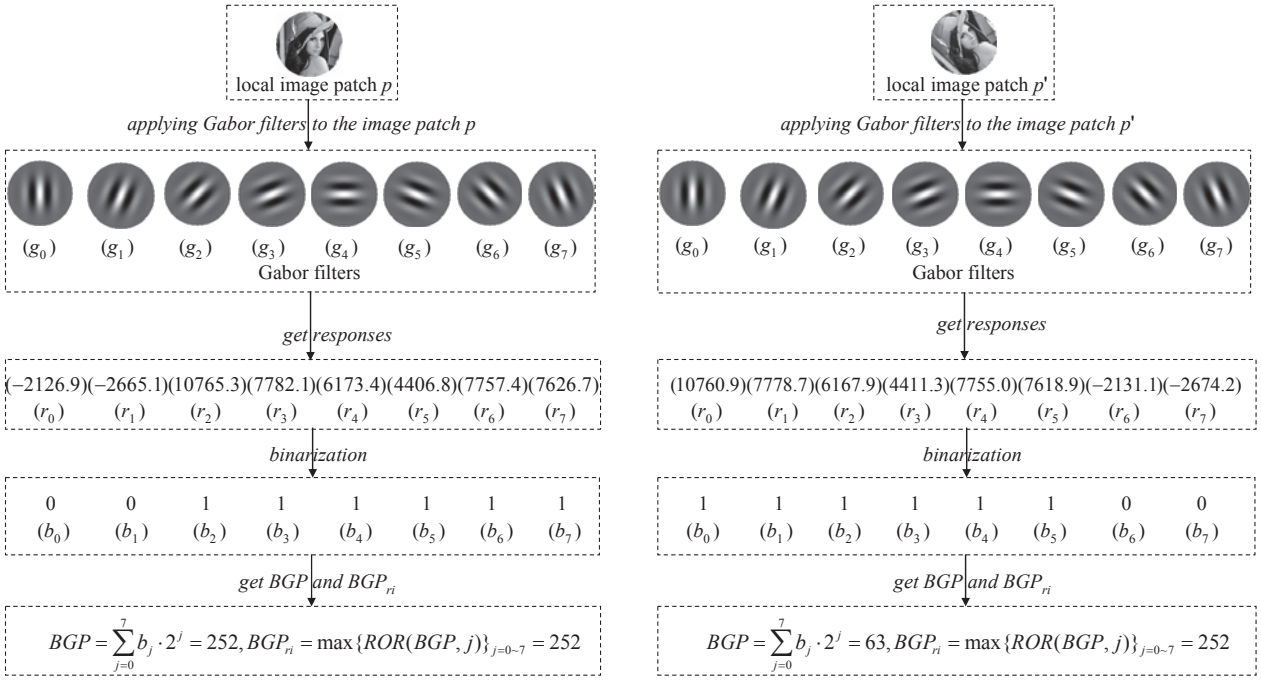

Fig. 1: Illustration for the calculation process of BGP_{ri}s, which consists of three steps, including Gabor filtering, binarization, and rotation invariant coding. The image patch p' used in (b) is a rotated version of p used in (a). BGP_{ri}s derived from p and p' are the same, which validates that BGP_{ri} has the characteristics of rotation invariance.

2. ROTATION INVARIANT BINARY GABOR PATTERNS

Here we present our novel texture feature extractor, namely rotation invariant binary Gabor pattern (BGP_{ri}), in detail.

2.1. Rotation invariant binary Gabor patterns

Let's briefly review Gabor filters at first. 2D Gabor filters are usually expressed as even-symmetric and odd-symmetric ones separately. They are defined as

$$g_e(x,y) = \exp\left(-\frac{1}{2}\left(\frac{x'^2}{\sigma^2}+\frac{y'^2}{(\gamma\sigma)^2}\right)\right)\cos\left(\frac{2\pi}{\lambda}x'\right) \quad (1)$$

$$g_o(x,y) = \exp\left(-\frac{1}{2}\left(\frac{x'^2}{\sigma^2}+\frac{y'^2}{(\gamma\sigma)^2}\right)\right)\sin\left(\frac{2\pi}{\lambda}x'\right) \quad (2)$$

where $x' = x\cos\theta + y\sin\theta$, $y' = -x\sin\theta + y\cos\theta$. λ represents the frequency of the sinusoid factor, θ represents the orientation of the normal to the parallel stripes of the Gabor function, σ is the sigma of the Gaussian envelope and γ is the spatial aspect ratio.

