Brushlet Features for Texture Image Retrieval

Chibiao Chen and Kap Luk Chan
Information System Research Lab, School of EEE
Nanyang Technological University, Singapore
Email: {P146702456 & eklchan}@ntu.edu.sg

Chi-Fa Chen
Dept. of Electrical Engineering
I-Shou University, Taiwan
Email: cfchen@isu.edu.tw

Abstract

In this paper, we employ a new adaptive basis of functions — brushlets for extracting texture properties. Brushlets are functions which are well localized with only one peak in the frequency domain. Hence, a representation of texture in terms of spatial frequency distributions can be constructed. The Brushlet features are used in texture image retrieval experiments to assess its effectiveness by comparing with retrieval results obtained using other commonly used wavelet and Gabor based representations. Experiments using the Brodatz texture database indicates that the brushlet features achieve a very good retrieval comparable to and slightly better than that using Gabor features and better than the wavelet features. The advantage of the brushlet features is that they require far less computation to extract than the Gabor features for achieving comparable performance.

1. Introduction

Texture can be found on many types of objects, such as woods, fabrics, and in natural scene. In a digital image library, images containing various types of textures are also often found. Texture is also one of the key descriptors for representing image content in the proposed MPEG-7 standards. The ability to efficiently describe and analyze textured pattern is then of fundamental importance to image analysis as well as to general image retrieval.

Many tools can be used to describe textures. In image retrieval, wavelet basis and Gabor basis functions have been used to represent textures. Wavelets provide an octave based decomposition of the Fourier plane with a poor angular resolution. Wavelet packets make it possible to adaptively construct an optimal tiling of the Fourier plane and have been used in texture classification [5]. However the tensor product of two real valued wavelet packets is always associated with four symmetric peaks in the frequency plane. It is therefore not possible to selectively tune and localize a unique frequency. It is also not orientation selective. Gabor filters are very popular in image analysis [3][4][7] and they show excellent texture analysis ability. Comparing to wavelet basis, Gabor basis is both frequency tunable and orientation selective. However, a filter bank consisting of many Gabor filters are often constructed to cover the entire frequency spectrum and the original image must convolve with all these filters to extract texture features. Obviously, the computation load is high.

In order to obtain a better angular resolution than the standard wavelet packets we can expand the Fourier plane into a set of *bruslets*. Brushlets were proposed by F. G. Meyer and R. R. Coifman and have been applied in image compression [8]. A brushlet is a function reasonably well localized with only one peak in frequency. Furthermore, the brushlet is a complex valued function with a phase. The phase of the 2D brushlet provides valuable information about the orientation. These properties make the brushlets suitable for texture and directional image analysis.

The objective of this paper is to study the use of this new adaptive basis of functions — *brushlets* for extracting texture properties. This paper is organized as follows. In section 2, we review the construction of brushlet bases. In section 3 we describe the image retrieval algorithm based on a brushlet decomposition of the image. Results of experiments are presented in section 4. Conclusions are given in section 5.

2. The Brushlet Bases

In signal processing, a local analysis of the Fourier spectrum of a signal is often interested. In order to analyze the local frequency content of a signal, a smooth window function is used to cut the support of the signal into adjacent intervals. Then a local Fourier analysis is performed inside each interval.

Let $G = \{g(t-m)\exp(2i\pi nt) : m, n \in \mathbf{Z}\}$ be a collection of functions in $L^2(\mathbf{R})$, where g is some fixed square-integrable bump. The Balian-Low theorem [1] states that if G is an orthogonal basis, then the Heisenberg product of g

must be infinite! In order to circumvent this obstacle raised by the Balian-Low theorem, various Wilson bases [2][6] have been constructed that use sines and cosines rather than exponentials. Such a mechanism results in basis functions that contain equal energy at both positive and negative frequencies. However, using exponentials is much preferred because the phase of the exponentials will provide information about the direction of the pattern when describing images in two dimensions. With the *smooth local periodization* technique introduced in [9], we can avoid the Balian-Low phenomenon while using exponential functions that have all but an arbitrarily small amount of energy localized in just the positive part of the frequency spectrum.

