Scale-space texture description on SIFT-like textons

Yong Xu[a], Sibin Huang[b], Hui Ji[b], Cornelia Fermüller[c]

[a] *School of Computer Science & Engineering, Southern China Univ. of Tech., Guangzhou, 510640, China*
[b] *Department of Mathematics, National University of Singapore, Singapore 117543*
[c] *Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, U.S.A.*

Abstract

Image texture is a powerful cue for the semantic description of scene structures that exhibit a high degree of similarity in their image intensity patterns. This paper describes a statistical approach to image texture description that combines a highly discriminative local feature descriptor with a powerful global statistical descriptor. Based upon a SIFT-like feature descriptor densely estimated at multiple window sizes, a statistical descriptor, called the multifractal spectrum (MFS), extracts the power-law behavior of the local feature distributions over scale. Through this combination strong robustness to environmental changes including both geometric and photometric transformations is achieved. Furthermore, to increase the robustness to changes in scale, a multi-scale representation of the multi-fractal spectra under a wavelet tight frame system is derived. The proposed statistical approach is applicable to both static and dynamic textures. Experiments showed that the proposed approach outperforms existing static texture classification methods and is comparable to the top dynamic texture classification techniques.

Key words: Texture, multi-fractal analysis, image feature, wavelet tight frame

1. Introduction

Image texture has been found a powerful cue for characterizing structures in the scene, which give rise to image patterns that exhibit a high degree of similarity. Classically image texture was used for classification of materials, such as cotton, leather, or wood, and more recently it has been used also on unstructured parts of the scene, such as forests, buildings, grass, trees, or shelves in a department store. An image texture descriptor becomes useful for semantic description and classification, if it is highly discriminative and at the same time robust to environmental changes ([45]). Environmental changes can be due to a wide range of factors, such as illumination changes, occlusions, non-rigid surface distortions and camera viewpoint changes.

Starting with the seminal work of [17], texture has been studied in the context of various applications ([13]). Earlier work was concerned with shape from texture (e.g.

Email addresses: yxu@scut.edu.cn (Yong Xu), maths@nus.edu.sg (Sibin Huang), matjh@nus.edu.sg (Hui Ji), fer@umiacs.umd.edu (Cornelia Fermüller)

[1, 14, 27]), and most of the recent works are about developing efficient texture representations for the purpose of segmentation, classification, or synthesis. There are two components to texture representations: statistical models and local feature measurements. Some widely used statistical models include Markov random fields (e.g. [11, 39]), joint distributions, and co-occurrence statistics (e.g. [18, 19, 31]). Local measurements range from pixel values over simple edge responses to local feature descriptors and filter bank responses (e.g. [6], [16], [20], [21], [25], [26], [29], [39], [41], [42]).

Approaches employing sophisticated local descriptors usually compute as statistics various texton histograms based on some appearance based dictionary. Depending on the percentage of pixel information used in the description, these approaches can be classified into two categories: *dense* approaches and *sparse* approaches. Dense approaches apply appearance descriptors to every pixel. For example, Varma et al [39] used the responses of the MR8 filter bank, consisting of a Gaussian, a LOG filter and edges in different directions at a few scales. In contrast, sparse approaches employ appearance-based feature descriptors at a sparse set of interest points. For example, Lazebnik et al [20] obtained impressive results by combining Harris & Laplacian keypoint detectors and RIFT & Spin image affine-invariant appearance descriptors. Both, the sparse and dense approaches have advantages and disadvantages. The sparse approaches achieve robustness to environmental changes because the features are normalized. However, they may lose some important texture primitives by using only a small percentage of the pixels. Also, there are stability and repeatability issues with the keypoint detection of existing point or region detectors. By using all pixels, the dense approaches provide rich information for local texture characterizations. However, on the negative side the resulting descriptions tend to be more sensitive to significant environmental changes. To address the sensitivity, novel adaptive region processing is needed, which however works well only for sparse sets of image points.

In order to achieve good robustness necessary for semantic classification, both components of texture description, the local appearance descriptors and the global statistical characterization, should accommodate environmental changes. In the past, very robust local feature descriptors have been developed, such as the widely used SIFT features ([23]). Most approaches making use of these feature points use the histogram for global statistical characterization. However, the histogram is not invariant to global geometrical changes. Furthermore, important information about the spatial arrangement of local features is lost. An interesting statistical tool, the so-called MFS (multi-fractal spectra) was proposed in [42] as an alternative to the histogram. The advantage of the MFS is that it is theoretically invariant to any smooth transform (bi-Lipschitz geometrical transforms), and it also encodes additional information regarding the regularization of the spatial distribution of pixels. A similar concept was used also in other texture applications, for example in texture segmentation [5]. In [42] the MFS was applied to simple local measurements, the so-called *local density function*. Although the MFS descriptor proposed in [42] has been demonstrated to have strong robustness to a wide range of geometrical changes including viewpoint changes and non-rigid surface changes, its robustness to photometric changes is weak. The main reason is that the local feature description is quite sensitive to photometric changes. Moreover, the simple local measurements have limited discriminative information. On the other hand, local feature

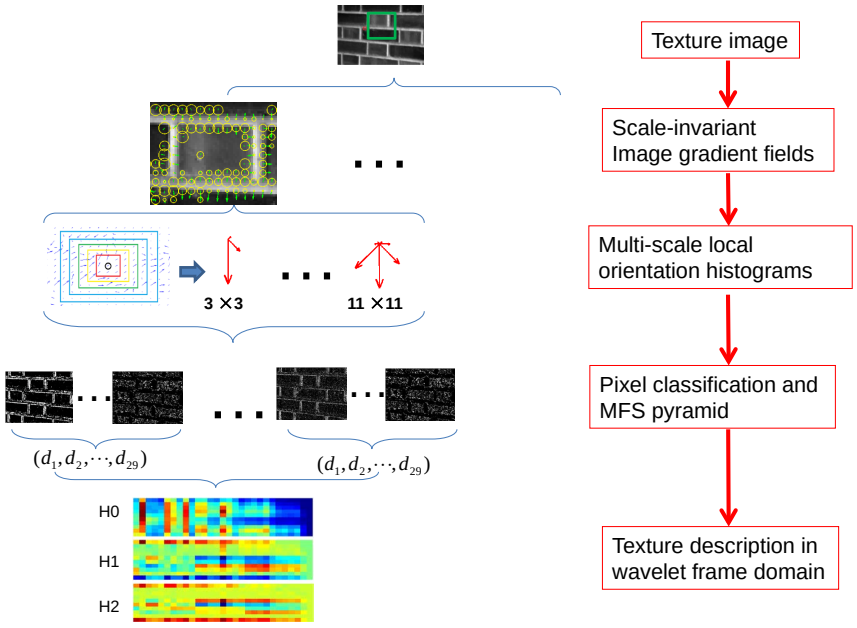


Figure 1: Outline of the proposed approach

descriptors, such as SIFT [23], have strong robustness to photometric changes as has been demonstrated in many applications. In particular, the gradient orientation histogram used in SIFT and variations of SIFT has been widely used in many recognition and classification tasks including texture classification (e.g. [20]).

