

Gabor Filters and Grey-level Co-occurrence Matrices in Texture Classification

Jing Yi Tou¹, Yong Haur Tay¹ and Phooi Yee Lau²

¹*Computer Vision & Intelligent Systems (CVIS) Group,
Faculty of Information & Communication Technology
Universiti Tunku Abdul Rahman (UTAR), MALAYSIA.*

²*Instituto de Telecomunicacoes, PORTUGAL.*

Email: tayyh@mail.utar.edu.my

Abstract

Texture classification is a problem that has been studied and tested using different methods due to its valuable usage in various pattern recognition problems, such as wood recognition and rock classification. The Grey-level Co-occurrence Matrices (GLCM) and Gabor filters are both popular techniques used on texture classification. This paper combines both techniques in order to increase the accuracy. The paper used 32 textures from the Brodatz texture dataset with 1024 training samples and 1024 testing samples. GLCM achieved a recognition rate of 84.00%, Gabor filters achieved 79.58% while combination of GLCM and Gabor filters achieved a recognition rate of 88.52%, which is better than both methods. The experiments showed that the best result can be achieved by using a GLCM with grey level of 16, spatial distance of one pixel and combine with Gabor features decomposed to six features.

Key words: Texture classification, Gabor filters, Grey-level Co-occurrence Matrices

1. Introduction

Texture classification has been a field that is frequently studied as it is applicable onto many applications, such as wood species identification [1], rock texture classification [2] and defects inspection.

Since texture analysis techniques can be implemented in various machine learning problems, by studying the algorithms of texture classification, we can also implement them into other similar implementations involving texture-liked subjects such as text detection, face detection and etc.

The main objective of the paper is to examine the performance using GLCM and Gabor filters in the texture classification and by combining both techniques to enhance the accuracy.

Section 2 introduces the techniques used in this paper. Section 3 introduces the experiment dataset used and the experiments conducted. Section 4 reveals the experiment result and discussion on the analysis of the results. Section 5 is the conclusion and future works of the paper.

2. GLCM and Gabor Filters

2.1. GLCM

GLCM is proposed by Haralick et al back in 1973 [3]. It is then widely been used for various texture analysis applications, such as texture classification [4], rock texture classification, wood classification and etc.

The GLCM is generated by cumulating the total numbers of grey pixel pairs from the images. Each GLCM will be generated by defining a spatial distance d and an orientation, which can be 0 degree, 45 degree, 90 degree or 135 degree at a selected grey level G . The GLCM produced will be of size $G \times G$.

When the GLCM is constructed, $C_d(r,n)$ represents the total pixel pair value where r represents the reference pixel value and n represents the neighboring pixel value according to the spatial distance and orientation defined. The joint probability density function normalizes the GLCM by dividing every set of pixel pairs with the total number of pixel pairs used and is represented using $p(r,n)$ as shown in Eq. (1) [5].

$$p(r,n) = \frac{1}{\sum_{r=0}^{G-1} \sum_{n=0}^{G-1} C_d(r,n)} C_d(r,n) \quad (1)$$

Textural features are extracted from the GLCMs for classification process. There are a total of fourteen features for GLCM [1]. The textural features that are commonly used are shown in Eq (2) to Eq (5) [5].

$$\text{Energy:} \quad \sum_{r=0}^{G-1} \sum_{n=0}^{G-1} p(r,n)^2 \quad (2)$$

$$\text{Entropy:} \quad \sum_{r=0}^{G-1} \sum_{n=0}^{G-1} p(r,n) \log p(r,n) \quad (3)$$

$$\text{Contrast:} \quad \frac{1}{(G-1)^2} \sum_{r=0}^{G-1} \sum_{n=0}^{G-1} (r-n)^2 p(r,n) \quad (4)$$

$$\text{Homogeneity:} \quad \sum_{r=0}^{G-1} \sum_{n=0}^{G-1} \frac{p(r,n)}{(1 + |r-n|)} \quad (5)$$

2.2. Gabor Filters

The Gabor filters, also known as Gabor wavelets, is inspired by the concept of mammalian simple cortical cells [6].