2.1. Brushlet Fundamentals

Consider a cover $\mathbf{R} = \bigcup_{n=-\infty}^{n=+\infty}[\alpha_n, \alpha_{n+1})$. Let $p_n = \alpha_{n+1} - \alpha_n$, and $c_n = \frac{\alpha_n + \alpha_{n+1}}{2}$. Around each α_n we define a neighborhood of radius ϵ. Let r be a ramp function such that

$$r(t) = \begin{cases} 0 & \text{if } t \le -1 \\ 1 & \text{if } t \ge 1 \end{cases},$$

and $r^2(t) + r^2(-t) = 1,\ \forall t \in \mathbf{R}$. Let b be the bump function supported on $[-\epsilon, \epsilon]$, $b(t) = r\left(\frac{t}{\epsilon}\right) r\left(-\frac{t}{\epsilon}\right)$. Let w_n be the windowing function supported on $[-\frac{p_n}{2} - \epsilon, \frac{p_n}{2} + \epsilon]$

$$w_n(t) = \begin{cases} r^2\left(\frac{t+\frac{p_n}{2}}{\epsilon}\right) & \text{if } t \in [-\frac{p_n}{2} - \epsilon, -\frac{p_n}{2} + \epsilon) \\ 1 & \text{if } t \in [-\frac{p_n}{2} + \epsilon, \frac{p_n}{2} - \epsilon). \\ r^2\left(\frac{-t+\frac{p_n}{2}}{\epsilon}\right) & \text{if } t \in [\frac{p_n}{2} - \epsilon, \frac{p_n}{2} + \epsilon] \end{cases}$$

Let $e_{m,n}(t) = \frac{1}{\sqrt{p_n}} \exp\left(-2i\pi m \frac{t-\alpha_n}{p_n}\right)$ the collection of exponential functions.

By using the FFT and smooth local periodization technique [9], the 1D brushlet basis $\{v_{m,n}\}$ can be constructed [8]

$$\begin{aligned} v_{m,n} = \sqrt{p_n} \exp(2i\pi\alpha_n t) \exp(i\pi p_n t)[(-1)^m p_n \hat{w}_\sigma \\ \cdot(p_n t - m) - 2i \sin(\pi p_n t) p_n \hat{b}_\sigma(p_n t + m)], \quad (1) \end{aligned}$$

where $\sigma = \frac{\epsilon}{p_n}$ is defined as a steepness factor of the window w_n and $w_\sigma(t) = w_n(p_n t)$ is supported on $[-\frac{1}{2} - \sigma, \frac{1}{2} + \sigma]$. Bump function $b_\sigma(t) = b_n(p_n t)$ is supported on $[-\sigma, \sigma]$. $\hat{w}_\sigma$ and $\hat{b}_\sigma$ are the inverse FFT of w_σ and b_σ, repectively. Note that, in (1), p_n appears as a scaling factor of the analysis, and m is the translation index of the brushlet. $v_{m,n}$ has an expression similar to a wavelet. However, as opposed to a real valued wavelet, $v_{m,n}$ is a complex valued function with a phase. The phase encodes the orientation of the brushlet pattern in the two-dimensional case. The function $v_{m,n}$ is composed of two terms, localized around $\frac{m}{p_n}$ and around $-\frac{m}{p_n}$, that are oscillating with the frequency $c_n = \frac{\alpha_n + \alpha_{n+1}}{2}$. The first term is an exponential multiplied by the window $\hat{w}_\sigma$. Since $|\hat{b}_\sigma \le \sigma|$, the second term can be made as small as possible.

2.2. The 2D Brushlets

In the two dimensional case we define two partitions of $\mathbf{R}$, $\bigcup_{j=-\infty}^{j=+\infty}[\alpha_j, \alpha_{j+1})$ and $\bigcup_{k=-\infty}^{k=+\infty}[\beta_k, \beta_{k+1})$. Let $p_j = \alpha_{j+1} - \alpha_j$, $q_k = \beta_{k+1} - \beta_k$, $c_j = \frac{\alpha_j + \alpha_{j+1}}{2}$, and $d_k = \frac{\beta_k + \beta_{k+1}}{2}$. Consider the tiling obtained by the lattice rectangles $[\alpha_j, \alpha_{j+1}) \bigotimes [\beta_k, \beta_{k+1})$. The tensor product of bases $v_{m,j}$ and $v_{n,k}$ sequence, $v_{m,j} \otimes v_{n,k}$, is an orthonormal basis for $L^2(\mathbf{R}^2)$.

The tensor product $v_{m,j}(x) \otimes v_{n,k}(y)$ is an oriented pattern oscillating with the frequency (c_j, d_k) and localized at $(\frac{m}{p_j}, \frac{n}{q_k})$.

To illustrate the orientation selective property of the 2D brushlet, we have calculated the brushlet expansion of the Brodatz image D047 (see Figure 1). A first expansion was performed with a partitioning of the Fourier plane into four quadrants. The four sets of brushlets have the orientation $\frac{\pi}{4} + k\frac{\pi}{2}, k = 0, 1, 2, 3$. A second expansion has been performed using a finer grid. Each quadrant was further divided into four sub-quadrants. The sixteen set of brushlets have twelve different orientations. The orientations $\frac{\pi}{4} + k\frac{\pi}{2}$ are associated with two different frequencies. The four lattice squares around the origin characterize the DC terms of the expansion. The other squares correspond to higher frequency textures.

