Viewpoint invariant texture description using fractal analysis

Yong Xu[1]**, Hui Ji**[2]**, Cornelia Fermüller**[3]

[1] School of Computer Science & Engineering, South China University of Technology, Guangzhou, 510640, China.
[2] Department of Mathematics, National University of Singapore, Singapore, 117643.
[3] Center for Automation Research, University of Maryland, College Park, MD 20742-3275

Abstract   Image texture provides a rich visual description of the surfaces in the scene. Many texture signatures based on various statistical descriptions and various local measurements have been developed. Existing signatures, in general, are not invariant to 3D geometric transformations, which is a serious limitation for many applications. In this paper we introduce a new texture signature, called the multi-fractal spectrum (MFS). The MFS is invariant under the bi-Lipschitz map, which includes view-point changes and non-rigid deformations of the texture surface, as well as local affine illumination changes. It provides an efficient framework combining global spatial invariance and local robust measurements. Intuitively, the MFS could be viewed as a "better histogram" with greater robustness to various environmental changes and the advantage of capturing some geometrical distribution information encoded in the texture. Experiments demonstrate that the MFS codes the essential structure of textures with very low dimension, and thus represents an useful tool for texture classification.

1 Introduction

Image descriptors that capture essential properties of surfaces and are also invariant to environmental changes would constitute ideal tools for describing, matching

and recognizing these surfaces. In the Computer Vision literature the search for invariants started in the nineties (Weiss 1993), when researchers dug up the mathematical literature on algebraic and geometric invariants. Great importance has been given to the study of quantities which are invariant to the viewpoint from which the image is taken, as such quantities would facilitate the matching of surfaces. A number of projective invariants (Mundy and Zisserman 1992) have been found, which are defined on sets of points and lines and planar curves (Weiss 1993), and they have been used for object recognition and calibration. However, none of these descriptors is concerned with texture.

Computational studies on texture started with the seminal work of (Julesz 1965). Since then texture has been studied in the context of four major applications (Forsyth and Ponce 2002). Earlier work was concerned with shape from texture (Garding and Lindeberg 1996; Malik and Rosenholtz 1997; Bajcsy and Lieberman 1976; Aloimonos 1988), that is the recovery of surface orientation from texture. Most recent studies focus on the representations of texture for the purpose of segmentation, synthesis and classification. These representations are due to two factors: statistical models and local filter bank responses, and most descriptors combine both. The successful statistical models are Markov random fields (Efros and Leung 1999; Varma and Zisserman 2003), joint distribution and co-occurrence of pixels or filter responses (Kervrann and Heitz 1995; Konishi and Yuille 2000; Azencott et al. 1997; Portilla and Simoncelli 2000; Teuner et al. 1995). Often many filter or wavelet responses (Mallat 1989; Laine and Fan 1993) are obtained at every point or patch in the image from training images, and response distributions of training images are learned from histograms and clusters of the filter responses. Synthesis can then be achieved by constructing textures with the same distribution, segmentation by assigning class-membership, and classification by comparing a novel texture with the distributions of the training images.

Most texture descriptors are sensitive to changes in viewpoint and lighting. However, the importance of creating robust descriptions has been recognized (Zhu et al. 1998). A number of researchers have proposed to build representation by combining images from many views or illumination directions, which have been registered (Dana and Nayar 1998; Liu et al. 2001; Zalesny and Van Gool 2001; Neubeck 2004; Leung and Malik 2001; Chantler et al. 1988). Closest related to this work are a number of studies, which achieve robustness to view-point changes by using local descriptors or filter responses, which are rotationally or locally affine invariant (Cula and Dana 2001; Schmid 2001; Varma and Zisserman 2002, 2003). Particularly good results have been achieved with the method in (Lazebnik et al. 2005), which does not extract measurements everywhere, but only at spatial locations found by interest operators.

Here we introduce a novel texture descriptor called the MFS (multifractal spectrum vector), which is based on fractal geometry theory. The MFS is globally invariant under the *bi-Lipschitz transform* (see Equation 8), a very general transform which includes perspective transforms (viewpoint changes) and general texture surface deformations. Furthermore, the MFS has low dimension, and is very efficient to compute. We demonstrate with a number of experiments on synthesized and real image textures that, first the MFS in practice is very robust to geometric

3D transforms and non-rigid smooth transforms. Second, its performance is similar to the top methods in traditional texture retrieval and classification on standard texture datasets, and better on the high resolution dataset.

Fractal geometry has been used before in the description and classification of textures (Espinal et al. 1998; Peleg 1984; Vehel et al. 1992), for texture segmentation (Conci and Monteiro 2000; Kan and Blac-Talon 2000; Kaplan 1999; Chaudhuri and Sarkar 1995), as well as synthesis (Kaplan and Kuo 1995). However, the invariance of the fractal dimension to bi-Lipschitz maps has not been utilized in the vision community. Existing approaches, which globally use fractal geometry, consider the image intensity as a surface and compute its fractal dimension in 3D (Peleg 1984), or they describe this surface as generated by fractal Brownian motion (Kaplan 1999) and expressed by the so-called Hurst parameter. In either case the fractal texture is parameterized by only a scalar number, which is not enough for a good characterization of the texture. Actually most fractal-based approaches describe the texture by only a few numbers. For example, (Chaudhuri and Sarkar 1995) computes the fractal dimensions of a few filtered and intensity modified versions of the image, and (Peleg 1984) computes the logarithmic texture parameters at multiple scales instead of enforcing the fractal behavior on all scales. For those Methods which utilize the multi-fractal-spectrum (Kan and Blac-Talon 2000; Xia et al. 2006), with the exception of (Vehel et al. 1992), it is only used for local feature description, in which case there is no global invariance to bi-Lipschitz maps. Moreover, in previous work the fractal dimension was defined only on the image intensity, which makes it sensitive to changes in illumination.

In our approach, we look at the image as an union of point sets. The point sets are the level sets under a level set categorization of image points. Intuitively, the MFS is the extension of the fractal dimension. The MFS encodes the fractal dimension of every level set. By choosing different categorizations of image points different MFS vectors are obtained. The categorizations in our MFS approach are based on the *density function* (also referred to as Hölder exponent or local fractal dimension (Varma and Garg 2007)) defined on (functions of) the image intensity, and this makes it more robust to variations in illumination.

The paper is organized as follows. Section 2 gives a brief introduction to fractal geometry and MFS theory. Section 3 presents our MFS texture descriptor and its implementation. Section 4 shows the MFS's invariance to spatial bi-Lipschitz transforms and local affine illumination changes and discusses the MFS in comparison to the histogram. Section 5 provides experiments on texture retrieval and classification and a comparison to other methods. Section 6 presents the conclusions and discusses future work.

A preliminary version of this work has appeared in (Xu et al. 2006). The new paper has a different implementation of the MFS. Instead of the moment-based approach (Appendix A), a direct method (Sec. 3.4) is employed for better performance. Furthermore, in the experiments on classification (Sec. 5), a new SVM based procedure has been included to automatically select the optimal subset of the original MFS, leading to improved results.
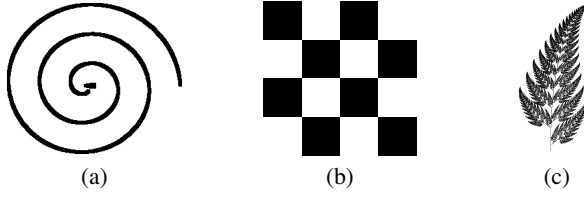
<center>(a)   (b)   (c)</center>

Fig. 1 Fractal dimension D in R^2. (a) Smooth spiral curve with $D = 1$. (b) The checkerboard with $D = 2$. (c) Fractal fern leaf with $D \approx 1.7$.

2 Fractal Theory and the MFS

We first give a brief introduction to fractal theory. In the late eighties, it has been realized that irregular objects provide a much better representation of many natural phenomena than do the regular objects of classical geometry. Fractal geometry was developed, which provides a general framework for studying irregular sets as well as regular sets.