Here we propose a new statistical framework that combines the global MFS statistical measurement and local feature descriptors using the gradient orientation histogram. Such a combination will lead to a powerful texture descriptor with strong robustness to both geometric and photometric variations. Fig. 1 gives an outline of the approach. First, four sets of scale-invariant image gradients are derived based on a modification of the scale-selection method introduced in [22]. Next, for each set of scale-invariant image gradients, at every pixel a multi-scale gradient orientation histogram is computed with respect to multiple window sizes. Then, using a rotation-invariant pixel classification scheme defined on the orientation histograms, pixels are categorized, and the MFS is computed for every window size. The MFSs corresponding to different window sizes together make up an MFS pyramid. The final texture descriptor is derived by sampling the leading coefficients (that is, coefficients of large magnitude) of the MFS pyramids under a tight wavelet frame transform ([7]).

The rest of the paper is organized as follows. Section 2 gives a brief review of the basic tools used in our approach. Section 3 presents the algorithm in detail, and Section 4 is devoted to experiments on static and dynamic texture classification. Section 5 concludes the paper.

2. Preliminaries

In this section, we give a brief review on some of the tools used in our approach: the multi-fractal analysis and the tight framelet system.

2.1. Multi-fractal analysis

Multi-fractal analysis ([12]) is built upon the concept of the *fractal dimension*, which is defined on point sets. Consider a set of points E in the 2D image plane with same value of some attribute, e.g., the set of image points with same brightness. The fractal dimension of such a point set E is a statistical measurement that characterizes how the points in E are distributed over the image plane when one zooms into finer scales. One definition of the fractal dimension, associated with a relatively simple numerical algorithm, is the so-called *box-counting* fractal dimension, which is as follows: Let the image plane be covered by a mesh of $n \times n$ squares. Let $\#(E, \frac{1}{n})$ be the number of squares that intersect the point set E. Then the fractal dimension, denoted as $\dim(E)$, is defined as

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

In other words, the fractal dimension $\dim(E)$ measures the *power law* behavior of the spatial distribution of E over the scale $1/n$:

$$\#(E, \frac{1}{n}) \propto (1/n)^{-\dim(E)}.$$

In a practical implementation, the value of n is bounded by the image resolution, and $\dim(E)$ is approximated by the slope of the line fitted to

$$\log \#(E, \frac{i}{N}) \quad \text{with respect to } -\log \frac{i}{N} \quad \text{for } i = 1, 2, \ldots, m,$$

with N denoting the image resolution. In our implementation we set $m = 3$ and use the least squares method to estimate the slope.

Multi-fractal analysis generalizes the concept of the fractal dimension. One approach of applying multi-fractal analysis to images is is to classify the pixels in the image into multiple point sets according to some associated pixel attribute α. For each value of α in its feasible discretized domain, let $E(\alpha)$ be the collection of all points with the same attribute value α. The MFS of E then is defined as the vector $D(\alpha)$ vs α, where $D(\alpha)$ is the box-counting fractal dimension of the point set $E(\alpha)$. For example, in [42] the density function (a function describing the local change of the intensity over scale) was used as the pixel attribute. The density was quantized into n values, and the fractal dimensions these n values are combined into the MFS vector.

2.2. Wavelet frame system

Instead of directly using the MFS vector as the texture descriptor, we decompose it under a shift-invariant wavelet frame system and only take the leading wavelet coefficients (coefficients with large magnitude). The reason for doing so is to further

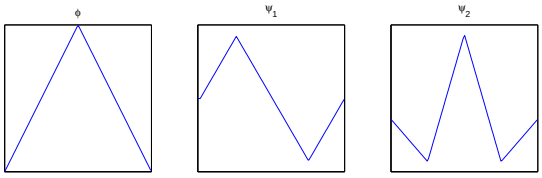


Figure 2: Piecewise linear wavelet frame system ([7]).

increase the robustness of the resulting texture descriptor by removing in-significant coefficients which are sensitive to environmental changes. In this section, we give a brief review on wavelet frame systems. For in-depth theoretical analysis and practical implementation, see for example [7, 3, 36].

A wavelet frame system is a redundant system that generalizes the orthonormal wavelet basis (see [7] for more details). Wavelet tight frames have greater flexibility than orthonormal bases by sacrificing orthonormality and linear independence, but they have the same efficient decomposition and reconstruction algorithms as orthonormal wavelet bases. The filters used in wavelet frame systems have many attractive properties, not present in those used in orthonormal wavelet systems: *e.g.*, symmetry (anti-symmetry), smoothness,and shorter support. These nice properties make wavelet frame systems ideal for building a descriptors with strong robustness.

An MRA-based wavelet frame system is based on a single scaling function $\phi \in L^2(\mathbb{R})$ and several wavelet functions $\{\psi_1, \ldots, \psi_r\} \subset L^2(\mathbb{R})$ that satisfy the following refinable equation:

$$\phi(t) = \sqrt{2}\sum_k h_0(k)\phi(2t-k); \quad \psi_\ell(t) = \sqrt{2}\sum_k h_\ell(k)\phi(2t-k),\ \ell = 1, 2, \ldots, r.$$

Let $\phi_k(t) = \phi(t-k)$ and $\psi_{k,j,\ell} = \psi_\ell(2^j t - k)$. Then for any square integrable function $f \in L^2(\mathbb{R})$, we have a multi-scale representation of f as follows:

$$f = \sum_{k=-\infty}^{\infty} c_k \phi_k(t) + \sum_{\ell=1}^{r}\sum_{j=0}^{\infty}\sum_{k=-\infty}^{\infty} d_{k,j,\ell}\psi_{k,j,\ell}, \tag{1}$$

where $c_k = \int_{\mathbb{R}} f(t)\phi_k(t)dt$ and $d_{k,j,\ell} = \int_{\mathbb{R}} f(t)\psi_{k,j,\ell}(t)dt$. The equation above is called the perfect reconstruction property of wavelet tight frames. The coefficients $\{c_k\}$ and $\{d_{k,j,\ell}\}$ are called low-pass and high-pass wavelet coefficients. The wavelet coefficients can be efficiently calculated by a so-called cascade algorithm (see e.g. [24]). In this paper, we use the piece-wise linear wavelet frame developed in ([7]):

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

See Fig. 2 for the corresponding ϕ and ψ_1, ψ_2. We follow [3] for a discrete implementation of the multi-scale tight frame decomposition without downsampling. For convenience of notation, we denote such a linear frame decomposition by a rectangular

matrix A of size $m \times n$ with $m > n$. Thus, given any signal $\mathbf{f} \in \mathbb{R}^n$, the discrete version of (1) is expressed as follows:

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

where $\mathbf{w} \in \mathbb{R}^m$ is the wavelet coefficient vector of $\mathbf{f}$. It is noted that we have $A^T A = I$ but $AA^T \neq I$ unless the tight framelet system degenerates to an orthonormal wavelet system.