Suppose that $g_0 \sim g_{J-1}$ are J Gabor filters (even-symmetric or odd-symmetric) sharing the same parameters except the parameter for orientation. Their orientations are $\{\theta_j = j\pi / J :| j = 0, 1,\ldots, J\text{-}1\}$. Let R denote the radius of the filter masks used. Now we want to define BGP_{ri} at the location **x** on a given image. Consider a circular image patch p with a radius R centering at **x**. After applying $g_0 \sim g_{J-1}$ to the patch p (that means multiplying p with $g_0 \sim g_{J-1}$ in a point-wise manner and then summing up all the elements), we can get a response vector $r = \{r_j :| j = 0 \sim J\text{-}1\}$. Then by binarizing r, we can get a binary vector $b = \{b_j :| j = 0 \sim J\text{-}1\}$. Each b_j is either 0 or 1. By assigning a binomial factor 2^j for each b_j, we can transform b into a unique binary Gabor pattern (BGP) number that characterize the spatial structure of the local image texture

$$BGP = \sum_{j=0}^{J} b_j \cdot 2^j \quad (3)$$

The BGP operator produces 2^J different output values, corresponding to the 2^J different binary patterns that can be formed by the J elements in b. In order to achieve rotation invariance, i.e., to assign a unique identifier to each rotation invariant binary Gabor pattern, we adopt a similar strategy as used in LBP [1]. We define the "rotation invariant binary Gabor pattern (BGP_{ri})" as

$$BGP_{ri} = \max\{ROR(BGP, j) \mid j = 0,1,...,J-1\} \quad (4)$$

where $ROR(x, j)$ performs a circular bitwise right shift on the J-bit number x j times and the subscript ri means "rotation invariant". For example, if the bits string $b_0 \sim b_7$ has eight bits as 00001010, then $BGP = \sum_{j=0}^{7}(2^j \cdot b_j) = 80$ while $BGP_{ri} = 160$ (that is 00000101_2). If $J = 8$, BGP_{ri}s can have 36 different values, which are listed in Table 1.

We illustrate the calculation process of BGP and BGP_{ri} through an example shown in Fig. 1. To calculate BGP_{ri}, 8 even-symmetric Gabor filters $g_0 \sim g_7$ are utilized. The radius of the filter masks is 101. In Fig. 1a, the circular image

patch p is of the radius 101 and we want to calculate the BGP and BGP_{ri} of its central position. At first, by applying $g_0 \sim g_7$ to p, we can get the responses $r_0 \sim r_7$. Then binarize $r_0 \sim r_7$ to get the binary vector $\{b_j :| j = 0 \sim 7\}$. Based on $\{b_j\}$, we can calculate BGP and BGP_{ri} for the central position of the image patch p according to formulas Eqs. (3) and (4), respectively. In this case $BGP_{ri} = 252$. The image patch p' used in Fig. 1b is rotated from p by 45° anticlockwise. The process to calculate BGP and BGP_{ri} from p' is similar to the process shown in Fig. 1a. We can clearly see that after a rotation, BGPs derived from p and p' are not the same anymore while BGP_{ri}s remain unchanged. This demonstrates that BGP_{ri} can rotation invariantly characterize the spatial structure of a local image patch.

Table 1: 36 unique BGP_{ri}s extracted using 8 Gabor filters

00000000	00000001	00010001	00001001	00000101	00100101
00010101	01010101	00000011	01000011	00100011	00010011
01010011	00110011	00001011	01001011	00101011	00011011
01011011	00000111	01000111	00100111	01100111	00010111
01010111	00110111	01110111	00001111	01001111	00101111
01101111	00011111	01011111	00111111	01111111	11111111

2.2. Multi-resolution analysis

In real applications, a multi-resolution analysis can usually lead to better results. With our BGP_{ri} operator, such a multi-resolution analysis can be easily achieved since the Gabor filter used in BGP_{ri} is inherently an excellent tool for the multi-resolution analysis. To this end, by varying parameters λ and σ, we can have Gabor filters at different resolutions. For each selected resolution, we can have a specific BGP_{ri} operator. The multi-resolution analysis can then be accomplished by combining the information provided by these BGP_{ri} operators at different resolutions.

