Combining powerful local and global statistics for texture description*

Yong Xu[1], SiBin Huang[1], Hui Ji[2] and Cornelia Fermüller[3]
[1]School of Computer Science & Engineering, South China Univ. of Tech., Guangzhou 510006, China
[2]Department of Mathematics, National University of Singapore, Singapore 117542
[3]Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, U.S.A.
{yxu@scut.edu.cn, huang.sibin@mail.scut.edu.cn, matjh@nus.edu.sg, fer@umiacs.umd.edu}

Abstract

A texture descriptor is proposed, which combines local highly discriminative features with the global statistics of fractal geometry to achieve high descriptive power, but also invariance to geometric and illumination transformations. As local measurements SIFT features are estimated densely at multiple window sizes and discretized. On each of the discretized measurements the fractal dimension is computed to obtain the so-called multifractal spectrum, which is invariant to geometric transformations and illumination changes. Finally to achieve robustness to scale changes, a multi-scale representation of the multifractal spectrum is developed using a framelet system, that is, a redundant tight wavelet frame system. Experiments on classification demonstrate that the descriptor outperforms existing methods on the UIUC as well as the UMD high-resolution dataset.

1. Introduction

A main challenge for texture analysis and recognition lies in achieving robustness to a wide range of geometric and photometric transforms (Zhu [23]), e.g. illumination changes, occlusions, non-rigid deformations of the surface, and viewpoint changes. Thus, there have been enduring interests in constructing a compact description which not only is highly discriminative to intra-class textures, but also robust to inter-class variations and environmental changes.

In the past, a number of texture descriptions have been proposed to achieve robustness to environmental changes by using local descriptors or filter responses (e.g. [3], [6], [7], [8], [10], [11], [12], [19], [20]). The basic idea of these approaches is to compute a texton histogram based on some appearance-based texton dictionary. These approaches can be classified into the dense and the sparse ones. The *dense* approach uses appearance descriptors at every pixel. For example, Varma *et al*. [19] used filter bank responses. The *sparse* approach uses appearance descriptors at a sparse set of keypoints. For example, Lazebnik *et al*. [7] and Zhang *et al*. [22] showed good results on texture classification using a texture description based on affine-invariant appearance descriptors.

Both, the sparse and the dense approach, have advantages and disadvantages. By using appearance descriptors only on selected interesting points, the sparse approach can make the resulting description more robust to environmental changes. However, with only a sparse subset, it might miss important texture primitives and does not provide enough measurements for later statistical characterization. Also, often there are issues on the stability and repeatability of the associated region detector. The dense approach uses the appearance descriptors at all available pixels, which provides rich information for characterizing the texture. However, it also is more sensitive to significant environmental changes. The reason is that the appearance descriptors usually are not invariant to significant environmental changes, unless sophisticated adaptive region processing is used, which in general only works well on a sparse set of points.

There are two major components to all sparse and dense methods: the local appearance descriptor and the global statistical characterization. In order to achieve invariance to environmental changes, both components should be invariant to these changes. Most current approaches take the histogram as the statistical characterization. An interesting replacement of the histogram was proposed in Xu *et al*. [21], the so-called MFS (multi-fractal spectra). While the histogram bins the elements according to some criteria and counts the elements in each bin, the MFS characterizes the elements in each bin using fractal geometry, and this characterization encodes the spatial distribution of the image pixels in the bin. The MFS has also been used in other applications, *e.g*., texture segmentation (Conci *et al*. [2]). The pixel classification of the method (Xu *et al*. [21]) is based on the local density function. The method showed solid invariance to a wide range of geometrical changes including

*Project supported by the National Nature Science Foundation of China (No. 60603022).

viewpoint changes and non-rigid surface changes and reasonable robustness to illumination changes. However, since the method is based on very simple measurements (intensity, Laplacian and gradients) and has very low dimension, it has limited discriminability. On the other hand, there has been great progress on designing robust features in recent years. The best known is the SIFT feature by Lowe *et al*. [9]. This inspires us to develop a new texture description, which combines the statistical power of multi-fractal analysis with state-of-the-art local appearance descriptors to achieve both high discriminative power and solid invariance to most environmental changes.

The appearance descriptor in our proposed approach is based on the well-established SIFT descriptor (Lowe *et al*. [9]), but without the keypoint detection. By calculating the orientation histograms on the square neighborhood of a pixel, the SIFT descriptor becomes very robust to local rotation and illumination changes. However, the orientation histogram is not robust to changes in scale, because the size of the neighborhood is fixed. In order to obtain invariance to the scaling effect, the orientation histogram has to be used under multiple levels (multiple neighborhood sizes on which the orientation histogram is evaluated). This also has the benefit that more information is produced for later statistical characterization.