3. The components of the texture descriptor

Our algorithm, which takes as input a texture image, consists of four computational steps:

1. The first step is to calculate four types of scale-invariant image gradients in the scale-space of the texture image. The scale selection used for computing image gradients is based on two measures of images in scale-space: the Harris measure and the Laplacian measure. For each measure, the scale is determined by the maximum or the minimum of the measure over scale. This results in four scale-invariant image gradient fields for the given texture image.

2. Next, for each scale-invariant image gradient field, the local orientation histogram of every pixel is computed over m windows sizes ($m = 8$ in our implementation). Similar as in the SIFT feature approach, we use 8 directions in the orientation histogram. Two types of orientation histogram are used: one simply counts the number of edges in a direction and the other uses the summation of edge energy. Thus, in total we obtain $8 * m$ sets of local orientation histograms for the given image.

3. For each of the 8 kinds of orientation histogram, an MFS pyramid is calculated, with the m levels of the pyramid corresponding to the m window sizes. At every level, the orientation histograms are first discretized into n ($n = 29$ in our implementation) sets using rotation-invariant templates, and an MFS vector is computed on this classification. At the end of this step, we have 8 MFS pyramids of size $m \times n$.

4. Finally, a sparse tight framelet coefficient vector of each MFS pyramid is estimated, by keeping only the frame coefficients of largest magnitude and setting to 0 all others. The total dimension of the resulting texture descriptor in our implementation is 1392.

Next, we give a detailed description of every step.

3.1. Scale-invariant image gradient field

The texture measurement of the proposed method is built upon the image gradients of the given image. To suppress variations of image gradients caused by possible scale changes, we compute the image gradients in scale-space. Given an image $I(x, y)$,

its linear scale-space $L(x, y; \sigma)$ is obtained by convolving $I(x, y)$ with an isotropic Gaussian smoothing kernel of standard deviation σ:

$$g(x, y; \sigma) = \frac{1}{2\pi\sigma} e^{-(\frac{x^2+y^2}{2\sigma})}, \tag{2}$$

such that

$$L(x, y; \sigma) = (\, g(\cdot, \cdot; \sigma) * I)(x, y) \tag{3}$$

with a sequence of $\sigma = \{1, \ldots, K\}$ ranging from 1 to K ($K = 10$ in our implementation). At each pixel (x, y), the image gradient is calculated by

$$(\partial_x L(x, y; \sigma_*(x, y)), \quad \partial_y L(x, y; \sigma_*(x, y))),$$

where $\sigma_*(x, y)$ is determined by the scale selection method proposed in [22]. This method selects at every point the scale at which some image measurement takes on the extreme value. In our approach, these are the minimum and maximum of the Harris and the Laplacian measurement.

In more details, for each pixel (x, y) two measurements are calculated: the Harris measurement (4) defined as

$$M_H = L_x^2 L_y^2 - (L_x L_y)^2 - \alpha(L_x^2 + L_y^2)^2 \tag{4}$$

and the Laplacian measurement (5) defined as

$$M_L = \sigma(L_{x^2} + L_{y^2}) \tag{5}$$

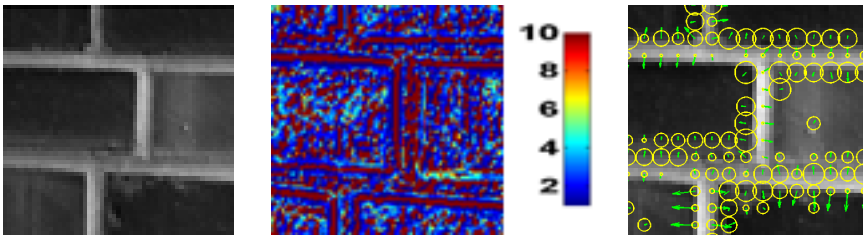
with $L_{x^m y^n}(x, y; t) = g * (\partial_{x^m y^n}(I(x, y)))$. In our implementation, parameter α is set to 0.05 and the Prewitt filters are used for computing the partial derivatives in scale-space, which are given as:

$$\frac{\partial}{\partial_x} : \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}; \quad \frac{\partial}{\partial_y} : \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$

The Harris measurement characterizes the edge energy along different directions. It takes on large values at corners, and small values at strong straight edges. The Laplacian measurement represents the second-order derivative information, and it is large at the center of intensity blobs and small on edges. Then, four scales are derived by taking the maximum/minimum value of the Harris/Laplacian measurement over scale. For each scale-selection method, the gradient magnitude and orientation are computed by applying the finite difference operator to $L(x, y; \sigma_*)$. See Fig. 3 for an illustration the scales selected at each pixel and the corresponding image gradients. At the end of this step, we have four scale-invariant image gradient fields.

3.2. Multi-scale local orientation histograms

Our proposed local feature descriptor relies on the local orientation histogram of image pixels, which also is used in SIFT ([23]) and similar features. Its robustness to illumination changes and invariance to in-plane rotations has been demonstrated



(a) Sample texture region (b) Scale σ (c) Image gradient

Figure 3: (a) Sample texture region; (b) selected scale σ based on the maximum of the Laplacian measure in scale-space with the scale ranging from 1 to 10; (c) corresponding image gradient field, where the circle at a point denotes the size of the Gaussian smoothing kernel (defined by σ) when computing the gradient.