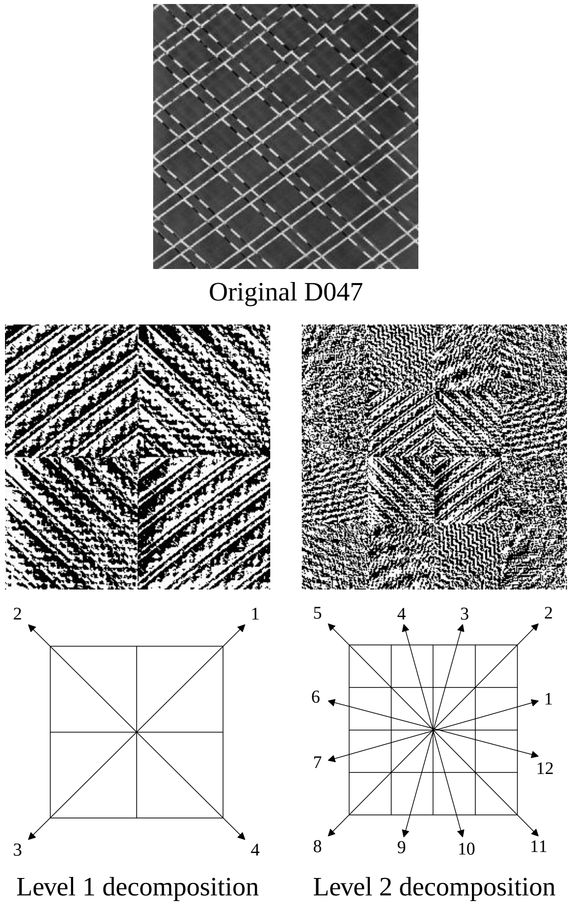
3. Image Retrieval with Brushlet Features

In this section we introduce the feature definitionsbased on the brushlet decomposition of textured images, and their use in image retrieval experiments.

3.1. Texture Feature Extraction and Representation

For a given retrieval image sample I, we extract its texture feature using a level 2 brushlet decomposition. The decomposition divides the spectrum into 16 blocks with 12 distinct directions as shown in the bottom right part of Figure 1. Denote the brushlet coefficients in each block $I_i(j,k)$ ($i = 1, \ldots, 16$, $j = 1, \ldots, R$, and $k = 1, \ldots, C$, where R and C are the number of rows and columns in each block, respectively.), then we use the mean μ_i and the standard deviation σ_i ($i = 1, \ldots, 16$) of the brushlet coefficients to represent the blocks for retrieval purpose:

$$\mu_i = \frac{1}{RC} \sum_j^R \sum_k^C |I_i(j,k)|, \qquad (2)$$


Original D047

Level 1 decomposition

Level 2 decomposition

Figure 1. Brushlet decompositions (imaginal parts) of Brodatz texture D047 and their associated directions

and

$$\sigma_i = \sqrt{\frac{1}{RC-1}\sum_{j}^{R}\sum_{k}^{C}\left(|I_i(j,k)| - \mu_i\right)^2}. \qquad (3)$$

A feature vector is now constructed using μ_i and σ_i as feature components

$$F = \left[\mu_1\ \sigma_1\ \mu_2\ \sigma_2\ \ldots\ \mu_{16}\ \sigma_{16}\right]. \qquad (4)$$

3.2. Distance Measure

Consider two image patterns m and n, and let $F^{(m)}$ and $F^{(n)}$ represent the corresponding feature vectors. Then the distance between the two patterns in the feature space is defined to be

$$d(m,n) = \sum_{i}^{16} d_i(m,n), \qquad (5)$$

where

$$d_i(m,n) = \left|\frac{\mu_i^{(m)} - \mu_i^{(n)}}{\alpha(\mu_i)}\right| + \left|\frac{\sigma_i^{(m)} - \sigma_i^{(n)}}{\alpha(\sigma_i)}\right|. \qquad (6)$$

$\alpha(\mu_i)$ and $\alpha(\sigma_i)$ are the standard deviations of the respective features over the entire image database, and are used to normalize the individual feature components.

3.3. Texture Retrieval Algorithm

Upon presentation of a query pattern, the pattern is processed to compute the feature vector as in Equation 4. The distance $d(m,n)$, where m is the query pattern and n is a pattern from the database for all images in the database, is computed. The distances are then sorted in ascending order and the closest set of N patterns are then retrieved.