We propose an approach to obtain robustness to scale changes using as input the multi-level orientation histogram at every pixel. The basic idea is as follows. For each level we calculate an MFS using the pixel classification based on the orientation histogram of this level. Then, given a set of MFSs, the robustness to scale changes is obtained by adding another statistical layer to the MFS, such that the representation is not sensitive to the level at which the MFS is estimated. This second-layer statistical characterization on the set of MFSs is based on the multi-scale representation of the MFS under a redundant framelet system ([4]).

In summary, our approach starts from the orientation histogram of all pixels under multiple levels. A first statistical characterization is obtained by calculating the MFS with respect to the orientation histogram of each level. Then a second statistical characterization is obtained from the set of MFSs of all levels by using the leading coefficients of the set under a tight framelet system. This results in a texture description that is robust to both geometrical changes and illumination changes.

The remainder of the paper is organized as follows. Section 2 gives a brief review on two mathematical tools used in our approach: multi-fractal analysis and tight framelet system. Section 3 presents the detailed description of our approach. Section 4 presents experimental results on texture classification and demonstrates the performance of our method in comparison to other methods.

2. Preliminary knowledge

Here we give a brief review on two mathematical tools used in our approach. One is multi-fractal analysis; the other is tight framelet system.

2.1. Multi-Fractal analysis ([5])

We begin with the introduction of the concept of fractal dimension. Consider a point set on the image, for example the set of image points which have a certain value. The fractal dimension of a given point set E is a statistical measurement describing how the point set E appears to fill space when one zooms in to finer scales. A popular model of the fractal dimension is the so-called *box-counting* fractal dimension. Let the space be covered by a mesh of $n \times n$ squares. Given a point set $E \subset \mathbb{R}^2$, let $\#(E, \frac{i}{n})$ be the number of $\frac{i}{n}$-mesh squares that intersect E for $i = 1, 2, \cdots$. Then the fractal dimension $dim(E)$ of E ([5]) is defined as

$$\dim(E) = \lim_{n\to\infty} \frac{\log \#(E, \frac{1}{n})}{-\log \frac{1}{n}}.$$

In practice, as the resolution is limited, we estimate the slope of $\log \#(E, \frac{i}{n})$ with respect to $-\log \frac{i}{n}$ for $i = 1, 2, \cdots m (m \leq n)$ using the least squares method.

Multi-fractal analysis is a generalization of the fractal dimension when a single exponent is not enough to describe a complicated dynamics. Divide the space into multiple point set E_α according to some categorization. For each point set E_α, which is the collection of all points with the same α, let $dim(E_\alpha)$ denote its fractal dimension. The MFS is described by the multi-fractal curve $D(\alpha)$ vs. α. In the classical definition of the MFS, the categorization is defined by the density function [21].

2.2. Tight framelet system ([4])

A tight framelet system is a redundant generalized wavelet system, which is particularly suited for presenting the signal under multiple resolutions. A countable set of functions $X \subset L^2(\mathbb{R})$ is called a *tight frame* of $L^2(\mathbb{R})$ if

$$f = \sum_{h\in X} \langle f, h\rangle h, \quad \forall f \in L^2(\mathbb{R}). \tag{1}$$

where $\langle \cdot, \cdot \rangle$ is the inner product of $L^2(\mathbb{R})$. The tight frame is a generalization of an orthonormal basis. It has greater flexibility than an orthonormal basis by sacrificing the orthonormality and the linear independence, but it still allows for perfect reconstruction like the orthonormal basis. The filters of framelets have attractive properties, not present in those of wavelets: *e.g*., symmetry (anti-symmetry), smoothness, shorter support.

Given a finite set of generating functions

$$\Psi := \{\psi_1, \cdots, \psi_r\} \subset L^2(\mathbb{R}),$$

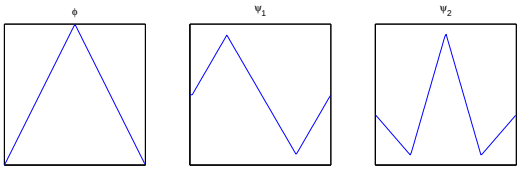


Figure 1. Piecewise linear framelets.