The fractal dimension is the key quantity in the study of fractal objects. Fundamental to the fractal dimension is the concept of "measurement at scale δ". For each δ, we measure an object in a way that ignores irregularity of size less than δ, and we see how these measurements behave as δ goes to 0. Scientists found that most of the natural phenomena satisfy the power law, which states that the estimated quantity (for example the length of a coastline (Mandelbrot 1982)), is proportional to $\left(\frac{1}{\delta}\right)^D$ for some D. For most natural objects, D is almost the same for small scales δ. Thus we can compute its limit, which is called the *fractal dimension*.

In our formulation we consider objects in R^2. Considering an irregular point set E defined on R^2, the fractal dimension of E is defined as

$$\dim(E) = \lim_{\delta \to 0} \frac{\log N(\delta, E)}{-\log \delta}, \tag{1}$$

where $N(\delta, E)$ is the smallest number of sets of diameter less than δ that cover E. The set is made up of closed disks of radius δ or squares of side δ. In practical computations, one usually divides up the space with a mesh of boxes of size δ, called the δ-mesh boxes (δ-squares in two dimensions), and counts the boxes occupied by the point set. The fractal dimension computed this way is called box counting dimension.

Intuitively, the fractal dimension is a statistical quantity that gives a global description of how complex or how irregular a geometric object is. The fractal dimension D of any object in R^2 is in the range of $0 \leq D \leq 2$. A point has a fractal dimension of 0, any smooth curve has a fractal dimension of 1, and a completely filled rectangle has a fractal dimension of 2, which are the same as their integer topological dimensions. Irregular sets have a fractional dimension between 0 and 2. Indeed, most man-made geometric objects have an integer fractal dimension, while most objects in nature have a fractional fractal dimension. An illustration is given in Fig. 1.
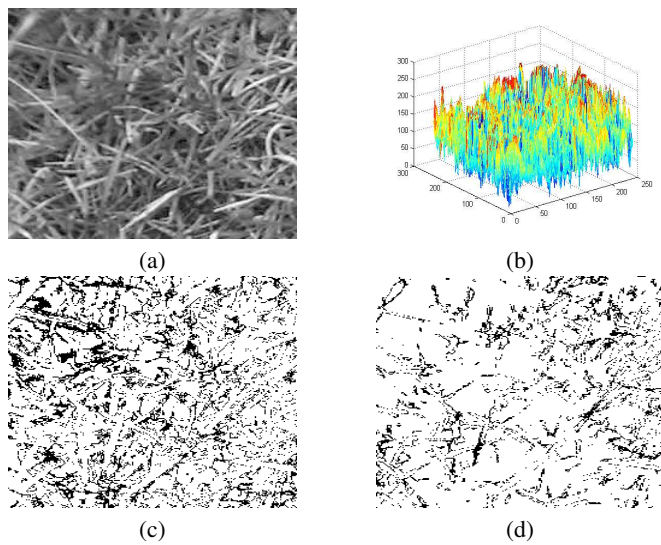
(a)   (b)

(c)   (d)

Fig. 2 (a) Original texture image. (b) 3D surface visualization of the image with $D = 2.79$. (c) Black-and-white image obtained by thresholding intensity values between 100 and 120, its fractal dimension $D = 1.67$. (d) Black-and-white image obtained by thresholding intensity values between 80 and 100, its fractal dimension $D = 1.49$.

The question arises then whether natural textures are objects with fractional dimension information. Let's check a natural texture first. A typical grass texture image is shown in Fig. 2 (a). If we represent the image as a 3D-surface in Fig. 2(b) by taking the image intensity as the height, it is seen that this surface is highly "irregular"; its box fractal dimension in 3D amounts to 2.79. (The box fractal dimension in 3D is found by covering the space with a δ-mesh grid of cubes and evaluating the exponential changing ratio of the number of cubes intersecting the surface with respect to the resolution δ.)

So, the answer is quite positive: textures in nature do encode fractal dimension information which reflects the irregularity of textures. However, the fractal dimension alone, as defined above, does not provide a rich description. It is just one number. Physicists and Applied Mathematicians have developed multifractal analysis as an extension to classical fractal analysis for more powerful descriptions. The concept of the fractal dimension is extended to the concept of the multifractal spectrum (MFS) as follows: First define a point categorization on the object function according to some criteria. Then the fractal dimensions are calculated for every point set from this categorization. The resulted MFS vector gives a rich description of the inherent texture structure.

To give a demonstration, we take a simplistic categorization of image points based on the intensity values of the pixels. The image pixels are categorized by their intensities. Then for every pixel set, we obtain a black-and-white image by assigning 0 to pixels in this set and 1 to all other pixels. The collection of the fractal dimensions of all the black-and-white images is called an MFS vector. Two

examples of such obtained images are shown in Fig. 2 (c) and Fig. 2 (d). Clearly the distributions of points in these two images are very "irregular", and they are more like fractal objects with fractional fractal dimensions.

Obviously, different criteria for defining the point categorization lead to different fractal spectra. The simplistic approach demonstrated above has some drawbacks in practice. A categorization based on image intensity is not robust to illumination changes. Therefore, we will introduce another categorization method which is based on the idea of the "density function". The density function, which is a local descriptor, can be defined on different functions of the image intensity (for example output of various edge filters). Intuitively, the density function of a quantity defines the change of that quantity over scale, and like the fractal dimension, which in contrast is defined globally, models this change by the power law behavior.

3 MFS for Textures

The MFS is the vector of the fractal dimensions of some categorization of the image. We follow the classic approach in the MFS literature of introducing the local density function in the categorization. Other terms for the density, that have been used in the literature, are Hölder exponent (Falconer 1997) or local fractal dimension (Varma and Garg 2007).

3.1 The density function

Let μ be a measure function on R^2 (A description of the different μ used in our texture description will be given in Section 3.3. one example for μ is the integral of pixel-wise intensities in an area). For $\boldsymbol{x} \in R^2$, denote $B(\boldsymbol{x}, r)$ as the area with center $\boldsymbol{x}$ and radius $r > 0$ (a closed disk of radius r or a square of length r). We describe $\mu(\boldsymbol{x}, r)$ at a point $\boldsymbol{x}$ as an exponential function of r, i.e. $\mu(\boldsymbol{x}, r) = kr^{d(\boldsymbol{x})}(\boldsymbol{x})$, with $d(\boldsymbol{x})$ the density function and k some constant. Thus, $\log(\mu(\boldsymbol{x}, r)) = \log k + d(\boldsymbol{x}) \log r$. Then the *local density function* of $\boldsymbol{x}$ is defined as

$$d(\boldsymbol{x}) = \lim_{r \to 0} \frac{\log \mu(B(\boldsymbol{x}, r))}{\log r}. \tag{2}$$

The density function describes how locally the measurement μ satisfies the power law behavior. It somehow measures the "non-uniformness" of the intensity distribution in the region neighboring the measured point. Fig. 3 illustrates the concept. An intensity patch whose intensities are decreasing from the center to the boundary (i.e. the intensity surface is concave down) has a density < 2 (Fig.3a and b). For a uniform intensity patch the density $= 2$ (Fig. 3c). Finally, if the surrounding intensity increases from the center to the boundary (i.e. the intensity surface is concave up), the density > 2. Figure 4 demonstrates the estimated density function for the grass texture.
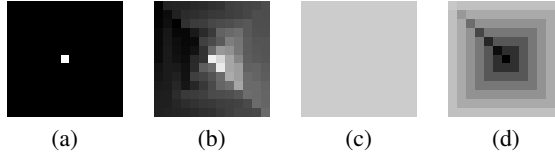
Fig. 3 Local density of intensity at the center point (a) 0.18; (b) 1.5; (c) 2; (d) 2.5. 7 sizes of r ranging from 1 to 13 pixels have been used, and the intensities are defined on the range [1..256] with black being 1 and white being 256. The measurement $\mu(B(x, r))$ is the sum of the intensity pixels in the square of $r \times r$ pixels. The density, $d(\boldsymbol{x})$, is obtained as the slope of the line fitted to the data $\{\log r, \log \mu(B(\boldsymbol{x}, r))\}$.
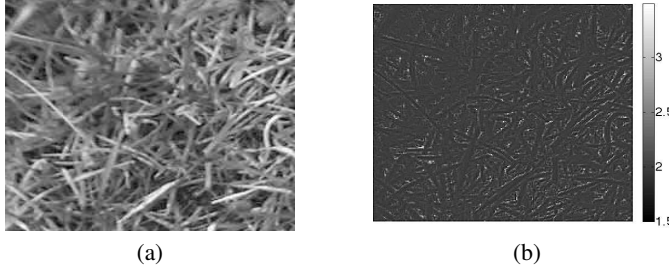

Fig. 4 (a) Original grass texture image as in Fig 2. (b) Corresponding density measurements estimated from six levels of resolution. The density measurements range from 1.5 to 3.4.