4. Experimental Results

In this section, we present our experimental results. The texture database used in the experiments consists of the 112 texture classes from the Brodatz album. Each of the 512×512 images is divided into 16 non-overlapping sub-images of size 128×128, thus creating a database of 1792 texture images. All 1792 images are used as query. When an image is used as a query, it is removed from the database. The performance is measured in terms of the average retrieval rate which is defined as the average percentage number of patterns belonging to the same image as the query pattern in the top $N = 45$ most similar images.

To assess the usefulness of brushlets, we compare the retrieval results obtained by brushlet features with those obtained by Gabor and wavelet packet features. For the sake of fairness, the decomposition levels performed in brushlet and wavelet packet expansions are set to 2 and number of scales in the Gabor filtering is also set to 2. In our texture retrieval application, a pair of two distinct directions, which have a difference of π, have the same effect on the experimental results. Therefore, the number of directions of the Gabor filters are set to 6. The wavelet packets used in the experiments are 4-tag Daubechies wavelet functions. The texture feature extraction and representation, and distance measure used in wavelet packet and Gabor based retrieval algorithms are same as those used in brushlet based retrieval algorithm. The average retrieval rates for the 112 texture image classes in the database are shown in Table 1.

The overall retrieval rates and standard deviations for the entire image database are shown in Table 2. From Table 2, we can see the overall retrieval of brushlet based algorithm is the best because that the distribution of the retrieval rates of brushlet based algorithm is also the most *compact* one as well as the marginally higher averages than the Gabor based algorithm. The computation times for extracting features of one texture on a PENTIUM 533 CPU using MATLAB 5.3.1 are shown in Table 3. From Table 3, we can see the brushlet feature extraction is significantly faster than the Gabor features. Hence, the brushlet features can be considered better than the compared features for this texture image retrieval.

However we also note that there are several individual image classes bear very low retrieval rate. For example, the rate of D043 is 20.70% and the rate of D044 is 24.22%. This is due the big variation in the original images, i.e the texture is non-homogenous. We divided a big image into 16 small images and consider these images formed an image class in the image database. Therefore, the images in one of these 2 image class do not resemble each other.

5. Conclusion

In this paper, a new texture analysis tool, the brushlet, is used to decompose a texture and represent the texture. Experimental results show that these brushlet features are quite effective. Comparisons show the brushlets based features perform better than Gabor based features and much better than wavelet packets features in the terms of retrieval rates and their distributions. This can be explained as follows: The Gabor filters used in our experiments are complex functions. They also provide excellent frequency localization ability. Besides, in a Gabor based system, arbitrary number of directions and scales can be used. This indicates that the analysis ability of Gabor can be *infinite*. However, all these come at the expense of computation burden; wavelet packets of real functions cannot localize a unique frequency in the Fourier spectrum. The angular resolution of brushlets is better than that of wavelet packets too. The main advantage of brushlets is less computation required for achieving comparable retrieval performance as using Gabor features.

References

[1] I. Daubechies. Ten lectures on wavelets. *CBMS-NSF Conference Series in Applied Mathematics*, SIAM Ed. 1992.

[2] I. Daubechies, S. Jaffard, and J. L. Journé. A simple Wilson orthnormal basis with exponential decay. *SIAM J. Math. Anal.*, 22:554–572, 1991.

[3] J. G. Daugman. Complete dicrete 2D gabor transforms by neural networks for image analysis and compression. *IEEE Trans. Acoust., Speech, Signal Processing*, 36:1169–1179, 1988.

[4] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[5] A. Laine and J. Fan. Texture classification by wavelet packet signatures. *IEEE Trans. Pattern Anal. Machine Intell.*, 15:1186–1191, 1993.

[6] H. Malvar. Lapped transform for efficient transform/subband coding. *IEEE Trans. Acoust. Sign. Speech Process. Machine*, 38:969–978, 1990.

[7] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Machine Intell.*, 18(8):837–842, 1996.

[8] F. G. Meyer and R. R. Coifman. Brushlet: A tool for directional image analysis and image compression. *Applied and Computational Harmonic Analysis*, 4:147–187, 1997.

[9] M. V. Wickerhauser. *Adapted Wavelet Analysis from Theory to Software*. A K Peters, Ltd., A K Peters, Ltd., 289 Linden Street, Wellesley, MA 02181, USA, 1994.