one particularly interesting type of tight frame $X \subset L^2(\mathbb{R})$ is a tight frame defined by the dilations and the shifts of the generators from Ψ:

$$X := \{\psi_{\ell,j,k} : 1 \le \ell \le r;\ j,k \in \mathbb{Z}\} \tag{2}$$

with $\psi_{\ell,j,k} := 2^{j/2}\psi_\ell(2^j \cdot -k)$, $\psi_\ell \in \Psi$. A *tight framelet system* is generated by constructing the set of framelets Ψ, which starts with a refinable function $\phi \in L^2(\mathbb{R})$ such that $\phi(x) = \sum_k h_0(k)\phi(2x-k)$ for some low-pass filter h_0. Then the set of framelets is defined as $\psi_\ell(x) = \sum_k h_\ell(k)\phi(2x-k)$ with high-pass filters h_ℓs, which satisfy the so-called *unitary extension principle*:

$$\tau_{h_0}(\omega)\overline{\tau_{h_0}(\omega+\gamma\pi)} + \sum_{\ell=1}^{r} \tau_{h_\ell}(\omega)\overline{\tau_{h_\ell}(\omega+\gamma\pi)} = \delta(\gamma)$$

for $\gamma = 0, 1$, where $\tau_h(\omega)$ is the trigonometric polynomial of the filter h: $\tau_h(\omega) = \sum_k h(k)e^{ik\omega}$. In this paper, we use the piece-wise linear framelets ([4]):

$$h_0 = \frac{1}{4}[1,2,1];\ h_1 = \frac{\sqrt{2}}{4}[1,0,-1];\ h_2 = \frac{1}{4}[-1,2,-1].$$

See Fig. 1 for the corresponding ϕ and ψ_1, ψ_2.

Therefore, we have a multi-scale decomposition of any function $f \in L^2(\mathbb{R})$:

$$f(x) = \sum_{\ell=1,2,j,k\in\mathbb{Z}} 2^{\frac{j}{2}}\langle f, \psi_\ell(2^j x-k)\rangle\psi_\ell(2^j x-k). \tag{3}$$

In this paper, we use a discrete implementation which uses a more redundant multi-level tight framelet decomposition without downsampling, as in Cai *et al*. [1]. Given the linear-piecewise framelet system associated with $h_k, k = 0,1,2$, let $\mathcal{H}$ denote the convolution operator with filter h under the Neumann boundary condition, The multi-level decomposition operator up to L can be written as

$$A = \begin{bmatrix} \left(\prod_{\ell=0}^{L-1}\mathcal{H}_0^{(L-\ell)}\right)^T; \\ \left(\mathcal{H}_1^{(L)}\prod_{\ell=1}^{L-1}\mathcal{H}_0^{(L-\ell)}\right)^T; \left(\mathcal{H}_r^{(L)}\prod_{\ell=1}^{L-1}\mathcal{H}_0^{(L-\ell)}\right)^T; \\ \vdots \\ \left(\mathcal{H}_1^{(1)}\right)^T; \left(\mathcal{H}_r^{(1)}\right)^T \end{bmatrix}^T, \tag{4}$$

where $\mathcal{H}^{(i)}$ denote the convolution operators associated with the given filter H at the i-th level without downsampling (See Cai *et al*. [1] for more details). Given any vector $\mathbf{f} \in \mathbb{R}^n$, then the framelet coefficients $A\mathbf{f}$ and $\mathbf{f}$ are related as

$$\mathbf{f} = A^T(A\mathbf{f}).$$

It is noted that we have $A^TA = I$ but $AA^T \neq I$ unless the tight framelet system is degenerated to an orthonormal wavelet system.

3. Our texture description

There are three stages in the computation of our texture description.

1. Compute the orientation histogram at each pixel with respect to 8 different neighborhood sizes ranging from $3\times3, 5\times5, \cdots, 17\times17$.

2. For each neighborhood size, discretize the orientation histograms into 29 values, and for each value compute the MFS of the image resulting in a 8×29 matrix of MFSs.

3. Perform a multi-scale decomposition under a tight framelet system on each of the 29 columns of the MFS matrix to construct a descriptor of size $(2L+1)\times8\times29$ and extract the leading coefficients, where L is scale.

Next we give a detailed description of each step.

3.1. Multi-level orientation histograms

The orientation histogram proposed in [9] has shown good robustness to illumination changes and is invariant to rotation. Given a pixel, the gradient magnitude and the orientation are computed for all pixels in the neighborhood, from which the orientation histogram is formed by discretizing orientations and weighing by the gradient magnitude. See Fig. 2 for illustration. The final orientation histogram is rotated to align its dominant orientation with a fiducial direction.

In our implementation we use only 8 gradient directions, covering 45 degrees each, as too many orientations will lead to instability for small neighborhood sizes. Neighborhoods of different sizes will lead to different histograms. In our approach we use multiple neighborhood sizes to generate a sequence of orientation histograms at every point, in total 8 orientation histograms. The size of the square neighborhood ranges from 3×3 to 17×17 with a step size of 2.

3.2. Pixel classification and the MFS

In order to boost the efficiency of our algorithm, we take a fixed bin partitioning scheme based on 29 classes of orientation histogram templates. The 29 template classes (shown in Fig. 3) are constructed based on the spatial structure of the orientation histogram, that is, the number of significant image gradient orientations and their relative positions.