3.2 Definition of the MFS

For any $\alpha \in R$, define

$$E_\alpha = \{\boldsymbol{x} \in R^2 : d(\boldsymbol{x}) = \lim_{r \to 0} \frac{\log \mu(B(\boldsymbol{x}, r))}{\log r} = \alpha\}. \tag{3}$$

That is, E_α is the set of all image points $\boldsymbol{x}$ with local density α. Usually this set is irregular and has a fractional fractal dimension $dim(E_\alpha)$. Thus we obtain a point categorization $\{E_\alpha : \alpha \in R\}$ of the image with an MFS denoted as

$$f(\alpha) = \{dim(E_\alpha) : \alpha \in R\}, \tag{4}$$

where $dim(\cdot)$ is defined as in eq. (1). The MFS $f(\alpha)$ defined by (4) is the natural extension of the concept of fractal dimension. It is a vector of fractal dimensions.

3.3 Combination of multiple MFS vectors

Intentionally, we haven't given the definition of the measurement function μ yet. The first approach is to work directly on the intensity domain. Denoting with $I(\boldsymbol{x})$ the intensity of pixels $\boldsymbol{x}$, we define $\mu(B(\boldsymbol{x}, r))$ as:

$$\mu(B(\boldsymbol{x}, r)) = \int \int_{B(\boldsymbol{x}, r)} (G_r * I) d\boldsymbol{x}, \tag{5}$$
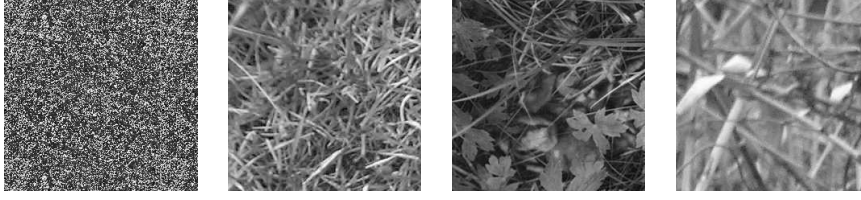
Fig. 5 Four textures: carpet, grass, leaf, tree from left to right.

where $''*''$ is the 2D convolution operator and G_r is a Gaussian smoothing kernel with variance r:

$$G_r(\boldsymbol{x}) = \frac{1}{r\sigma\sqrt{2\pi}} e^{\frac{-\|\boldsymbol{x}\|_2^2}{2\sigma^2 r^2}},$$

for some pre-defined parameter σ. In other words, $\mu(B(\boldsymbol{x}, r))$ is the sum of the average intensity values inside the disk $B(\boldsymbol{x}, r)$. This results in the definition of the density of the intensity function, which describes how the intensity at a point changes over scale (as we change the size of the neighborhood). The corresponding MFS vector encodes the fractal dimension for multiple values of the density of the intensity.

In practice the measurement function $\mu(B(\boldsymbol{x}, r))$ is still not very robust to large illumination changes due to limitations on the available image resolution. But we can define the density function on other quantities. One can imagine many meaningful definitions for $\mu(B(\boldsymbol{x}, r))$ such that the corresponding MFS is less effected by illumination changes. One choice is to define $\mu(B(\boldsymbol{x}, r))$ on the energy of the gradients. Consider $\{f_k, k = 1, 2, 3, 4\}$ to be four directional differential operators along the vertical, horizontal, diagonal and anti-diagonal directions. Then we define the measurement function $\mu(B(\boldsymbol{x}, r))$ for the image texture I as:

$$\mu(B(\boldsymbol{x}, r)) = (\int\int_{B(\boldsymbol{x}, r)} \sum_k (f_k * (G_r * I))^2)^{\frac{1}{2}} d\boldsymbol{x}. \tag{6}$$

Another meaningful choice for $\mu(B(\boldsymbol{x}, r)$ is the sum of the Laplacians of the image inside $B(\boldsymbol{x}, r)$, i.e

$$\begin{aligned}
\mu(B(\boldsymbol{x}, r)) &= \int\int_{B(\boldsymbol{x}, r)} |\nabla^2(G_r * I)| dx \\
&= \int\int_{B(\boldsymbol{x}, r)} |\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)(G_r * I)| d\boldsymbol{x}.
\end{aligned} \tag{7}$$

Different definitions of $\mu(B(\boldsymbol{x}, r))$ lead to different MFS vectors which capture different aspects of the structure of textures. Multiple MFS vectors could and should be combined to better describe the texture.

Fig. 7 (a) shows the MFS (based on (5)) for the four textures in Fig. 5. It demonstrates that the MFS vectors of the different textures are significantly different. Fig. 7 (b) shows the three MFS vectors based on the three measurement functions μ described above.
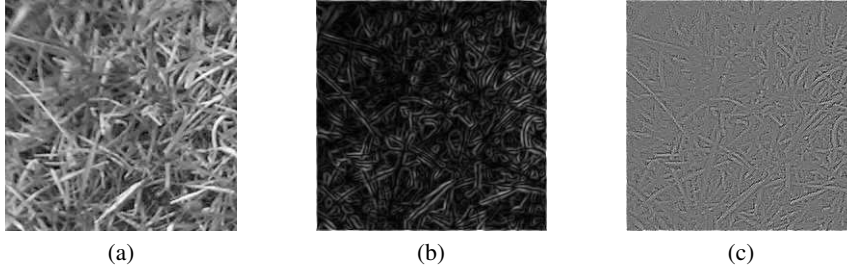
(a)   (b)   (c)

Fig. 6 (a) Intensity image of the grass texture. (b) Energy of the first order derivatives as defined in (6). (c) The Laplacian of the image.

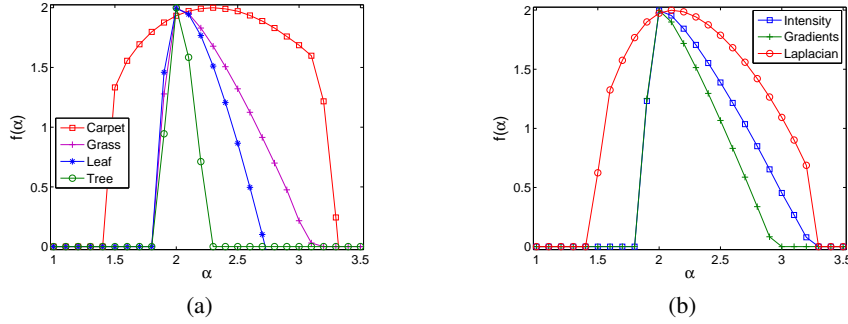(a)   (b)

Fig. 7 (a) The MFS of the density of intensity for the four textures shown in Fig. 5. (b) The MFS for the three measurement functions applied to the grass texture as shown in Fig. 6.