The Gabor filters is represented by Eq. (6) where x and y represent the pixel position in the spatial domain, w_0 represents the radial center frequency, θ represents the orientation of the Gabor direction, and σ represents the standard deviation of the Gaussian function along the x - and y - axes where $\sigma_x = \sigma_y = \sigma$ [6].

$$\Psi(x,y,w_0,\theta) = \frac{1}{2\pi\sigma^2} e^{-((x \cos \theta + y \sin \theta)^2 + (-x \sin \theta + y \cos \theta)^2)/2\sigma^2} \times [e^{i(w_0 x \cos \theta + w_0 y \sin \theta)} - e^{-w_0^2 \sigma^2/2}] \quad (6)$$

The Gabor filter can be decomposed into two different equations, one to represent the real part and another to represent the imaginary part as shown in Eq. (7) and Eq. (8) respectively [6].

$$\Psi_r(x,y,w_0,\theta) = \frac{1}{2\pi\sigma^2} \exp\left\{-\left(\frac{x^2+y^2}{\sigma^2}\right)\right\} \times [\cos w_0 x' - e^{-w_0^2 \sigma^2/2}] \quad (7)$$

$$\Psi_i(x,y,w_0,\theta) = \frac{1}{2\pi\sigma^2} \exp\left\{-\left(\frac{x^2+y^2}{\sigma^2}\right)\right\} \times \sin w_0 x' \quad (8)$$

where

$$x' = x \cos \theta + y \sin \theta \quad y' = -x \sin \theta + y \cos \theta$$

In this paper, we used $\sigma = \pi / w_0$. Gabor features are derived from the convolution of the Gabor filter Ψ and image I as shown in Eq. (9) [6].

$$C_{\Psi} = I(x, y) * \Psi(x, y, w_0, \theta) \quad (9)$$

The term $\Psi(x, y, w_0, \theta)$ of Eq. (9) can be replaced by Eq. (7) and Eq. (8) to derive the real and imaginary parts of Eq. (6) and is represented by C_{Ψ}^r and C_{Ψ}^i respectively. The real and imaginary parts are used to compute the local properties of the image using Eq. (10) [6].

$$C_{\Psi}(x,y,w_0,\theta) = \sqrt{\|C_{\Psi}^r\|^2 + \|C_{\Psi}^i\|^2} \quad (10)$$

The convolution can be performed using a fast method by applying Fast Fourier Transform (FFT), point-to-point multiplication and Inverse Fast Fourier Transform (IFFT). It is performed on three radial center frequencies or scales, w_0 and eight orientations, θ . In this paper, the radial center frequencies and orientations are represented by θ_m in Eq. (11) where $n \in \{0, 1, 2\}$ and $m \in \{0, 1, 2, \dots, 7\}$ [6]. This method is to reduce the computations comparing to conventional method of using a smaller subwindow to perform convolution over the whole image.

$$w_n = \frac{\pi}{2(2)^{n/2}} \quad \theta_m = \frac{\pi}{8} m \quad (11)$$

2.2.1. Reducing Dimensionality. The Gabor features are at a high-dimensional space. However, higher dimension will affect the learning of the classification. A down-sampling can be performed by omitting values from the Gabor features with a factor of ρ . The Gabor features are concatenated to form a feature vector as shown in Eq. (12) [6].

$$C^{(\rho)} = (C_{\Psi}^{(\rho)}(x,y,w_0^1,\theta^1), C_{\Psi}^{(\rho)}(x,y,w_0^1,\theta^2), \dots, C_{\Psi}^{(\rho)}(x,y,w_0^1,\theta^m), \dots, C_{\Psi}^{(\rho)}(x,y,w_0^n,\theta^m))^T \quad (12)$$

A Principal Component Analysis (PCA) can be performed to further decompose the Gabor feature size. Singular Value Decomposition (SVD) can be performed for a faster method of decomposing the feature vector. The SVD decomposes an $i \times j$ matrix C into three matrices as shown in Eq. (13)

$$C = u \times \lambda \times v^T \quad (13)$$

where p is the minimum of i and j , u is a matrix of dimension $i \times k$, λ is a matrix of dimension $k \times k$, v is a matrix of dimension $j \times p$ [7].

A feature size s is selected to decompose the matrix u from $m \times k$ to an $m \times s$ matrix by directly discarding the values from the matrix beyond the size to Φ . The final Gabor feature matrix Γ after the decomposition is shown in Eq. (14) [6].

$$\Gamma^{(p)} = \Phi^T (C^{(p)} - \Psi) \quad (14)$$

2.3. k-Nearest Neighbor (k-NN)

The k-NN is used as the classifier to classify the textures. The k-NN compares each testing sample with all the training samples and chooses the k training samples with the smallest Euclidean distance. The class with highest number of the k training samples chosen will be the winning class for the testing sample.

The method requires comparison with all training samples, therefore will be slower if the training set is larger, but will affect the results if the training set is not large enough. In this paper, k-NN used for each experiment setting is using $k \in \{1, 2, 3, \dots, 10\}$.