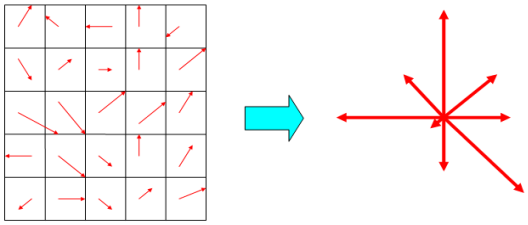


Figure 2. Orientation histogram when using the neighborhood of size 5×5.

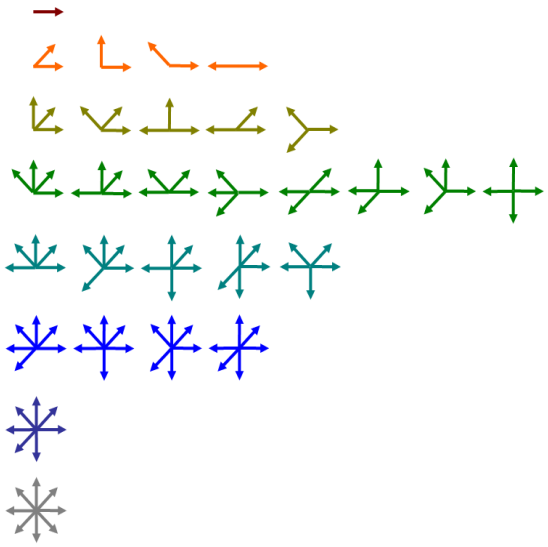


Figure 3. Basic elements of 29 orientation histogram templates.

Each of the class templates includes all rotated and mirror-reflected versions of the basic element (as shown in Fig. 3). Fig. 4 shows one rotated example and one mirror reflected example. To obtain from the orientation histogram a template, each bin in the orientation histogram is set to 0 if the amount is less than $\frac{1}{8}$ of the total amount and set to 1 otherwise.

Then for each level (neighborhood size of orientation histogram) we compute one MFS feature vector as follows: For each of the 29 templates we obtain a binary image by setting the pixel to 1 if it has that template, and 0 otherwise. The box-counting fractal dimension is computed for these 29 binary images resulting in the MFS vector. In total we have 8 MFS vectors of dimension 29. We denote such a multi-level set of MFSs by a 8×29 matrix $K(s, n)$ with s denoting the level (neighborhood size of the orientation histogram) and n denoting the histogram template.

It is easy to see that using orientation histogram templates, the pixel classification is invariant to rotation,

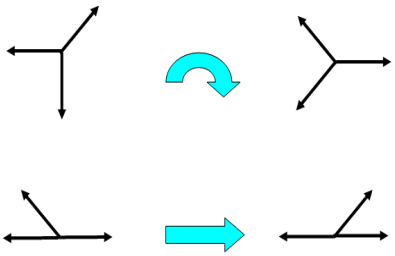


Figure 4. One rotated element and one mirror-reflected element from the basic elements shown in the right column.

mirror-reflection, and also it is more robust to illumination changes than the original orientation histogram in [9]. In addition, adopting the MFS as the statistical characterization leads to global invariance to bi-Lipschitz spatial transforms ([21]), which includes 3D view-point changes and non-rigid surface changes.

3.3. Construction of texture description

The multi-level MFS $K(s, n)$ already achieves invariance to illumination and many geometrical transforms. In order to get better robustness to scale changes, we construct a multi-scale representation of $K(s, n)$ using a non-downsampling tight framelet transform along the axis s.

The non-downsampling tight framelet transform provides a redundant presentation which can characterize the point singularity well in a multi-scale fashion. The L-scale decomposition of $K(s, n)$ under a 1d tight framelet system with respect to s is defined as

$$\mathcal{F}(j, s, n) := AE,$$

where A is defined in (4). E are the framelet coefficients. $\mathcal{F}$ consists of two parts: one output of low-pass filtering using h_0 at the scale 2^{-L}; multiple outputs of high-pass filtering using $h_1, \cdots, h_r$ at multiple scales ranging from $2^{-1}, \cdots, 2^{-L}$. The output of each high-pass filtering has three variables: scale $2^{-j}, j = 1, \cdots, L$, level $s, s = 1, \cdots, 8$ and bin index $n, n = 1, 2, \cdots, 29$. The new constructed texture representation $\mathcal{F}$ of $K(s, n)$ under a tight framelet system inherently provides invariance to not too large scale changes in the image.

The motivation for using the multi-scale representation $\mathcal{F}(j, s, n)$ of $E(s, n)$ is that a scale change in the image corresponds to the same scale change in the neighborhood size of the orientation histogram. For example, if the image is zoomed in by 20%, the orientation histogram of a pixel w.r.t. a neighborhood size of 5×5 corresponds to that of the same pixel w.r.t. a neighborhood size of 6×6. Thus, robustness to the scaling effect in the image is equivalent to

robustness to a shift of $E(s,n)$ along the axis of s. By using a multi-scale representation $\mathcal{F}(j,s,n)$ of E, the robustness to a shift of $E(s,n)$ is improved. As we see, the output of low-pass filtering under scale L is the local average across multiple levels s, and the output of multi-scale high-pass filtering could be viewed as the information about the changes of E w.r.t. scale changes, which often is consistent across multiple scales for natural textures.