3.4 Computation of the MFS

The experiments in this paper are based on a direct implementation of the MFS. We compute the density $\alpha(\boldsymbol{x})$ for each pixel $\boldsymbol{x}$ of the image by linear fitting the slope of the line in a scaling plot of $\log \mu(B(\boldsymbol{x}, r))$ against $\log r$. Then we take a discrete sample set $\{\alpha_i, i = 1, \cdots, N\}$ from the interval $[1, 3.5]$ and find for each α_i the corresponding point set E_{α_i} by collecting all pixels whose densities are close to α_i under some threshold. The fractal dimension $f(\alpha_i)$ for each point set E_{α_i} is computed by fitting the slope of the line in the scaling plot of $\log N(\delta, E_{\alpha_i})$ vs $\log \frac{1}{\delta}$.

It is worth noting that the multifractal spectrum generally (in applications in other fields) is not computed by this direct approach. Instead, the so-called moment-based algorithm (see Appendix A) is widely adopted. All demonstrations in Sections 3 and 4 have been computed with the moment-based algorithm. Its advantage is its efficiency and robustness and that it does not require the explicit computation of the density. However, the MFS computed by this approach is not optimal for the application of texture classification. One reason is there doesn't exist a reliable approach to measure how well the fractal dimension of each density

value is estimated; the other reason is that there is correlation among the elements in the MFS computed by this approach. On the other hand, using the direct method could lead to some "bad" density values. However, these bad density values can be thrown away automatically in the learning during the classification. For this reason we use the direct approach in our experiments on texture classification. We should mention that the fractal geometry literature keeps exploring better techniques for computing the MFS, which also provide confidence intervals (Wendt et al. 2007).

4 Invariance of the MFS to various transforms

4.1 Spatial invariance

A very attractive property of the MFS is its invariance under the so-called *bi-Lipschitz* transform. A function $g : R^2 \to R^2$ is called a bi-Lipschitz transform, if there exist two constants $c_1 > 0$, $c_2 > 0$ such that for any $\boldsymbol{x}, \boldsymbol{y} \in R^2$,

$$c_1 \|\boldsymbol{x} - \boldsymbol{y}\| < \|g(\boldsymbol{x}) - g(\boldsymbol{y})\| < c_2 \|\boldsymbol{x} - \boldsymbol{y}\|, \tag{8}$$

where $\|\boldsymbol{x} - \boldsymbol{y}\|$ is the Euclidean metric between points $\boldsymbol{x}$ and $\boldsymbol{y}$. Basically, any smooth transform is a bi-Lipschitz transform. It is a very general mapping which includes translation, rotation, projective transformation, and texture warping of regular surfaces. It also includes changes in viewpoint and non-rigid deformations. Then we have the following theorem.

Theorem 1. *The MFS based on the density function as defined in (5, 6, 7) is invariant under any spatial bi-Lipschitz transform.*

The proof, which is given in Appendix B, consists of two steps. The first step shows that the level sets are preserved under a geometric transform. The second step shows that every level set in the original image and its corresponding level set in the transformed image have the same fractal dimension.

   We need to emphasize here that the choice of the function used to define level sets E_α in the MFS (here we used the density function) is actually very flexible. As long as this function is invariant to the local affine transform, the corresponding MFS derived from (4) is invariant to the spatial bi-Lipschitz transform. This can be shown easily by replacing the density function by a locally affine transform in step 1 of the proof. Thus, some common measurements on image pixels (such as Intensity; Magnitude of gradients; Magnitude of Laplacian, etc) as well as other filtered outputs of the image, are all fine choices for a spatial bi-Lipschitz invariant MFS. The reason why we adopt the density function here is for better robustness to illumination changes.

   Let us stress that the invariance of the MFS to bi-Lipschitz maps has been proven for images of infinite resolution. In practice we are dealing with images of finite resolution. Nevertheless, we found that the MFS is very robust to perspective transformations and a warping of the surface, even for low resolution images. Usually four to five levels of resolution (defined by radius r) are sufficient for a good estimation of the fractal dimensions.
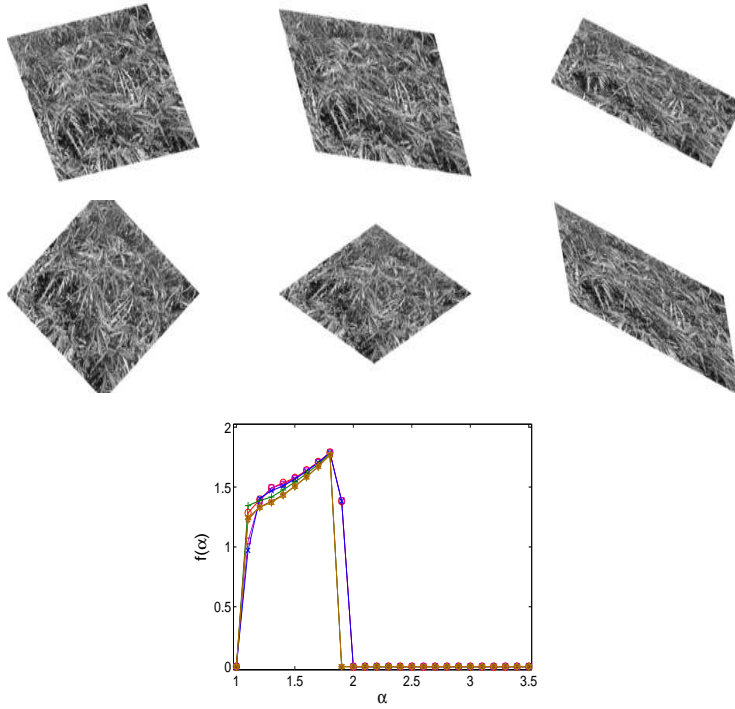
Fig. 8 Six perspective views of a *foliage* texture and the corresponding MFS vectors of the density of intensity.

Two examples are given: Fig. 8 shows the same foliage texture on a plane seen from different viewpoints. Fig. 9 shows six images of a tree texture on general smooth surfaces under perspective projections. It can be seen that in both cases the corresponding MFS vectors are nearly the same.

4.2 Illumination invariance

Another important factor to be considered is illumination. We also desire invariance to illumination changes. Consider a piecewise local linear transform on the illumination. In other words, there is a partitioning of the spatial domain $\{D_1, D_2, \cdots, D_N\}$, such that the intensity $I(\boldsymbol{x})$ transforms to

$$a_k I(\boldsymbol{x}) + b_k, \quad x \in D_k, \quad k = 1, \ldots, N,$$

where the a_k are the local multiplicative changes, and the b_k are the local additive changes in the intensity.

Theorem 2. *The MFS vector derived from (5) is invariant to local multiplicative changes in the image illumination. The MFS vectors derived from (6) and (7) are invariant to local linear changes in the image illumination.*
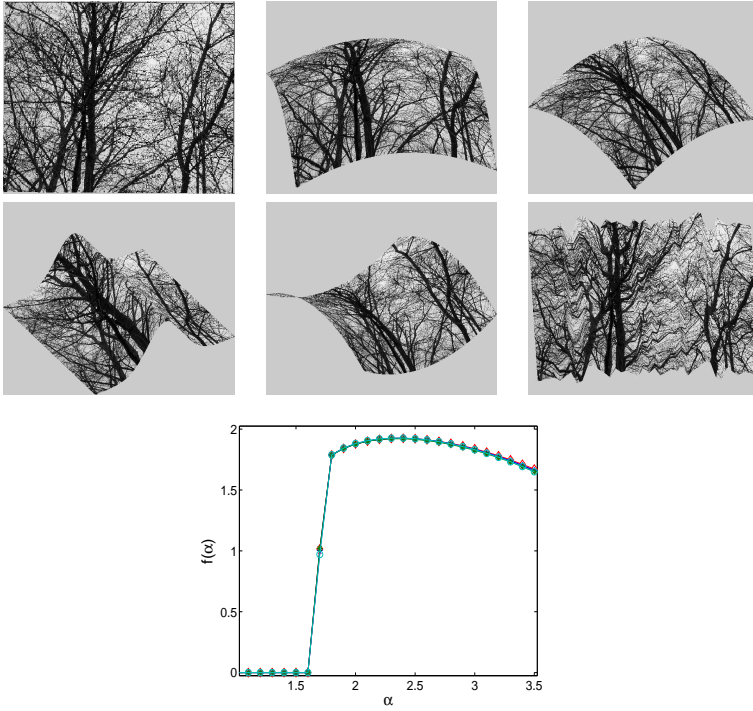
Fig. 9 Perspective images of tree texture on different general smooth surfaces and the corresponding MFS vectors of the density of intensity.