3. Experiments

The 32 textures were selected from the Brodatz texture dataset [8] as used in [10] and [11]. Only part of the dataset was used as the classification using the entire dataset can be harder to be accomplished [9]. Each of the texture were separated into 16 samples of size 64x64, each of the samples has four different variations, the original image, a rotated image, a scaled image and an image both rotated and scaled. For each texture, eight of the sixteen sets were randomly picked for training and the remaining for testing [10][11].

For GLCM method, four GLCMs were generated for the four different orientations. The four features, which are contrast, homogeneity, energy and entropy, were extracted from each of the four GLCMs. For the Gabor filters, a 64x64 filter was used, and down-sample with $\rho = 4$. Finally, the GLCM features were combined with the Gabor features.

All experiments were tested on ten different training and testing samples. These training sets are chosen by randomly selecting eight sub images and their respective variants as training samples and the remaining ones as testing samples for each class as mentioned at the first paragraph of this section. The experiments results shown in the next section are the average recognition rates of the ten different sets.

4. Results and Discussion

4.1. GLCM

Experiment of the GLCM method was run on different settings of grey level and spatial distance. The grey levels used in the experiment are 8, 16, 32, 64, 128 and 256 while the spatial distances used are one to five pixels. The recognition results of different parameters are shown in Table 1 where the horizontal bar represents the spatial distance d and the vertical bar represents grey level G . The results shown are the best results obtained for a certain value of k of the k-NN.

Table 1. Results of GLCM

	1	2	3	4	5
8	61.48	62.13	54.97	51.32	46.31
16	74.57	65.60	56.87	49.23	46.25
32	78.07	72.70	65.68	59.04	52.24
64	83.40	84.00	79.40	71.00	65.10
128	81.70	83.20	78.50	70.06	64.28
256	80.04	83.21	77.15	69.20	61.83

The results of the GLCM show that different spatial distance and grey level can affect the results, the grey level of 64 provides the best result. The suitable spatial distance may differ for different datasets and applications depending on the spatial distance that reveals the significance of pixel pairs. The spatial distances of one or two pixels are the most ideal spatial distances in this implementation. The grey level of 64 can produces better results than 128 and 256 because it will view a few grey values as one when degrading from the original grey-scaled image with 256 grey levels. The best result achieved by the GLCM method is 84.00%.

4.2. Gabor Filters

Experiment of Gabor filters produced 6,144 features after down-sampling. The Gabor features were further decomposed using PCA, where different feature size after the decomposition was tested and the results are shown in Table 2 where the vertical bars represents the feature size s .

Table 2. Results of Gabor Filters

s	Gabor Filters
30	71.45
20	75.07
19	75.61
18	76.05
17	76.42
16	76.44
15	77.06
14	77.40
13	77.62
12	78.01
11	78.24
10	79.09
9	79.40
8	79.50
7	78.65
6	79.58
5	75.20

The experiments results show that as the feature size decreases, the recognition rate increases until it reaches a feature size of six. The experiment shows that six is the ideal feature size that provides the best recognition rate. This shows that the larger the dimensionality of the feature space, the more difficult to achieve a better classification result. However, the feature size could not be too small as it does not have enough details to help classify the problem. The results also shows the more significant values are concentrated at the front of the Gabor features since features at the back are discarded but yet the recognition rate is improved. The results show that the reduction of feature space is important for the classification as it reduces the computational steps and provides better results. However, the recognition rate is poorer than the GLCM method which is only achieving 79.53%.

4.3. GLCM and Gabor Filters

The 16 features of the GLCM were combined with the Gabor features directly into a single feature vector without modifications of the values of the original features and classified by the k-NN. Shown in Table 3 is the recognition rate for the combination of GLCM and Gabor features where the horizontal bars represents the grey level G and vertical bars represents the feature size s .

Table 3. Results for GLCM + Gabor

	8	16	32	64	128
30	77.88	77.92	77.12	75.29	75.81
20	81.86	82.04	81.04	79.67	79.92
19	82.11	82.50	81.73	79.97	80.21
18	82.47	83.16	82.13	80.40	80.62
17	83.04	83.57	82.66	80.85	81.13
16	83.33	83.74	82.73	81.31	81.47
15	83.69	84.04	83.47	81.77	81.89
14	84.20	84.57	84.09	82.30	82.35
13	84.65	85.00	84.42	82.61	82.86
12	85.00	85.65	84.81	83.22	83.40
11	85.35	86.02	85.06	83.81	83.75
10	85.68	86.58	85.61	84.35	84.31
9	85.98	87.01	86.39	84.89	84.65
8	86.04	87.08	86.11	85.16	84.94
7	86.29	87.18	86.05	84.56	84.21
6	85.70	88.53	87.40	85.70	85.28
5	86.51	88.15	86.30	83.02	82.58

The comparison of the results of using GLCM and Gabor features combined for $d = 1$ and $d = 2$ are shown in Table 4. The results shown are the best results obtained for a certain value of grey level G in the GLCM and k of the k-NN. The horizontal bar represents the methods and vertical bar represents the feature size s .