Fig. 6 shows the single-scale framelet-based representation $\mathcal{F}$ for four MFSs corresponding to the four texture images shown in Fig. 5 (a).

4. Experimental evaluation

We evaluated the performance of the proposed texture descriptor on the classical texture classification task. Two datasets are tested: one is from UIUC ([16]) and the other is from UMD ([17]). The UIUC texture dataset consists of 1000 uncalibrated, unregistered images: 40 samples for each of 25 textures with a resolution 640×480 pixels, and the UMD texture dataset consists of 1000 uncalibrated, unregistered images: 40 samples for each of 25 textures with a resolution of 1280×900 pixels. Significant viewpoint changes and scale differences are present within these two texture datasets, and the illumination conditions are uncontrolled.

In our classification experiments, the training set is selected as a fixed size random subset of the class, and all remaining images are used as the test set. The texture description in our implementation is based on a two-scale framelet-based representation of the MFSs. The reported classification rate is the average over 200 random subsets. An SVM classifier (Tresp *et al*. [15]) is used in the experiments, which was implemented as in Pontil *et al*. [13]. The features of the training set are used to train the hyperplane of the SVM classifier using RBF kernels as described in Scholkopf *et al*. [14]. The optimal parameters are found by cross-validation.

We compared our method against the three methods in Lazebnik *et al*. [7], Varma *et al*. [18] and Xu *et al*. [21]. The first one is the (H+L)(S+R) method by Lazebnik *et al*. [7], which is based on a sophisticated point-based texture representation. The basic idea is to first characterize the texture by clusters of elliptic regions. The ellipses are then transformed to circles such that the local descriptor is invariant to affine transforms. Two descriptors (SPIN and SIFT) are defined on each region. The resulting texture descriptor is the histogram of clusters of these local descriptors, and the descriptors are compared using the EMD distance. The second method is the VG-fractal method by Varma and Garg [18], which use properties of the local density function of various image measurements resulting in a 13 dimensional descriptor. The resulting texture descriptor is the histogram of clusters of these local descriptors. The third method is the MFS method by Xu *et al*. [21], which is closest to our method. In [21] the pixel classification is based on the local density function at each point. In total three local density functions based on image intensity, image gradient and image Laplacian were defined, and the texture descriptor is obtained by combining the three MFSs based on the corresponding pixel classification.

The results on the UIUC dataset using the SVM classifier for the (H+L)(S+R) method is from [7]. All other results are obtained from our implementations. We denote our approach as OTF method. Fig. 7 shows the classification rate vs. the number of training samples on the UIUC dataset. Fig. 8 shows the classification percentage vs. the index of classes on the UIUC dataset based on 20 training samples. Fig. 9 and Fig. 10 show the same experiments for the UMD dataset.

From Fig. 7–Fig. 10 it can be seen that on the two datasets our method clearly outperformed the VG-fractal method and the MFS method. Also our method obtained better results than the (H+L)(S+R) method. We emphasize that heavy clustering is needed in both, the VG-fractal method and the (H+L)(S+R) method, which is very computationally expensive. On the contrary, our approach is much simpler and efficient without requiring clustering.