It is not difficult to justify Theorem 2. All we need to show is that the level sets defined by the density functions are preserved under the illumination transformation (corresponding to step 1 in Theorem 1). The density function (defined in eq. 2) estimates the ratio of the measurement μ with respect to the resolution (defined by r) under log coordinates. A local multiplicative change in the image intensity is transformed into a local additive change of the measurement μ under log coordinates, and additive changes don't influence the ratio. This proves that the level sets and thus the MFS are invariant to local multiplicative changes. Furthermore, the measurements defined from (6) and (7) are based on the image derivatives, and thus they are not affected by local additive changes in the intensity.

Fig. 10 shows two sets of nine real texture images taken by a family digital camera. The three images in each row depict the same scene from different viewpoints under different illumination conditions. The corresponding MFSs are shown in Fig. 11. As can be seen the MFSs of different textures are significantly different, while the MFSs of the same texture under different view-points are almost identical. Fig. 13 shows eight views of a cloth texture on the moving arm of a person. The illumination is not controlled. The corresponding MFSs again demonstrate the robustness under geometric deformations and changes in illumination.
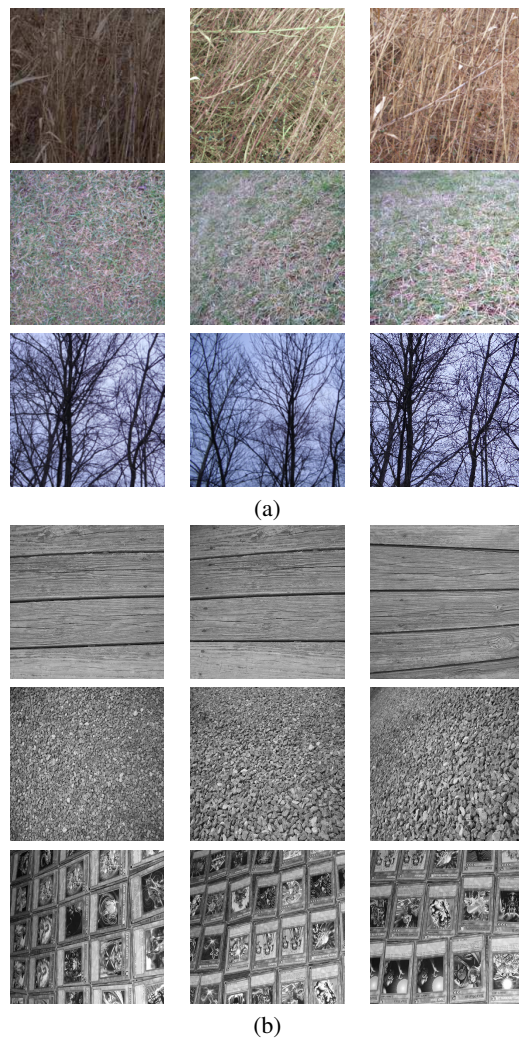
(a)

(b)

Fig. 10 Real 3D texture images under different lighting condition and view points. (a): First row: bulrushes B1, B2, B3. Second row: grasses G1, G2, G3. Third row: trees T1, T2, T3. (b): First row: blanket wood W1, W2, W3. Second Row: gravel S1, S2, S3. Third row: Yugioh cards M1,M2, M3.

4.3 The MFS in comparison to the histogram

The most popular global statistical texture descriptor is the histogram. It could be interpreted as a descriptor defined on some categorization of the image, such as a categorization of pixels based on intensity, or a categorization of feature points based on their neighboring texture information. The descriptor for each category is simply counting the number of points in the category. The drawback of such a
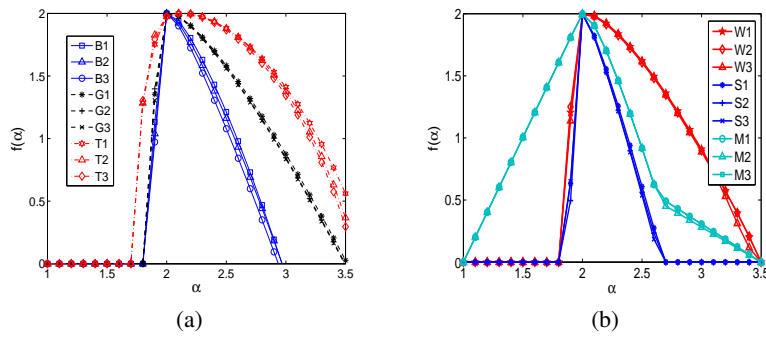
Fig. 11 The MFS of the density of intensity for the texture images in Fig. 10 (a) and (b) respectively.
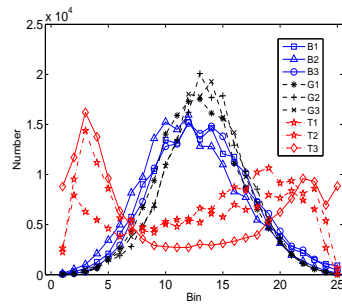

Fig. 12 The histograms of intensity measurements for the nine textures in Fig. 10 (a). The images were linearly normalized for intensity and the number of bins is the same as in the corresponding MFSs in Fig.11.

description is that the information of how pixels are spatially distributed is lost. This spatial information is not easy to enforce without losing invariance to spatial transforms. Here lies the power of fractal geometry. One could look at the MFS as adding a multi-resolution analysis layer to the histogram. Whereas the histogram only counts the number of points in a category under one resolution, the MFS counts the number of points in a category under multiple resolutions and estimates the exponential changing ratio of the number of points with respect to the resolution r. This multi-resolution analysis encodes spatial information in form of the geometrical distribution of the point sets.

The traditional intensity-based histogram is quite robust to small spatial changes, but not invariant to perspective transformations, and it is very sensitive to changes in illumination. Defining the histogram instead on textons (Leung and Malik 2001; Cula and Dana 2001; Varma and Zisserman 2002) reduces the negative effects of changes in illumination, but in trade-off to larger sensitivity to global geometric transforms. In contrast, the MFS can also deal with illumination by incorporating various low-level local feature measurements without sacrificing global spatial in-
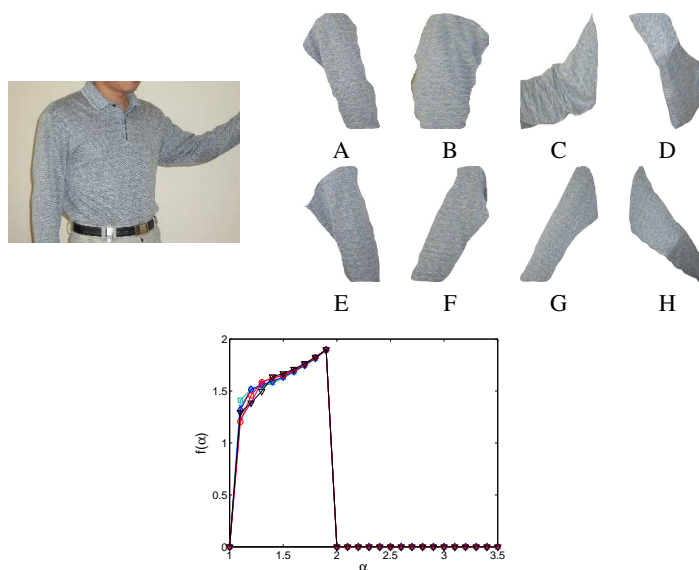
Fig. 13 The arm part extracted from a human gesture video under different illumination conditions.

variance, thus providing greater descriptiveness and more robustness on illumination changes. It surpasses the histogram in the sense that it captures more geometrical information of the texture and is invariant to global perspective transformations. Fig. 12 shows the histograms for the 9 textures in Fig. 10 (a), demonstrating better robustness of the MFS to geometric transformations and illumination changes.