3. CLASSIFICATION SCHEME

For a fixed resolution, we can have two specific BGP_{ri} operators, one is based on even-symmetric Gabor filters, and the other is based on odd-symmetric ones. If s different resolutions are considered, we will have $2s$ BGP_{ri} operators in total. Using each BGP_{ri} operator, a normalized histogram can be constructed by counting the frequencies of BGP_{ri} responses over the whole image. So altogether we will have $2s$ such kinds of histograms. Then, we can concatenate them together to form a large histogram h, and regard it as the descriptor of the image.

We use the χ^2 distance to measure the dissimilarity of sample and model histograms. Thus, a test sample T will be assigned to the class of model L that minimizes

$$D(T, L) = \sum_{m=1}^{M} (T_m - L_m)^2 / (T_m + L_m) \qquad (5)$$

where M is the number of bins, and T_m (L_m) is the value of the sample (model) histogram at the m^{th} bin.

4. EXPERIMENTS AND DISCUSSIONS

4.1. Determination of parameters

Parameters involved in BGP_{ri} are empirically determined. Specifically, we make use of Gabor filters at three resolutions, and the corresponding $\{(\lambda_i, \sigma_i) :| i = 0, 1, 2\}$ are set as (1.3, 0.7), (5.2, 2.5), and (22, 4.5). The spatial aspect ratio γ of all the Gabor filters used is set as 1.82. For each selected resolution, Gabor filters along 8 different orientations are used and their orientations are $\{\theta_j = j\pi / 8 :| j = 0, 1,\ldots, 7\}$. That means J is set as 8. Thus, the normalized histogram generated by using a BGP_{ri} operator has 36 bins (see Table 1). At each resolution, two BGP_{ri} operators are generated, one of which is based on even-symmetric Gabor filters and the other is based on odd-symmetric ones. Thus, altogether there are 6 BGP_{ri} operators and they can generate 6 histograms correspondingly. Then, these 6 histograms are concatenated together directly to form a large histogram h with 216 bins, which is regarded as the descriptor of the image and is used for the classification purpose.

4.2. Database and methods for comparison

We conducted experiments on a modified CUReT database provided at [10, 11]. It contains 61 textures and each texture has 92 images obtained under different viewpoints and illumination directions. The proposed BGP_{ri} was compared with the other five state-of-the-art rotation invariant texture representation methods, LBP [1], MR8 [6], Joint [7], BIF [5] and M-LBP [4]. For LBP, we combined the information extracted by three operators $\text{LBP}^{riu2}_{8,1}$, $\text{LBP}^{riu2}_{16,3}$, and $\text{LBP}^{riu2}_{24,5}$ together. For MR8, 40 textons were clustered from each of the 61 texture classes using the training samples and thus the texton dictionary was of the size 2440 (61×40). In Joint, the size of the image patch was selected as 7×7, and also 40 textons were clustered from each class. For BIF, we implemented it by ourselves and the parameters were set the same as the ones described in [5].

4.3. Classification results

In order to get statistically significant classification results, N training images were randomly chosen from each class while the remaining $92 - N$ images per class were used as the test set. The partition was repeated 1000 times independently. The average accuracy along with one standard deviation for each method is reported in Table 2.

In addition to the classification accuracy, we also care about the feature size and the classification speed of each method. At the classification stage, the histogram of the test image will be built at first and then it will be matched to all the models generated from the training samples. In Table 3, we list the feature size (number of histogram bins), the time cost for one test histogram construction and the time cost for one matching at the classification stage by each method. All the algorithms were implemented with Matlab 2010b

except that a C++ implemented kd-tree (encapsulated in a MEX function) was used in MR8 and Joint to accelerate the labeling process. Experiments were performed on a Dell Inspiron 530s PC with Intel 6550 processor and 2GB RAM.

Table 2: Classification results (%)

	$N = 46$	$N = 23$	$N = 12$	$N = 6$
LBP	95.74±0.84	91.95±1.43	86.45±2.23	78.06±3.31
MR8	97.79±0.68	95.03±1.28	90.48±1.99	82.90±3.45
Joint	97.66±0.68	94.58±1.34	89.40±2.39	81.06±3.74
BIF	97.38±0.68	94.95±0.99	90.67±2.09	83.52±3.55
M-LBP	98.12±0.53	95.80±1.17	91.27±2.46	83.32±3.94
BGP_{ri}	**98.70±0.46**	**96.80±1.00**	**93.09±2.03**	**86.52±3.43**

Table 3: Feature size and time cost (msec)