5. Summary and Conclusions

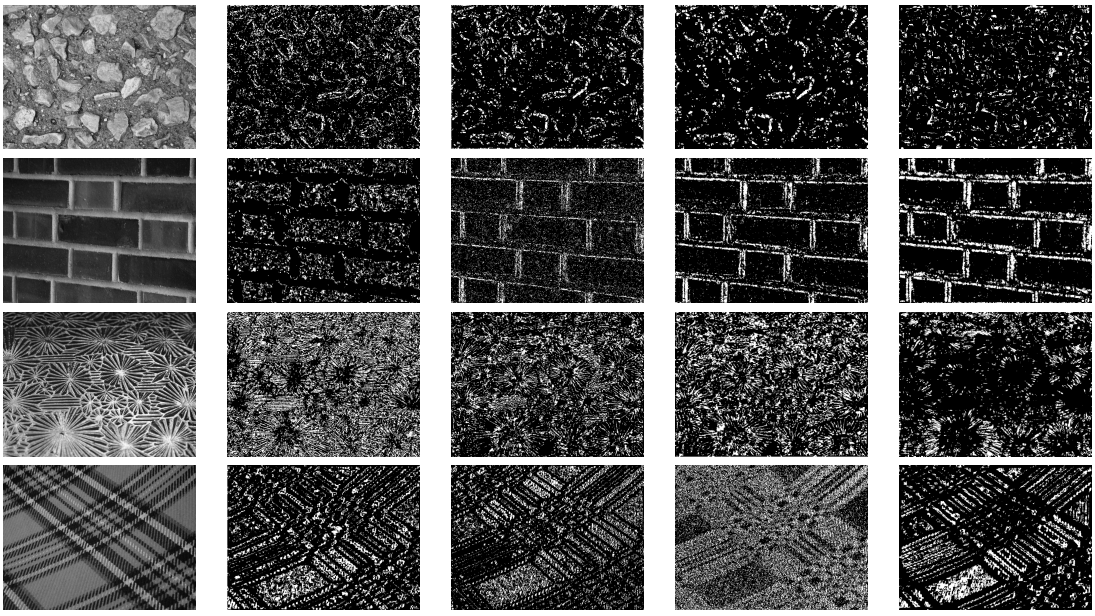
In this paper we proposed a texture descriptor based on the MFS defined on local orientation histograms under multiple neighborhood sizes. Using the multi-level orientation histograms leads to a pixel classification which is robust to illumination changes and local geometric changes. Combining the orientation histograms globally using the MFS multi-fractal spectrum introduced in [21], we obtain a texture representation which has strong invariance to both illumination changes and environmental changes. Furthermore, we projected the MFSs into a tight frame system to enhance the invariance to large scale changes. Our texture descriptor also is very efficient and simple to compute without requiring feature detection and clustering. The experiments show that our method performed particularly well on texture classification when using an SVM-based classifier.

References

[1] J. Cai, R. H. Chan and Z. Shen, "A framelet-based image inpainting algorithm", *Applied and Computational Harmonic Analysis*, 24(2), pp. 131-149, 2008.

[2] A. Conci and L. H. Monterio, "Multifractal characterization of texture-based segmentation", *ICIP*, pp. 792-795, 2000.

[3] K. Dana and S. Nayar, "Histogram model for 3d textures", *CVPR*, pp. 618-624, 1998.



(a) (b) (c) (d) (e)

Figure 5. Column (a) are four different texture images. Column (b)–(e) are examples of binary images with respect to pixel classification based on the orientation histogram templates.

[4] I. Daubechies, B. Han, A. Ron and Z. Shen, "Framelets: MRA-based constructions of wavelet frames", *Applied and Computational Harmonic Analysis*, 14, pp. 1-46, 2003.

[5] K. J. Falconer, *Techniques in Fractal Geometry*, John Wiley, 1997.

[6] E. Hayman, B. Caputo, M. Fritz and J. O. Eklundh, "On the significance of real-world conditions for material classification", *ECCV*, pp. 253-266, 2004.

[7] S. Lazebnik, *Local Semi-local and Global Models for Texture, Object and Scene Recognition*, Ph.D. Dissertation, University of Illinois at Urbana-Champaign, 2006.

[8] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons", *IJCV*, 43(1), pp. 29-44, 2001.

[9] D. Lowe, "Distinctive image features from scale invariant keypoints", *IJCV*, 60(2), pp. 91-110, 2004.

[10] B. B. Mandelbrot, *The Fractal Geometry of Nature*, San Francisco, CA: Freeman, 1982.

[11] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan and A. Yamada, "Color and texture descriptors", *IEEE Trans. on Circuits and Systems for Video Technology*, 11(6), pp. 703-715, 2001.

[12] F. Mindru, T. Tuytelaars, L. Van Gool and T. Moons, "Moment invariants for recognition under changing viewpoint and illumination", *CVIU*, 94(1-3), pp. 3-27, 2004.

[13] M. Pontil and A. Verri. "Support vector machines for 3D object recognition", *PAMI*, 20(6), pp. 637-646, 1998.

[14] B. Scholkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, Cambridge, MA, 2002.

[15] V. Tresp and A. Schwaighofer, "Scalable kernel systems", *Proceedings of ICANN 2001, Lecture Notes in Computer Science 2130*, pp. 285-291. Springer Verlag, 2001.

[16] UIUC: http://www-cvr.ai.uiuc.edu/ponce_grp/data/index.html.

[17] UMD: http://www.cfar.umd.edu/~fer/High-resolution-database/hr_database.htm.

[18] M. Varma and R. Garg, "Locally invariant fractal features for statistical texture classification", *ICCV*, 2007.