5 Experimental Evaluation

We evaluated the performance of the MFS descriptor on classic texture *retrieval* and *classification* and compared it to the three methods in (Lazebnik et al. 2005; Varma and Zisserman 2003; Varma and Garg 2007) summarized below.

Retrieval and classification are implemented as described in (Lazebnik et al. 2005) using the procedure of (Liu and Picard 1996). Briefly, for retrieval, given a query image we selected other images from our database in an increasing order of distance, i.e., from the most similar to the least similar. Each image in the database is used as a query image once, and the performance is summarized as a plot of average recall vs. the number of retrievals. Average recall is defined as the ratio of the number of images retrieved from the same class as the query image over the number of images in that class averaged over all queries. For example, perfect performance for a given class would correspond to an average recall of 100 % after n-1 retrievals, where n is the number of images in that class. For classification we used the nearest-neighbor technique. 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 reported classification rate is the average over 200 random subsets.

5.1 Other methods

The first method chosen for comparison is the sophisticated interest-point based texture representation of Lazebnik et al. (Lazebnik et al. 2005), which we call the LSP-method. The basic idea of the LSP-method is to characterize the texture by clusters of elliptic regions. The elliptic regions are localized with the Harris (H) and Laplacian (L) operators. The ellipses are transformed to circles. Thus this descriptor is locally invariant to affine transforms. Each region is represented by two descriptors: one encoding the histogram of the intensity distribution in a neighborhood (S) (an intensity-based modification of the spin images (Johnson and Hebert 1999); the other encoding the histogram of edges in a neighborhood (R) (a variation of SIFT features). For each image, the features are clustered to obtain a signature, that is a histogram of clusters with their center vectors. Signatures are compared using the earth mover's distance.

The LSP-method essentially represents an image texture by its frequency of texture elements. It is robust to many geometric transforms, because of the affine normalization of the regions of features. However, it is affected by large scale changes and large changes in view-point, as those cause changes in the description of features. The descriptor involves a very large number of parameters, a few thousands for a typical image. The reason is that every image is represented by a histogram of 40 clusters, each represented by a 100 dimensional vector. Furthermore, the LSP-method requires sophisticated preprocessing to compute the texture elements and K-mean clustering. Good performance of these two preprocessing steps are critical for the success of the LSP-method. In comparison, the MFS is simple and straightforward to compute as it neither relies on feature detection nor on clustering techniques.

The second method is the VZ-Joint method by Varma and Zisserman (Varma and Zisserman 2003), which uses a dense set of textons (elements of texture). The textons in this case encode the joint distribution of all pixels in a fixed sized neighborhood, and the descriptor is a one-dimensional histogram of these textons. This simple descriptor is non-invariant, but has been shown to outperform sophisticated texton descriptors based on the output of complex filters.

The third method (Varma and Garg 2007), which we call the VG-Fractal method uses statistics of local density functions and extensions. Images are prefiltered using the MR8 filter bank (Varma and Zisserman 2002), which consists of a Gaussian, a Laplacian, and the maximum edge and bar response at 3 scales. At every point, the local density (local fractal dimension), that is the slope of the line fitted to the data $\{\mu(\boldsymbol{x}, r), r\}$, as well as the intercept of the line are computed. From these filtered outputs, totally 13 values are chosen to form the feature vector. The feature vectors of a subset of the images in each class are clustered for the classification. The cluster centers are the textons. The descriptor for each image is a normalized histogram of the pixel texton labelings, and the distance between descriptors is measured using χ^2 statistics.
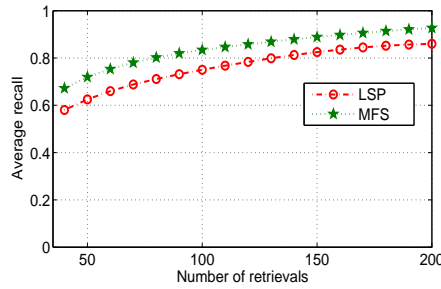
Fig. 14 Retrieval curves for the UIUC dataset by the MFS and the LSP methods in (Lazebnik et al. 2005).

In comparison to our method, this method uses the concept of fractal geometry only locally, namely for the description of feature vectors, while our method uses the the standard global concept of the fractal dimension. The method in (Varma and Garg 2007), thus, is not globally invariant. It is built on the same concept of the density function in (Xu et al. 2006) but uses more measurements on the behavior of the power law, which leads to much higher dimensional local feature vectors. Densely using these higher dimensional features gives the method good discriminative power in classification, which results in a performance similar to our method. As (Varma and Garg 2007) pointed out, their method is much more computationally expensive in space and time, mostly because of the K-mean clustering of 13 dimensional vectors at every image pixel.

5.2 Dataset from (Lazebnik et al. 2005)

The first experiment used the dataset provided in (Lazebnik et al. 2005), which consists of 1000 uncalibrated, unregistered images of size 640×480: 25 different textures each with 40 samples (available at (UIUC dataset)).

The results for the methods (Lazebnik et al. 2005) and (Varma and Zisserman 2002) are from the paper (Lazebnik et al. 2005), and the results for Varma and Garg's method are from their paper (Varma and Garg 2007). There are two components in (Lazebnik et al. 2005)'s implementation: region detector and texture descriptor. (Lazebnik et al. 2005) provides the performance comparison for multiple combinations. Here we only use the best one: (H+L)(S+R), which is a combination of both region detectors and both texture descriptors (For details see (Lazebnik et al. 2005)).

The images were intensity normalized, such that all images in a class have the same mean. In our method, we used a combination of three MFS vectors: an MFS defined on the density of intensity as defined by the measurement function (5), an MFS defined on the energy of gradients, and an MFS defined on the Laplacian. (We found that the density of the gradients and the Laplacian (as defined in (6) and (7)) was less discriminative than the measurements themselves. One reason is

that all images were intensity normalized, and thus the advantage of illumination invariance due to the density function was outweighed.) Totally 8 different radius values r ranging from 1 to 8 pixels were used in the estimation of the density as well as the estimation of the fractal dimension of the density of intensity. The density values were in the interval $[1, 3.5]$. Only 3 radius values ranging from 1 to 3 pixels were used in the estimation of the fractal dimension of the two other MFS vectors. The reason is that the patterns of these measurements of image derivatives are of less variability than the patterns of the density of image intensity, and thus large radius values would lead to too few level sets.

The MFS vector is computed as follows. First, for each measurement function, a 26 dimensional MFS vector of uniformly spaced values was computed respectively. Then a subset of the features was selected using the procedure in Chen and Lin (2006), which is based on a combination of the Fisher score method and SVM cross-validation. In our implementation, using the data-set, for each of the 26 features we computed the Fisher score and ranked the features according to their values. Then we found the number of features to be selected using SVM cross validation. That is, for differently sized feature vectors we performed cross-validation to find the number of features with lowest validation error. In this experiment, we selected 13 values for the density of intensity, 10 values for the energy of gradients and 10 values for the Laplacian, resulting in a 33-dimensional combined MFS vector. When combining the three individual MFS vectors, a weighting of: $\frac{1}{5}(1, 2, 2)$ was used. The distance between the combined MFS vectors was evaluated using the L_1 norm. For the classification we used the nearest neighbor algorithm.

Fig. 14 compares the MFS to the technique in (Lazebnik et al. 2005) for retrieval, showing that the performance of the MFS method is better by a noticeable margin. It is worth noting here that the number of parameters involved in the MFS is significantly smaller than in the method in (Lazebnik et al. 2005) (seventy-eight vs thousands).