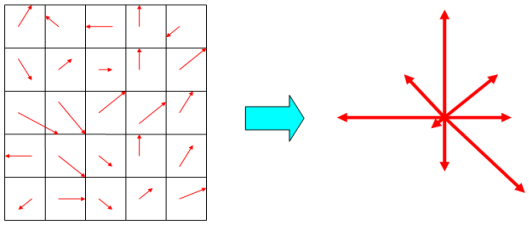


Figure 4: Orientation histogram for a neighborhood size of 5×5.

in many applications. For each image gradient field computed in the previous step, at every pixel, two types of local orientation histograms are computed. One simply counts the number of orientations; the other weighs them by the gradient magnitude. See Fig. 4 for an illustration. The gradient orientations are quantized into 8 directions, covering 45 degrees each. To capture information of pixels in a multi-scale fashion, for each pixel, we compute the orientation histograms at 8 window sizes ranging from 3×3 to 17×17. The orientation histograms, as in SIFT, are rotated to align the dominant orientation with a fiducial direction.

3.3. Pixel classification and the MFS

The next step is to compute the MFS vector. The MFS vector depends on how the pixels are classified. To obtain a reasonable statistics of the spatial distribution of pixels, the number of pixels in each class needs to be sufficiently large, and thus the number of the classes needs to be quite small.We thus need a meaningful way of discretizing the very large amount of possible orientation histograms. Our approach is to introduce a fixed bin partitioning scheme based on a set of basic orientation histogram templates.

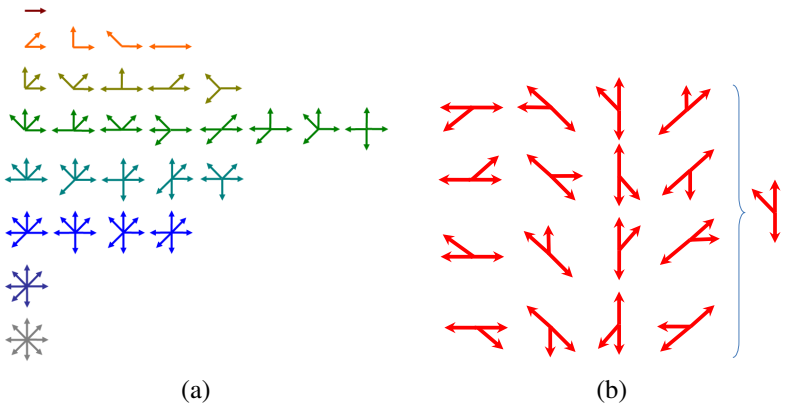


Figure 5: Twenty-nine Orientation histogram templates. (a) one representative element is shown for each class; (b) all the elements in one class, which are obtained from the possible mirror-reflections and rotations of the basic element.

First, the estimated orientation histograms are quantized as follows. For each bin in the orientation histogram, the value is set to 0 if the magnitude is less than $\frac{1}{8}$ of the overall magnitude and to 1 otherwise. We then define a partitioning scheme based on the topological structure of orientation histograms, with a total of 29 classes. Fig. 5 gives an illustration showing one basic element of each class. The proposed templates are defined on the basis of the number of significant image gradient orientations and their relative positions. Each template class contains the basic element shown in Fig. 5 (a) and all rotated and mirror-reflected copies that can be obtained from it (Fig. 5 (b)).

Next, for each window size the corresponding MFS feature vector is calculated as follows: For each template class (out of 29 classes), a binary image is derived by setting the value of the pixel to 1 if its associated template falls into the corresponding template class and to 0 otherwise (see Fig. 6. Thus, there are 29 binaries images. For each binary image the box-counting fractal dimension is computed, and the fractal dimensions are concatenated into a 29-dim MFS vector. The MFS feature vectors corresponding to different windows sizes are then combined into a multi-scale MFS pyramid. The size of this MFS pyramid is 8×29.

It is easy to see that the orientation histogram templates provide a pixel classification scheme which is invariant to rotation and mirror-reflection; in addition, the robustness to illumination changes is guaranteed by the orientation histogram itself ([23]). Using the MFS as the replacement of the histogram for statistical characterization leads to better robustness to global geometric changes.

3.4. Robustifying the texture descriptor in the wavelet frame domain

Recall that given a texture image, we derived eight types of orientation histograms, and each is associated with an MFS pyramid. The final step is to construct the texture descriptor by only taking the leading coefficients of the eight MFS pyramids in a wavelet frame domain. The purpose is to further increase the robustness of the texture

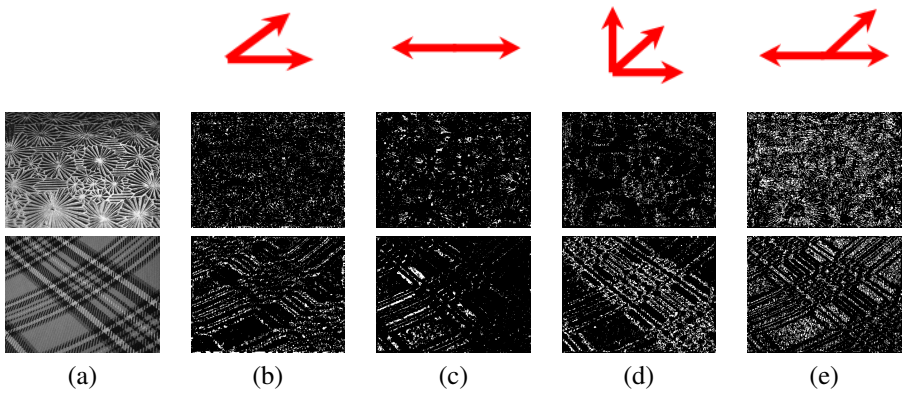


Figure 6: (a): Two texture images. (b)–(e): Examples of binary images with respect to pixel classification based on the orientation histogram templates.

descriptor to environmental changes. The construction is done as follows: We first decompose the MFS pyramid using the 1D un-decimal linear-spline framelet transform ([7]), as it has been empirically observed that the corresponding tight frame coefficients tend to be highly relevant to the essential structure of textures.

Let the matrix $E(s,n)$ denote the MFS pyramid where s denotes the scale (windows size of local orientation histogram) and n denotes the index of the template class. Let $\mathcal{F}$ denote the L-level decomposition of $E(s,n)$ under a 1D tight framelet system with respect to s defined as

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

where A is the frame decomposition operator, and j denotes the level of the frame decomposition. The multi-dimensional matrix $\mathcal{F}$ consists of two kinds of components: one component that is the output of the low-pass filtering using h_0 at the scale 2^{-L}, and multiple components that are the outputs of high pass filtering using $h_1, \cdots, h_r$ at multiple levels ranging from $2^{-1}, \cdots, 2^{-L}$. Each high-pass filter output has three variables: scale $2^{-j}, j = 1, \cdots, L$, level $s, s = 1, \cdots, 8$ and bin index $n, n = 1, 2, \cdots, 29$. See Fig. 7 for an illustration of the single level frame coefficients of the sample images in Fig. 6.