Table 1. Average retrieval rate for the 112 texture image classes in the database (B: Brushlet features, G: Gabor features, and W: Wavelet packet features)

Class	Rate (%)	Class	Rate (%)
	B G W		B G W
D001	99.22 95.70 95.70	D057	100.0 100.0 100.0
D002	88.28 75.39 64.84	D058	32.42 30.86 27.34
D003	82.03 95.70 93.75	D059	49.61 55.08 53.13
D004	100.0 100.0 77.84	D060	69.92 89.84 68.36
D005	89.84 75.00 52.73	D061	75.39 63.28 69.14
D006	100.0 100.0 100.0	D062	96.09 70.31 81.64
D007	64.45 56.25 48.44	D063	62.89 54.30 61.33
D008	97.66 95.70 91.41	D064	100.0 100.0 100.0
D009	100.0 100.0 85.16	D065	100.0 100.0 100.0
D010	82.03 80.08 76.56	D066	94.53 97.27 98.44
D011	96.88 100.0 84.77	D067	75.00 94.53 67.97
D012	90.63 87.50 64.06	D068	100.0 100.0 100.0
D013	59.77 63.28 43.36	D069	65.63 59.38 65.23
D014	100.0 100.0 100.0	D070	68.75 61.72 64.06
D015	59.38 63.67 69.92	D071	92.97 95.31 95.31
D016	100.0 100.0 99.22	D072	73.44 62.50 70.31
D017	100.0 100.0 93.36	D073	62.89 52.73 47.66
D018	93.75 97.27 78.13	D074	98.05 90.63 91.41
D019	93.75 89.84 73.83	D075	100.0 100.0 100.0
D020	100.0 100.0 100.0	D076	82.81 100.0 88.67
D021	100.0 100.0 100.0	D077	100.0 100.0 96.88
D022	58.20 56.64 53.52	D078	99.61 98.05 90.63
D023	93.75 84.77 78.22	D079	92.97 98.83 89.45
D024	96.48 98.83 97.66	D080	79.69 92.97 83.59
D025	60.55 85.94 68.36	D081	92.58 99.22 91.80
D026	97.66 100.0 97.66	D082	99.22 100.0 99.61
D027	95.70 93.36 92.58	D083	100.0 100.0 99.61
D028	100.0 99.22 98.83	D084	100.0 100.0 100.0
D029	100.0 98.83 94.92	D085	100.0 100.0 100.0
D030	81.64 76.56 82.03	D086	80.08 99.61 73.83
D031	92.97 57.81 60.16	D087	100.0 99.61 88.67
D032	100.0 100.0 100.0	D088	96.88 53.52 86.72
D033	100.0 95.31 98.44	D089	60.55 46.09 46.09
D034	100.0 94.14 91.02	D090	56.64 63.28 57.42
D035	93.36 97.66 91.02	D091	54.69 67.58 69.14
D036	78.91 75.00 67.19	D092	100.0 98.44 95.31
D037	95.31 90.63 68.36	D093	86.33 99.22 68.36
D038	75.78 69.92 61.72	D094	92.19 98.44 75.39
D039	53.91 50.78 48.44	D095	100.0 100.0 97.66
D040	82.03 73.44 72.27	D096	90.63 95.31 80.86
D041	92.97 77.34 73.83	D097	66.02 67.58 49.61
D042	51.17 35.16 44.92	D098	73.83 66.41 80.47
D043	20.70 18.36 13.67	D099	66.02 62.11 71.88
D044	24.22 21.48 20.70	D100	57.03 52.30 31.64
D045	32.03 23.05 25.00	D101	100.0 100.0 100.0
D046	65.63 60.94 71.48	D102	100.0 100.0 100.0
D047	100.0 100.0 100.0	D103	91.02 100.0 100.0
D048	100.0 100.0 100.0	D104	88.28 100.0 100.0
D049	100.0 100.0 100.0	D105	100.0 100.0 100.0
D050	62.89 73.83 69.14	D106	99.22 98.83 100.0
D051	100.0 92.97 93.75	D107	99.61 76.56 88.67
D052	53.13 49.61 44.92	D108	65.23 62.11 51.95
D053	100.0 100.0 100.0	D109	91.41 87.89 80.47
D054	59.38 51.95 37.89	D110	98.83 99.22 95.70
D055	100.0 100.0 99.61	D111	96.88 88.28 79.69
D056	100.0 100.0 100.0	D112	95.31 68.36 69.53

Table 2. Overall retrieval rates and standard deviations for the entire database

	Brushlet	Gabor	WP
Overall Retrieval Rate (%)	84.76	83.02	78.97
Standard Deviation (%)	18.52	21.00	21.24

Table 3. Average computation time for extracting features of one texture

	Brushlet	Gabor	WP
Computation Time (Sec)	0.439	0.600	0.276
Ratio	1.000	1.366	0.629