	Feature size	Time cost for one histogram construction	Time cost for one matching
LBP	54	87	0.022
MR8	2440	4960	0.089
Joint	2440	13173	0.089
BIF	1296	157	0.056
M-LBP	540	221	0.035
BGP_{ri}	**216**	**136**	**0.027**

Based on Table 2 and Table 3, we can have the following findings. First of all, BGP_{ri} can achieve higher classification accuracy than all the other methods evaluated, especially in the case of less training samples. Secondly, the proposed BGP_{ri} scheme requires a moderate feature size, a little bigger than LBP but much smaller than MR8, Joint, BIF, and M-LBP. The numbers of histogram bins for MR8, Joint, BIF, and M-LBP are 2400, 2400, 1296, and 540, while BGP_{ri} only needs 216 bins. Although the feature size of BGP_{ri} is a little bigger than LBP, considering the significant gain in the classification accuracy, it is deserved. Thirdly, these six schemes have quite different classification speeds. LBP runs fastest while BGP_{ri} ranks the second. Especially, BGP_{ri} works much faster than the two clustering based methods, MR8 and Joint. BGP_{ri} is nearly 40 times faster than MR8 and 100 times faster than Joint. In MR8 and Joint, to build the histogram of the test image, every pixel on the test image needs to be labeled to one item in the texton dictionary, which is quite time consuming. Such a process is not required in LBP, M-LBP, BIF, and BGP_{ri}. Besides, an extra training period is needed in MR8 and Joint to build the texton dictionary, which is also not required in LBP, M-LBP, GIF, and BGP_{ri}.

Therefore, in general, the proposed texture representation scheme BGP_{ri} has the merits of high classification accuracy, small feature size and fast classification speed. Compared with LBP, although it has a slightly larger feature size and works a little slower, its classification accuracy is remarkably better. Compared with MR8, Joint, BIF, and M-LBP, BGP_{ri} behaves better in all aspects, including the classification accuracy, the feature size and the running speed.

5. CONCLUSION

In this paper, we presented a novel rotation invariant texture representation method, namely BGP_{ri}. In our method, the dictionary is a set of pre-defined rotation invariant binary patterns called as "rotation invariant binary Gabor patterns (BGP_{ri}s)". BGP_{ri} is strongly robust to image's rotations and is theoretically gray-scale invariant. Experiments indicate that BGP_{ri} can achieve higher classification accuracy than the other methods evaluated, especially at the occasions where the training set is small. Compared with the other state-of-the-art methods, in addition to the higher classification accuracy, BGP_{ri} also has advantages of smaller feature size and faster classification speed, which makes it a more suitable candidate in real applications.

ACKNOWLEDGEMENT

This research is supported by the Fundamental Research Funds for the Central Universities (2100219033), the NSFC (60903120), and the Innovation Program of Shanghai Municipal Education Commission (12ZZ029).

6. REFERENCES

[1] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. PAMI*, vol. 24, pp. 971-987, 2002.

[2] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *Int. J. Comput. Vis.*, vol. 43, pp. 29-44, 2001.

[3] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: a comprehensive study," *Int. J. Comput. Vis.*, vol. 73, pp. 213-238, 2007.

[4] L. Zhang, L. Zhang, Z. Guo, and D. Zhang, "Monogenic-LBP: a new approach for rotation invariant texture classification," *ICIP'10*, pp. 2677-2680, 2010.

[5] M. Crosier and L.D. Griffin, "Using basic image features for texture classification," *Int. J. Comput. Vis.*, vol. 88, pp. 447-460, 2010.

[6] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *Int. J. Comput. Vis.*, vol. 62, pp. 61-81, 2005.

[7] M. Varma and A. Zisserman, "A statistical approach to material classification using image patch exemplars," *IEEE Trans. PAMI*, vol. 31, pp. 2032-2047, 2009.

[8] L.D. Griffin, "The second order local-image-structure solid," *IEEE Trans. PAMI*, vol. 29, pp. 1355-1366, 2007.

[9] J.G. Daugman, "Uncertainty relations for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Optical Society of America A*, vol. 2, pp. 1160-1169, 1985.

[10] K.J. Dana, B.V. Ginneken, S.K. Nayar, and J.J. Koenderink, "Reflectance and texture of real world surfaces," *ACM Trans. Graphics*, vol. 18, pp. 1-34, 1999.

[11] http://www.robots.ox.ac.uk/~vgg/research/texclass/data/curetcol.zip, 2008.