Recall that the un-decimal framelet tight frame is a redundant transform, and thus there is redundant information in the framelet coefficients of $\mathcal{F}$. In contrast to orthogonal mappings, redundant transforms tend to yield sparse leading coefficients with large magnitude. The next step then involves extracting these leading coefficients such that the resulting descriptor is compact and provides strong robustness to inter-class texture variations. In our approach, we simply keep the 70% leading coefficients with largest amplitude and set all others to 0. The final texture descriptor then consists of only leading framelet coefficients of all MFS pyramids.

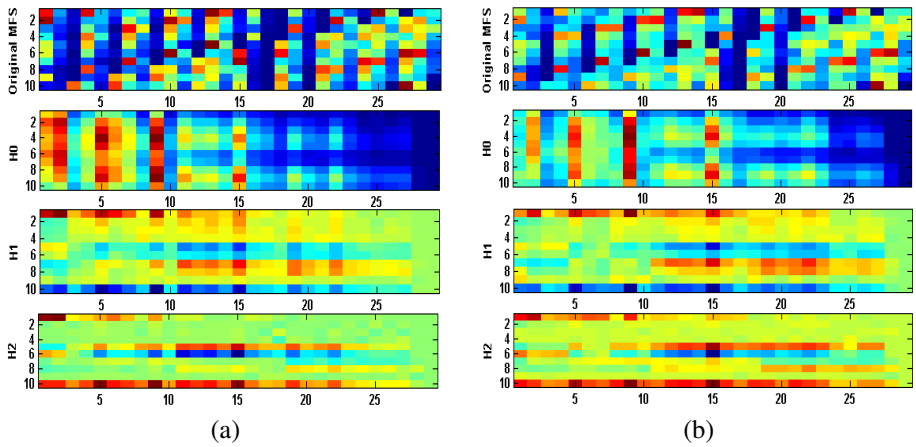


Figure 7: Framelet decomposition at a single scale. The low-pass components of the frame coefficient H_0 obtained by filtering with h_0. The high-pass components of the frame coefficients $\{H_1, H_2\}$ obtained by filtering with h_1 and h_2 respectively. (a): Framelet features of the glass image in Fig. 6 for max of Laplacian measure; (b): Framelet features of the plaid images in Fig. 6 for max of Laplacian measure

4. Experimental evaluation

The performance of the proposed texture description is evaluated for static and dynamic texture classification.

4.1. Static texture

We evaluated the performance of texture classification on two datasets, the UIUC dataset ([23]) and the high-resolution UMD dataset ([42]). Sample images of these datasets are shown in Fig. 8 and in Fig. 11. The UIUC texture dataset consists of 1000 uncalibrated and unregistered images: 40 samples for each of 25 textures with a resolution of 640×480 pixels. The UMD texture dataset also consists of 1000 uncalibrated and unregistered images: 40 samples for each of 25 textures with a resolution of 1280×900 pixels. In both datasets significant viewpoint changes and scale differences are present, and the illumination conditions are uncontrolled.

In our 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. A final texture description is based on a two-scale framelet-based representation. The reported classification rate is the average over 200 random subsets. An SVM classifier (Tresp et al [37]) is used, which was implemented as in Pontil et al [32]. 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 [34]. The optimal parameters are discovered by cross-validation.

The proposed texture descriptor is compared against three other texture descriptors: Lazebnik et al [20], Varma et al [38], and Xu et al [42]. he first one ([20]) is the so-called (H+L)(S+R) texture descriptor, which is based on a sophisticated point-based

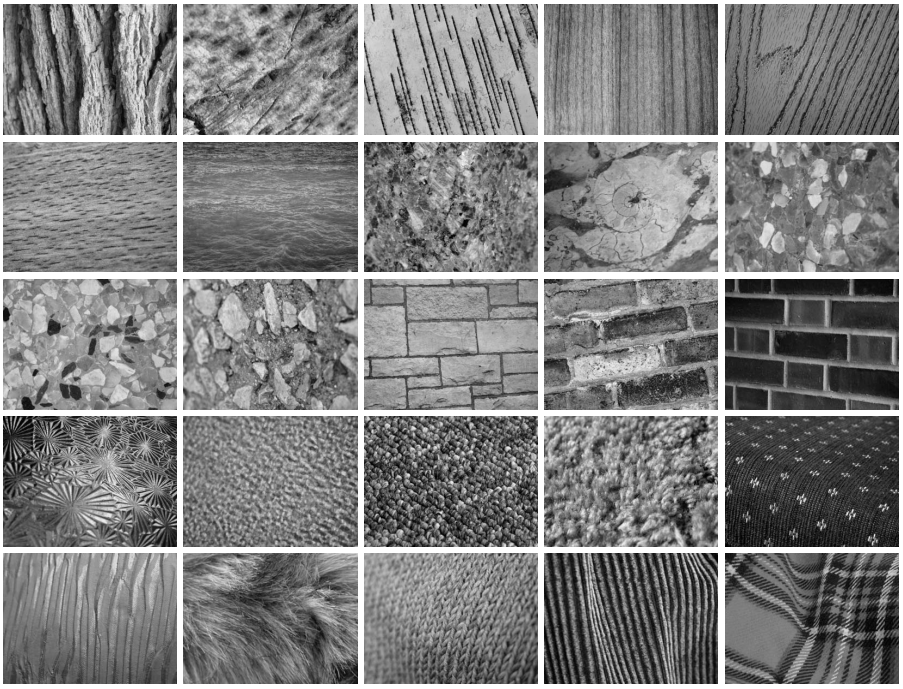


Figure 8: 25 sample textures from the UIUC dataset

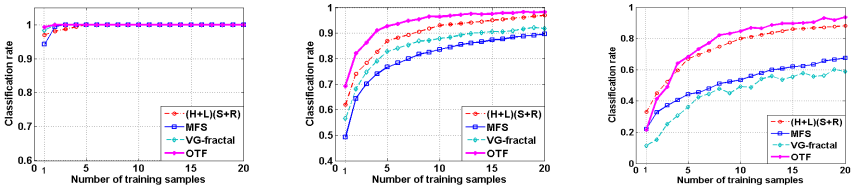


Figure 9: Classification rate vs. number of training samples for the UIUC dataset based on SVM classification. Four methods are compared: the (H+L)(S+R) method in Lazebnik et al[20], the MFS method in Xu et al [42], the VG-Fractal method in Varma et al[38] and our OTF method. (a) Classification rate for the best class. (b) Mean classification rate for all 25 classes. (c) Classification rate of the worst class.