Fig. 15 evaluates classification performance for our method and the three methods described above, and shows the influences of the number of samples on the classification rate. Note that the results for the other methods are taken from the respective papers, and for the VG-Fractal method (Varma and Garg 2007) data was available only for the mean and only for training samples, whose number is a multiple of five. The figure demonstrates that the MFS has an average classification rate, which is generally much better than the VZ-Joint method and comparable to the LSP and VG-Fractal methods. Our method is slightly better than the two other methods for a smaller number of training samples, and the LSP method slightly outperforms the other two methods for a larger number of training samples. In more details, the MFS performs worse than the LSP-method only in four of the 25 classes, all of which have significant illumination changes. For all other classes, it actually has a very impressive classification rate (see Fig. 15). This indicates that our MFS approach is less robust than the LSP-method for large illumination changes.
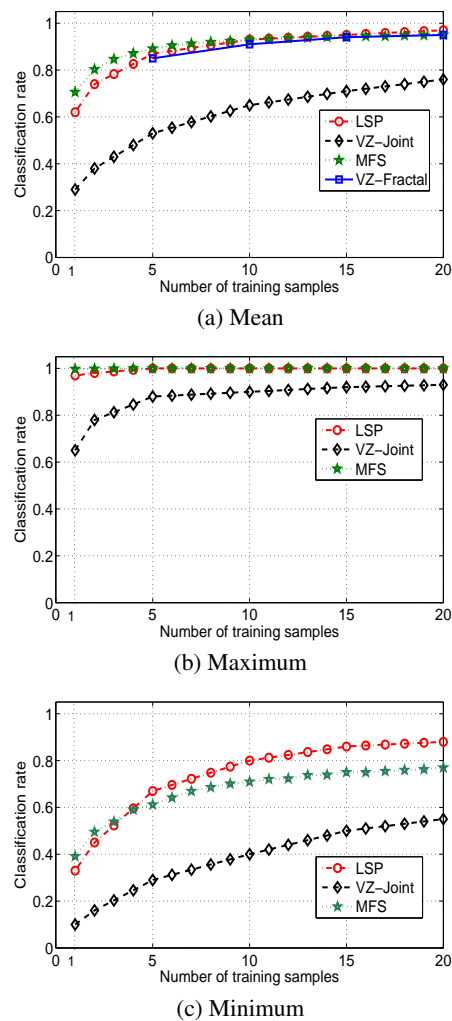
(a) Mean

(b) Maximum

(c) Minimum

Fig. 15 Classification rate vs. number of training samples per texture class for the UIUC dataset. Four methods are compared: the MFS method, the LSP method in (Lazebnik et al. 2005), the VZ-Joint method in (Varma and Zisserman 2002) and the VG-Fractal method in (Varma and Garg 2007). (a) Mean classification rate for all 25 classes; (b) Classification rate of the best class; (c) Classification rate of the worst class.

5.3 Dataset of high-resolution texture

Recall that the invariance of our MFS to local linear illumination changes has been obtained for the continuous case, assuming infinitely small r. For images of low resolution the r available to us is not small enough, and the descriptor changes
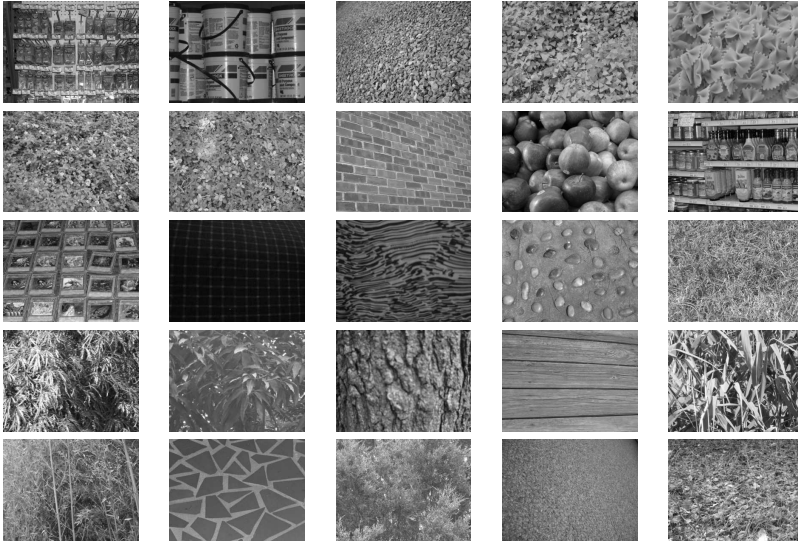
Fig. 16 UMD dataset. One sample each of the 25 texture classes used in the experiments of Section 5.2.

with illumination. To demonstrate this point, the next experiment uses images of high-resolution.

Furthermore, we would like to address that a major advantage of the MFS is its low dimension and simple implementation. In comparison, the feature-based method (Lazebnik et al. 2005) heavily depends on region detection. As a result, the clustering in later stages could cause troubles because of too many detected regions in high resolution images.

For this experiment, we collected a high resolution dataset (available at (UMD dataset)). All images were taken by a family camera. The dataset consists of 1000 un-calibrated, unregistered images of size 1280×960: 25 different textures each with 40 samples. Within each class, significant viewpoint changes and scale differences are present, and the illumination conditions are uncontrolled. The textures in this dataset are non-traditional, including images of fruits, shelves of bottles and buckets, various plants, and floor textures (See Fig. 16).

We compared our method against the LSP method and the VG-Fractal method. The implementation for the MFS is the same as in the previous experiment. We used the same three MFS vectors. However, the weighting scheme to combine the three MFS vectors was $\frac{1}{5}(3, 1, 1)$, in order to emphasize the measurements from density of intensity. The reason is that the density of intensity is more discriminative than the Gradient and Laplacian measurements and for higher image resolution its estimation is more stable.

In the implementation of the LSP method and the VG-Fractal method we made the following choices: The algorithm in (Lazebnik et al. 2005) usually detected a large amount of regions for many images. Thus, for each image we chose 5000
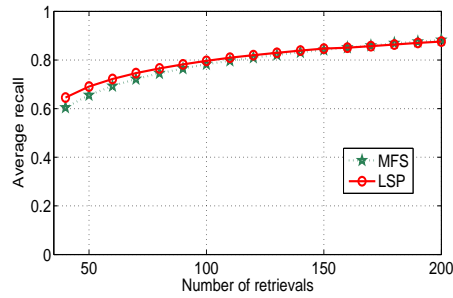
Fig. 17 Retrieval curves for the MFS and the LSP methods in (Lazebnik et al. 2005) using our high-resolution dataset.

regions out of the set, which are then clustered using K-means into 40 clusters. The algorithm in (Varma and Garg 2007) calls for clustering 13 dimensional vectors. To deal with the overwhelming computation when processing images of high resolution, we randomly sub-sampled one fourth of all image points and used only these feature vectors in the clustering.

Fig. 17 shows a comparison of our method to the technique in (Lazebnik et al. 2005) for texture retrieval. As can be seen, the MFS performed better than the LSP method. Fig. 18 evaluates classification performance using the same strategy as in Section 5.2. The figure indicates that the MFS method performed better than the LSP method by about 10% for the differently sized training sets, and slightly better than the VG-Fractal method. Our MFS texture descriptor as well as the VG-Fractal method benefit from the extra details provided by the high resolution of the images, while the LSP method hardly gains any benefit. It is noted that the LSP method performed worst on the plant textures because of their highly fractal structure. These are also the textures which the MFS captures very well.