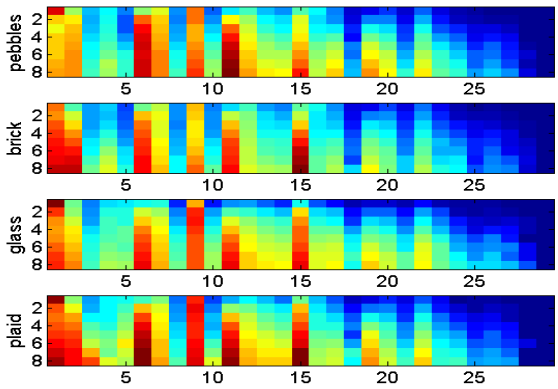
[19] M. Varma and A. Zisserman, "Classifying images of materials: Achieving viewpoint and illumination independence", *ECCV*, 3, pp. 255-271, 2002.

[20] J. Wu and M. J. Chantler, "Combining gradient and albedo for rotation invariant classification of 2D surface texture", *ICCV*, 2, pp. 848-855, 2003.

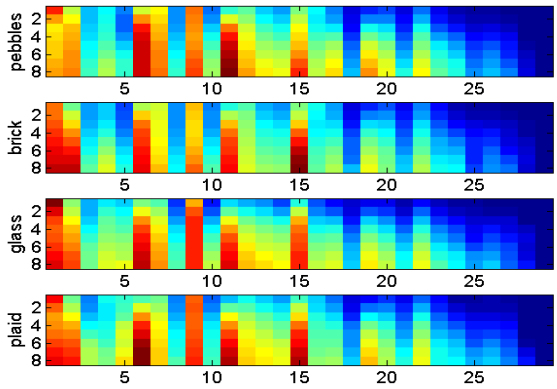
[21] Y. Xu, H. Ji and C. Fermuller, "A projective invariant for textures", *CVPR*, 2, pp. 1932-1939, 2006.

[22] J. Zhang, M. Marszalek, S. Lazebnik and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study", *IJCV*, 73(2), pp. 213C238, 2007.

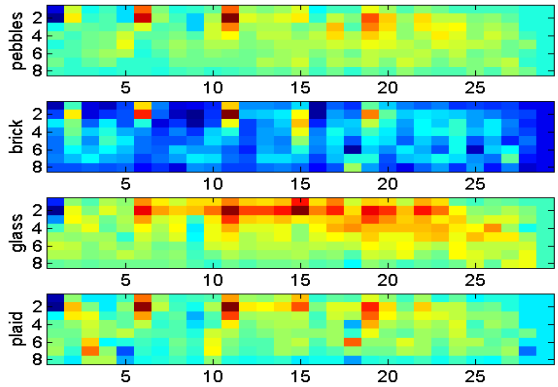
[23] S. C. Zhu, Y. Wu and D. Mumford, "Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling", *IJCV*, 27(2), pp. 107-126, 1998.



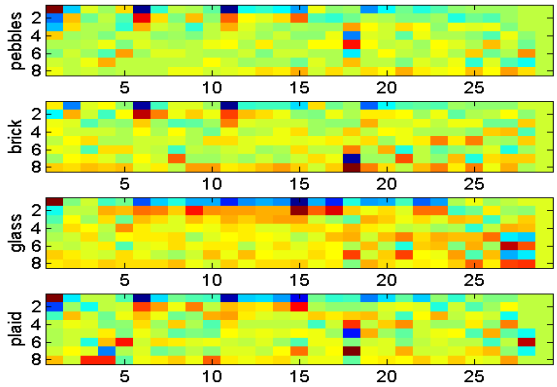
(a) Original MFSs of four different texture images.



(b) The framelet coefficients of MFSs for framelet h_0.

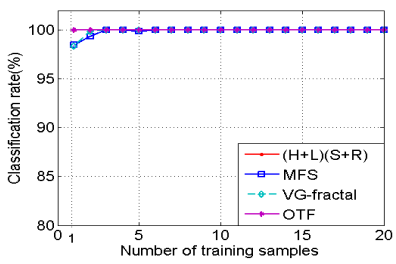


(c) The framelet coefficients of MFSs for framelet h_1.

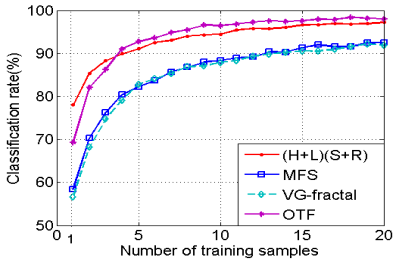


(d) The framelet coefficients of MFSs for framelet h_2.

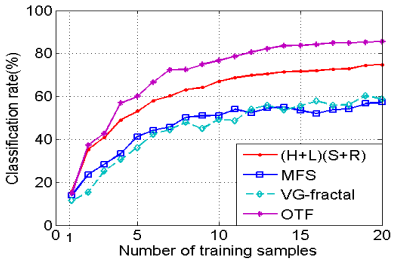
Figure 6. MFSs of four different texture images in Fig. 5 (a) and the corresponding framelet coefficients filtered by piece-wise linear framelets.