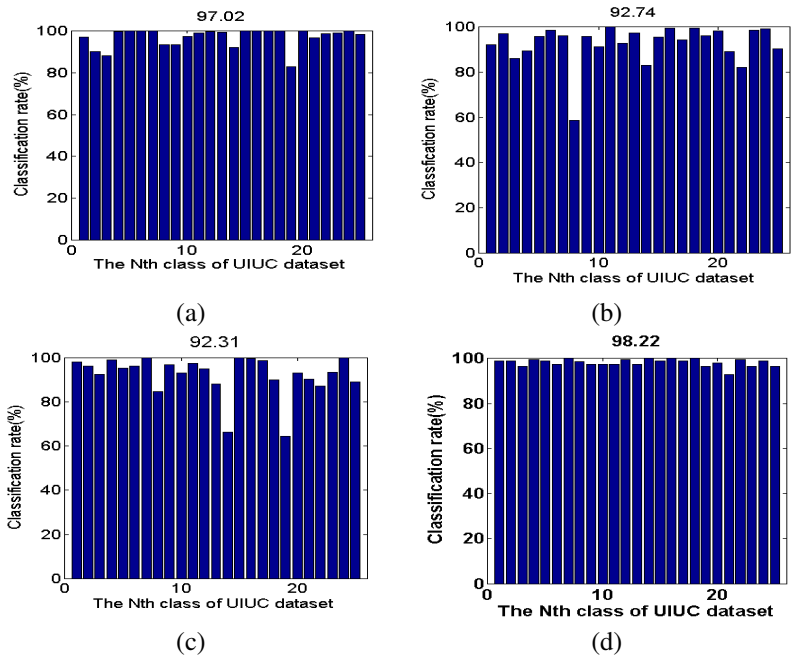


Figure 10: Classification percentage vs. index of classes for the UIUC dataset based on SVM classification. The number of training samples is 20. The number on the top of each sub-figure is the average classification percentage of all 25 classes. (a) Result of the (H+L)(S+R) method. (b) Result of the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

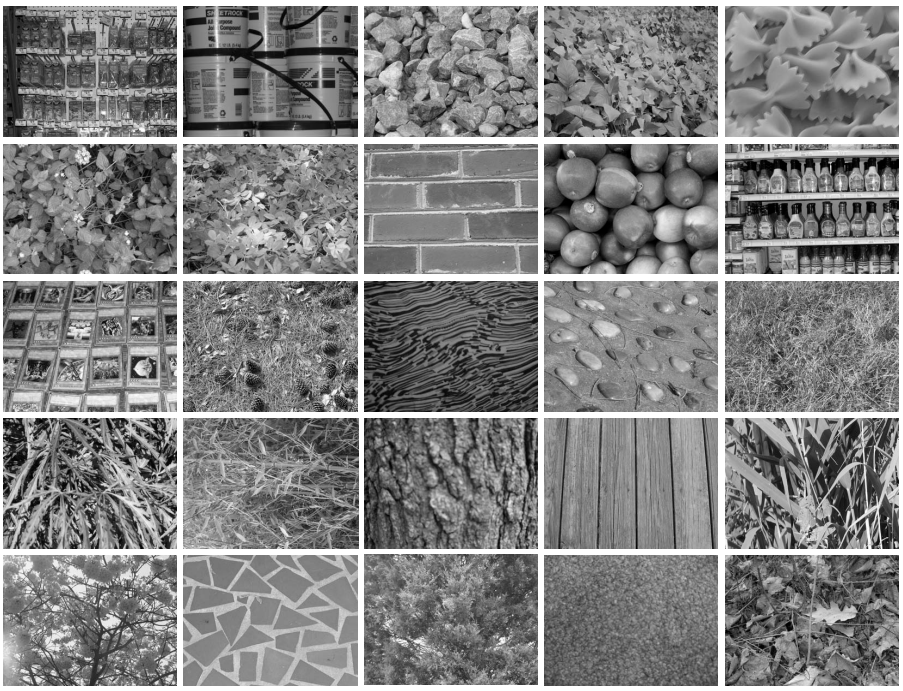
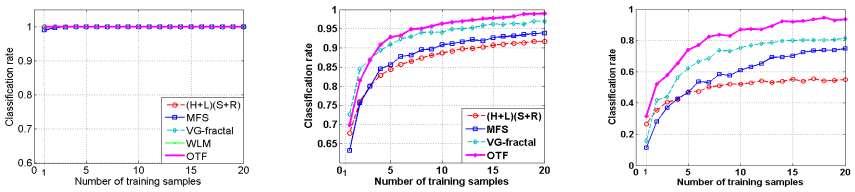


Figure 11: The 25 textures from the UMD dataset.

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 [38], which uses 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, the MFS method by Xu et al [42], derives the MFSs of simple local measurements (the local density function of the intensity, image gradient and image Laplacian). The texture descriptor is a combination of the three MFSs. The results on the UIUC dataset using the SVM classifier for the (H+L)(S+R) method is from [20]. The other results are obtained from our implementations. We denote our approach as OTF method. Fig. 9 shows the classification rate vs. the number of training samples on the UIUC dataset. Fig. 10 shows the classification percentage vs. the index of classes on the UIUC dataset based on 20 training samples. Fig. 12 and Fig. 13 show the results of the UMD dataset using the same experimental evaluation.

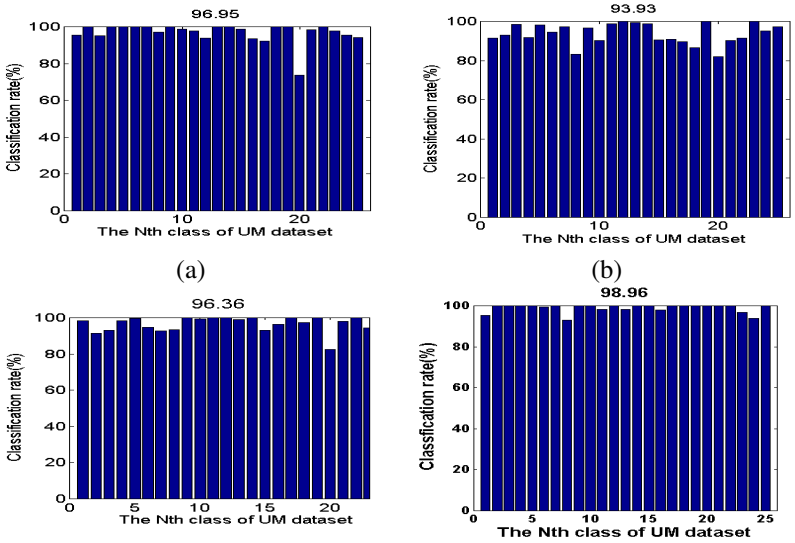
From Fig. 9 – Fig. 13, it is seen that our method clearly outperformed the VG-fractal method and the MFS method on both datasets. Also our method obtained better results

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. In contrast, our approach is much simpler and efficient without requiring clustering.