Table 4. Comparison of GLCM + Gabor

s	GLCM + Gabor ($d = 1$)	GLCM + Gabor ($d = 2$)
30	77.92	74.71
20	82.04	78.19
19	82.50	78.48
18	83.16	79.00
17	83.57	79.20
16	83.74	79.78
15	84.04	80.21
14	84.57	80.55
13	85.00	81.05
12	85.65	81.68
11	86.02	81.88
10	86.58	82.78
9	87.01	83.52
8	87.08	83.48
7	87.18	82.97
6	88.53	84.61
5	88.15	83.11

The experiment results show that GLCM method can outperform the Gabor filter but a combination of both methods is better than both of the methods when they are done separately. The combination of the two methods worked well because both methods can

extract useful features from the image but is not sufficient. However, they can complement each other to achieve higher accuracy. The best result is obtained with a setting of GLCM with grey level of 16, spatial distance of one pixel and Gabor feature size of six. The best recognition rate is 88.53%.

5. Conclusion

In our work, we have discovered that both GLCM and Gabor filters are useful in texture classification, but by combining both techniques, the accuracy is higher than both implementations. This shows that both techniques can extract useful features from the images and accumulating the useful features to improve the accuracy.

We have also discovered that the appropriate feature size for the Gabor is relatively small; this is because larger dimension of the feature size will cause difficulties for classification. However, the feature size could not be too low as it will contain too little information for classification purpose. In this paper, the best feature size tested for the Gabor filters is six. For the GLCM, the best spatial distances are usually one or two pixels as it is best describing the relationship of two neighboring pixels while the ideal grey level is usually lower than 256, as a lower grey level will merge a few grey levels as one, reducing the influence of slightly different grey levels for a same region on a subject due to differences of orientation during the acquisition. A lower grey level also produces a smaller GLCM whereby fasten the computation process.

Our future work will be working on deploying the application into an embedded platform. A classifier which stores less training information will be used for the embedded platform.

6. Acknowledgement

We would like to thank Wooi Hen Yap for sharing and discussion on Gabor filters as used in [6].

This research is partly funded my Malaysian MOSTI ScienceFund 01-02-11-SF0019.

7. References

[1] Y. L. Lew, "Design of an Intelligent Wood Recognition System for the Classification of Tropical Wood Species", Master of Engineering (Electrical) thesis, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Malaysia, 2005.

[2] M. Partio, B. Cramariuc, M. Gabboui, and A. Visa, "Rock Texture Retrieval using Gray Level Co-occurrence Matrix", *Proceedings of 5th Nordic Signal Processing Symposium*, 2002.

[3] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification", *IEEE Transactions on Systems, Man, and Cybernetics*, 1973, pp. 610-621.

[4] M. Tuceryan, and A. K. Jain, "Texture Analysis, The Handbook of Pattern Recognition and Computer Vision, Ed. 2", World Scientific Publishing Co., 1998.

[5] M. Petrou, and P. G. Sevilla, "Image Processing: Dealing with Texture", Wiley, 2006.

[6] W. H. Yap, M. Khalid, R. Yusof, "Face Verification with Gabor Representation and Support Vector Machines", *IEEE Proc. of the First Asia International Conference on Modeling & Simulation*, 2007.

[7] V. C. Klema, and A. J. Laub, "The Singular Value Decomposition: Its Computation and Some Applications", *IEEE Transactions on Automatic Control*, 1980, pp. 164-176.

[8] P. Brodatz, "Textures: A Photographic Album for Artists and Designers", Dover New York, 1996.

[9] R. W. Picard, T. Kabir, and F. Liu, "Real-time Recognition with the entire Brodatz Texture Database", *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, New York, 1993, pp. 638-639.

[10] K. Valkealahti, and E. Oja, "Reduced Multidimensional Co-occurrence Histograms in Texture Classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998, pp. 90-94.

[11] T. Ojala, K. Valkealahti, and M. Pietikainen, "Texture Discrimination with Multidimensional Distributions of Signed Gray Level Differences", *Pattern Recognition* 34, 2001, pp. 727-739.