(a)

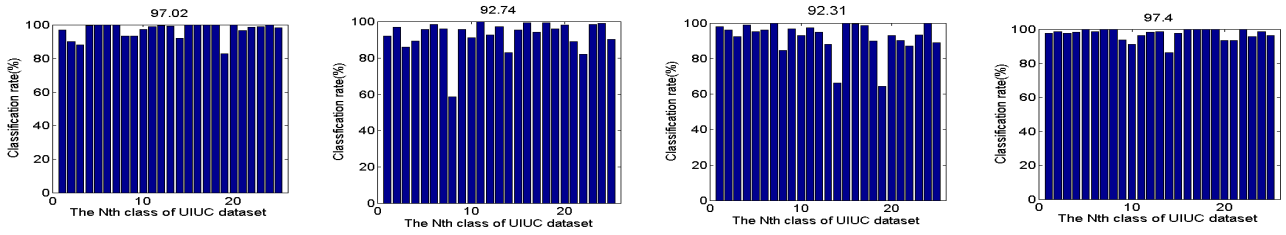


(b)



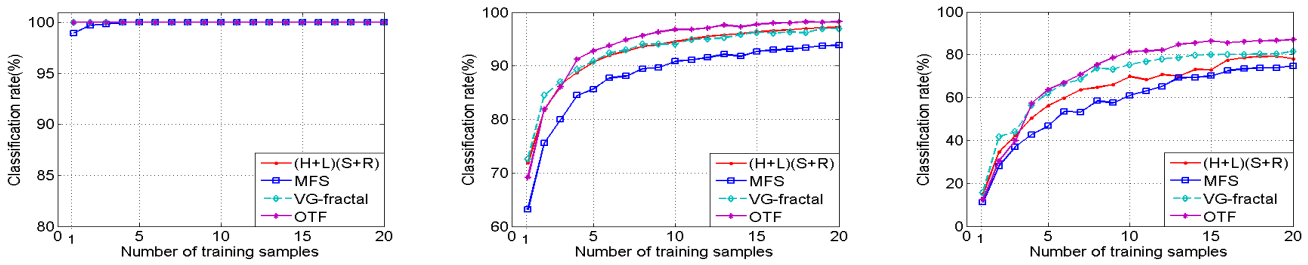
(c)

Figure 7. Classification rate vs. number of training samples on UIUC dataset based on SVM classifier. Four methods are compared: the (H+L)(S+R) method in Lazebnik *et al*. [7], the MFS method in Xu *et al*. [21], the VG-Fractal method in Varma *et al*. [18] and our OTF method. (a) Classification rate for the best class. (b) Mean classification rate for all 25 classes. (c) Classification rate for the worst class.



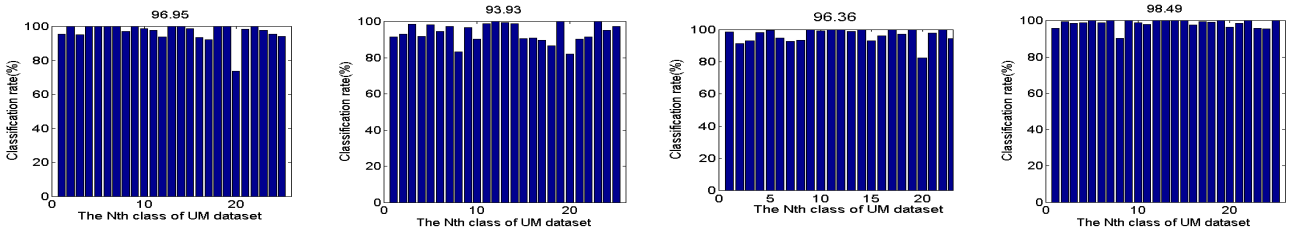
(a) (b) (c) (d)

Figure 8. Classification percentage vs. index of classes on UIUC dataset based on SVM classifier. The number of training samples is 20. The number on the top of each sub-figure is the average classification percentage of all 25 classes. (a) Result of the (H+L)(S+R) method. (b) Result of the MFS method. (c) Result of the VG-Fractal method. (d) Result of our OTF method.



(a) (b) (c)

Figure 9. Classification rate vs. number of training samples on UMD dataset based on SVM classifier. Four methods are compared: the (H+L)(S+R) method, the MFS method, the VG-Fractal method and our OTF method. (a) Classification rate for the best class. (b) Mean classification rate for all 25 classes. (c) Classification rate for the worst class.



(a) (b) (c) (d)

Figure 10. Classification percentage vs. index of classes on UMD dataset based on SVM classifier. The number of training samples is 20. The number on the top of each sub-figure is the average classification percentage of all 25 classes. (a) Result of the (H+L)(S+R) method. (b) Result of the MFS method. (c) Result of the VG-Fractal method. (d) Result of our OTF method.