(a) (b) (c)

Figure 12: Classification rate vs. number of training samples for the UMD dataset using SVM classification. 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 of the worst class;



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

Figure 13: Classification rate (in percentage) vs. index of classes on UMD dataset based on SVM classification. The number of training samples is 20. The number at the top of each sub-figure indicates the average classification rate over all 25 classes. (a) (H+L)(S+R) method. (b) MFS method. (c) VG-Fractal method. (f) OTF method.

4.2. Dynamic texture

Dynamic textures are image sequences with stochastically stable spatiotemporal behavior ([9]). Examples are video sequences of rivers, water, foliage, smoke, clouds,

fire and etc. Also, the applications concerning such video sequences are plenty, including surveillance, foreground and background separation (e.g. [10, 35]). In this section, using a similar concept as for static texture, we develop an efficient texture descriptor for dynamic texture with strong robustness to environmental changes.

The OTF method developed for static texture can be applied to describe dynamic texture without significant modifications. Different from static textures, dynamic textures not only vary in the spatial distribution of texture elements, but also vary in their dynamics over time. The basic idea in our approach is to view the dynamic texture as a 3D volume of data and examine it from three orthogonal views, i.e. the views along two perpendicular spatial axes (x- and y- axis) and the view along the time axis (t-axis). More specifically, for each axis, we apply the OFT method on every image slice of the volume along this axis and take the mean of all feature vectors of all image slices. Then, the texture descriptor for dynamic texture is defined as the weighted sum of the three mean OTF descriptors along the x-axis, y-axis and t-axis with weights being $(0.2, 0.2, 0.6)$ respectively. Also, since most dynamic textures have rather small scale changes, the step of calculating scale-invariant image gradients are omitted for the purpose of computational efficiency. Instead, we just use the standard image gradients.

One of the most popular dynamic texture benchmarks is the UCLA dataset used extensively for performance evaluation (*e.g.* [8, 15, 30, 33, 40]). The original UCLA dataset consists of 50 dynamic textures. Each texture is given in terms of four grayscale image sequences captured from the same viewpoint, resulting in a total of 200 sequences, each of which consists of 75 frames of size 110*160. The literature does not agree on a ground truth regarding the classification of the UCLA dataset. In [8, 33, 40] the following three classifications, termed DT9, SIDT, and DT7 were considered:

1. DT9 is a classification into 9 classes ([33, 15]). The categories contain boiling water(8), fire(8), flowers(12), fountains(20), plant(108), sea(12), smoke(4), water(12) and waterfall(16), where the numbers of elements of each class are given in brackets. Sample frames are shown in Fig. 14. In our experiments we used the original images of size 110*160.

2. SIDT was chosen to eliminate the effects due to biases in identical viewpoint selection. The sequences in the UCLA dataset were manually cropped into non-overlapping pairs of subsequences with a spatial resolution of 48*48 ([40]), resulting in a total of 400 sequences. Nearest-neighbor classification was applied in the recognition process.

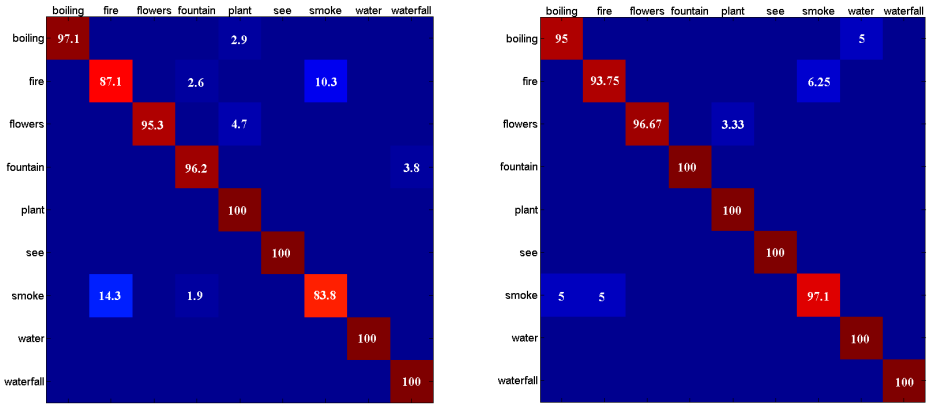
3. DT7 splits the original images spatially into left and a right halves resulting in 400 sequences, which were classified into seven different semantic categories [8] as follows: flames(16), fountain(8), smoke(8), turbulence(40), waves(24), waterfall(64), vegetation(240).

We compared our method method using both NN and SVM classifiers to the methods in [8] and [15] on the three categorizations (DT9, SIDT and DT7). See table 1 for a comparison of the methods. The confusion matrices for *DT9*, *SIDT* and *DT7* are shown in Fig. 15, Fig. 16 and Fig. 17 respectively. It can be seen that our methods compares favorably to the other two state-of-the-art methods.

Figure 14: Samples images from dynamic textures in the UCLA 9 dataset.

Method	DT9	SIDT	DT7
Derpanis et al[8]		60%(NN)	92.3%(NN)
Ghanem et al[15]	95.6%(SVM)		
OTF descriptor (NN)	95.32%	81.73%	95.48%
OTF descriptor (SVM)	97.15%		98.14%

Table 1: The classification table of DT9, SIDT and DT7



(a) (b)

Figure 15: Confusion matrix of DT9 for classification. (a) Result by NN classifier. (b) Result by SVM classifier.

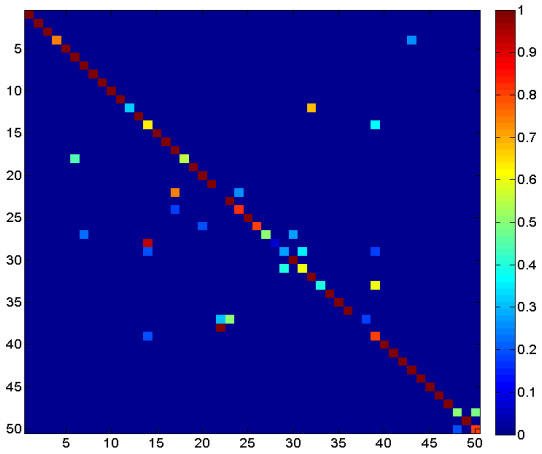
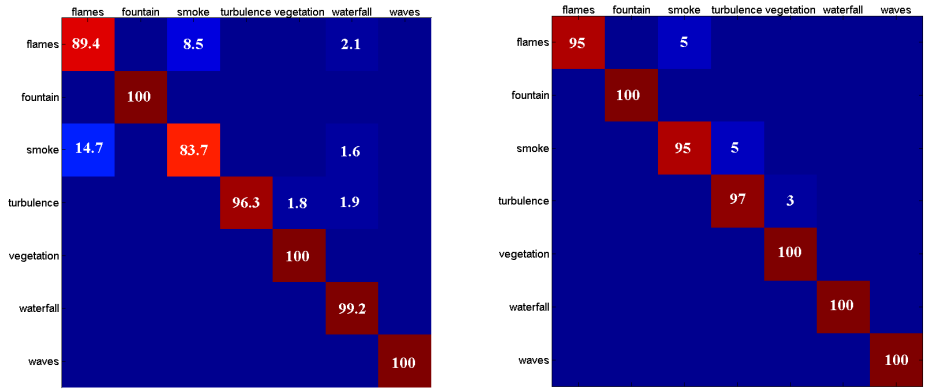


Figure 16: Confusion matrix of SIDT for classification by NN classifier.

5. Summary and conclusions

In this paper, we proposed a new texture descriptor, which applies the global MFS to local gradient orientation histograms. The proposed descriptor has strong robustness to both local and global illumination changes and is robust to many geometric changes. Locally, robustness to illumination changes and geometric variations is achieved by using templates of local gradient orientation histograms; robustness to local scale changes is achieved by using scale-invariant image gradient fields. Globally, the multi-fractal spectrum ([42]) and its sparse approximation in a wavelet frame system are employed



(a) (b)

Figure 17: Confusion matrix of DT7 for classification. (a) Result by NN classifier. (b) Result by SVM classifier.

to obtain further robustness to global environmental changes. Our texture description is rather efficient and simple to compute without feature detection and clustering. Experiments on static and dynamic texture classifications showed that our approach performed well. In future research, we would like to investigate how to apply the proposed framework to other recognition tasks including object recognition and scene understanding.

References

[1] J. Aloimonos, "Shape from texture, *Biological Cybernetics, Vol. 58, pp 345-360, 1988.*

[2] D.J. Heeger and J.R. Bergen, "Pyramid Based Texture Analysis/Synthesis", Computer Graphics Proceedings*, pp 229-238, 1995.*

[3] 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.*

[4] Y.-W. Chen and C.-J. Lin, "Combining SVMs with various feature selection strategies", Feature Extraction, Foundations and Applications*, Springer, 2006.*

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

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

[7] 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.*

[8] K. G. Derpanis and R. P. Wildes, "Dynamic texture recognition based on distributions of spacetime oriented structure", CVPR*, 2010.*

[9] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic texture", IJCV*, 2003.*

[10] G. Doretto, D. Cremers, P. Favaro and S. Soatto, "Dynamic texture segmentation. ICCV*, 2003.*

[11] A. Efros and T. Leung, "Texture synthesis by non-parametric smapling, ICCV, pp. 1039-1046, 1999.

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

[13] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, 2002.

[14] J. Garding and T. Lindeberg, "Direct computation of shape cues using scale-adapted spatial deriative operators", *IJCV*, 17(2), pp. 163-191, 1996.

[15] B. Ghanem and N. Ahuja, "Maximum margin distance learning for dynamic texture recognition", *ECCV*, 2010.

[16] 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.

[17] B. Julesz, "Texture and visual perception, *Science America*, 212, pp. 38-48, 1965.

[18] C. Kervrann and F. Heitz, "A Markov random field model-based approach to unsupervised texture segmentation using local and global spatial statistics, *IEEE Trans. on Image Process, Vol. 4(6), pp. 856-862, 1995.*

[19] S. M. Konishi and A.L. Yuille, "Statistical cues for domain specific image segmentation with performance analysis, CVPR, pp. 1125-1132, 2000.

[20] S. Lazebnik, "Local semi-local and global models for texture, object and scene recognition", *Ph.D. Dissertation, University of Illinois at Urbana-Champaign*, 2006.

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

[22] T. Lindeberg, "Automatic scale selection as a pre-processing stage for interpreting the visual world", *FSPIPA*, 130, pp. 9-23, 1999.

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

[24] S. Mallat, *A Wavelet Tour of Singapore Processing,* Third Edition: The Sparse Way, Academic Press, 2008.

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

[26] 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.

[27] J. Malik and R. Rosenholtz, "Computing local surface orientation and shape from texture for curved surfaces, *IJCV*, Vol. 23(2), pp. 149-168, 1997.

[28] K. Mikolajczyk and C. Schmid. "Scale and affine invariant interest point detectors". *IJCV*, 60(1), pp. 63-86, 2004.

[29] 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.

[30] P. Saisan, G. Doretto, Y. Wu, and S. Soatto, "Dynamic texture recognition", *CVPR*, II, pp. 58–63, 2001.

[31] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients, *IJCV, Vol. 40(1), pg. 49-71, 2000.*

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

[33] A. Ravichandran, R. Chaudhry, and R. Vidal, "View-invariant dynamic texture recognition using a bag of dynamical systems", CVPR*, 2009.*

[34] B. Scholkopf and A. Smola, Learning with kernels: Support Vector Machines, regularization, optimization and beyond*, MIT Press, Cambridge, MA, 2002.*

[35] J. R. Smith, C. Y. Lin and M. Naphade. "Video indexing using spatio-temporal wavelets," ICIP*, 2002.*

[36] Z. Shen, "Wavelet frames and image restorations", Proceedings of the International Congress of Mathematicians*, pp. 2834–2863. 2010.*

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

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

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

[40] F. Woolfe and A. Fitzgibbon, "Shift-invariant dynamic texture recognition", ECCV*, II, pp. 549–562, 2006.*

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

[42] Y. Xu, H. Ji and C. Fermuller, "Viewpoint invariant texture description using fractal analysis," IJCV*, 83 (1), pp. 85–100, 2009.*

[43] Y. Xu, X. Yang, H. B. Ling and H. Ji, "A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid", CVPR*, 2010.*

[44] 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. 213–238, 2007.*

[45] 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.*