6 Summary and Conclusions

We showed both in theory and experiment that the MFS offers a new and promising efficient texture description. Not only does it compete in performance with other top methods in classifying textures, but it also has a mathematically justified global bi-Lipschitz spatial invariance and local linear invariance to illumination. No other approach proposed in the past has such global properties. In practice, the MFS is very robust in handling spatial changes, and it has satisfactory performance in handling illumination changes. Experiments on a standard dataset demonstrated its effectiveness and efficiency in texture retrieval and classification tasks, achieving a performance comparable to the top feature based methods, but with far lower dimension. Furthermore, the experiments on a high-resolution texture database clearly showed that the MFS is capable of utilizing the extra details provided by images of high resolution to improve its performance. It is worth noting that the MFS is very efficient and simple to compute without requiring feature detection
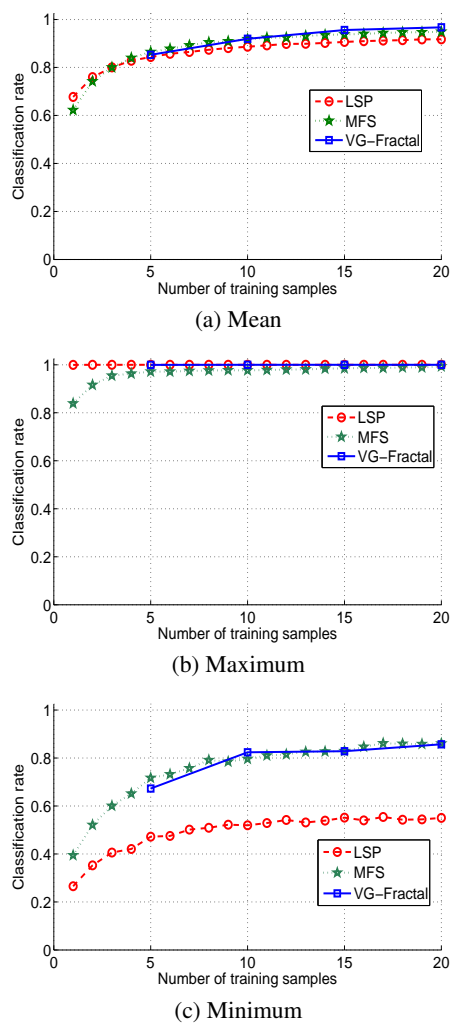
(a) Mean

(b) Maximum

(c) Minimum

Fig. 18 Classification rate vs. number of training samples for the high-resolution dataset. Three classification rates are plotted: (a) Mean classification rate of all 25 classes; (b) Classification rate of the best class; (c) Classification rate of the worst class. Three methods are compared: LSP-method, MFS method and VG-Fractal method.

and clustering. Thus a large image size doesn't impose much computational burden on the computation of the MFS descriptor.

In future work we will be investigating whether the MFS can be applied to other more discriminative measurements. Can we take local invariant feature descriptors, such as SIFT features (Lowe 2004), spin images (Johnson and Hebert 1999), or MSR features (Matas et al. 2002) and define an MFS on them to obtain

a global spatial descriptor? If this is the case, we will have a very useful tool for many recognition tasks in vision.

References

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

R. Azencott, J. Ping, and L. Younes, "Texture classification using windowed Fourier filters," *IEEE Trans. on Pattern Analysis and Machine Intelligence.*, Vol. 19(2), pp. 148-153, 1997.

R. Bajcsy and L. Lieberman, "Texture gradient as a depth cue", *Computer Graphics and Image Processing*, 5(1): pp. 52-67, 1976.

M. Chantler, M. Petrou, A. Penirsche, M. Schmid, G. McGunnigle, "Classifying surface texture while simultaneously estimating illumination direction", *Int'l Journal of Computer Vision*, Special Issue on Texture Analysis and Synthesis, Vol. 62 (1-2), pp. 83-96, 2005

B. B. Chaudhuri and N. Sarkar, "Texture segmentation using fractal dimension", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 17(1), pp. 7277, 1995.

Y.-W. Chen and C.-J. Lin, "Combining SVMs with various feature selection strategies." in *Feature extraction, foundations and applications.* Springer, 2006.

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

O.G. Cula and K.G. Dana, "Compact representation of bidirectional texture function", *CVPR*, Vol. I, pp. 1041-1047, 2001.

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

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

F. Espinal, B. D. Jawerth and T. Kubota, "Wavelet-based fractal signature analysis for automatic target recognition," *Opt. Eng.*, Vol. 37(1), pp. 166-174, 1998.

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

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

J. Garding and T. Lindeberg, "Direct computation of shape cues using scale-adapted spatial derivative operators,"*Int'l Journal of Computer Vision*, Vol. 17 (2), pp. 163-191, 1996.

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

A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(5), pp. 433-449, 1999.

L. Kam and J. Blanc-Talon, "Are multifractal multipermuted multinomial measures good enough for unsupervised image segmentation", *CVPR*, pp. 58-63, 2000.

L. M. Kaplan, "Extended fractal analysis for texture classification and segmentation", *IEEE Trans. on Image Processing*, Vol. 8(11), pp. 1572-1585, 1999.

L. M. Kaplan and C. C. J. Kuo, "Texture roughness analysis and synthesis via extended self similar (ESS) model", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 17(11), pp. 1043-1056, 1995.

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.

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

A. Laine and J. Fan, "Texture classification by wavelet packet signatures," *IEEE Trans. on Pattern Analysis and Machine Intelligence.*, Vol. 15(11), pp. 1186-1191, 1993.

S. Lazebnik, C. Schmid and J. Ponce, "A sparse texture representation using affine-invariant regions,"*IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 8(27), pp. 1265-1278, 2005.

T. Leung and J. Malik, " Representing and recognizing the visual appearance of materials using three-dimensional textons", *Int'l Journal of Computer Vision*, Vol. 43(1), pp. 29-44, 2001.

F.Liu and R.W. Picard, "Periodicity, directionality, and randomness: World features for image modeling and retrieval", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol 18(7), pp. 722-733, 1996.

X. Liu, Y. Yu, and H. Shum, "Synthesizing bidirectional texture functions for real-world surfaces", *Proc. ACM SIGGRAPH*, pp. 117-126, 2001.

D. Lowe, "Distinctive image features from scale-invariant keypoint", *Int'l Journal of Computer Vision*, 60(2), pp. 91-110, 2004.

J. Malik and R. Rosenholtz, "Computing local surface orientation and shape from texture for curved surfaces", *Int'l Journal of Computer Vision*, Vol. 23(2), pp. 149-168, 1997.

S.G. Mallat, "A theory for multi-resolution signal decomposition: the wavelet representation", *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 11, pp. 674-693, 1989.

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

J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions", *Proc. BMVC*, 2002.

T. Mihran, "Moment based texture segmentation," *Pattern Recognition Letters*, Vol. 15, pp. 659-668, 1994.

F. Mindru, T. Tuytelaars, L. Van Gool and T. Moons, "Moment invariants for recognition under changing viewpoint and illumination", *Computer Vision and Image Understanding*, Vol. 94(1-3), pp. 3-27, 2004.

J. L. Mundy and A. Zisserman, *Geometric Invariance in Computer Vision*, MIT Press, 1992.

A. Neubeck, A. Zalesny, L. Van Gool, "Viewpoint Consistent Texture Synthesis", *3D Data Processing, Visualization and Transmission (3DPVT)*, 2004.

S. Peleg, J. Naor, R. Hartley and D. Avnir, "Multiple resolution texture analysis and classification," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 6, pp. 518-523, 1984.

J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients", *Int'l Journal of Computer Vision*, Vol. 40(1), pg. 49-71, 2000.

C. Schmid, "Constructing models for content-based image retrieval", *CVPR*, Vol. 2, pp. 39- 45, 2001.

A. Teuner, O. Pichler, and B. J. Hosticka, "Unsupervised texture segmentation of images using tuned matched Gabor filters," *IEEE Trans. on Image Process*, Vol. 4(6), pp. 863-870, 1995.

UIUC dataset, `http://www-cvr.ai.uiuc.edu/ponce_grp/data/index.html`.

UMD dataset, `http://www.cfar.umd.edu/users/fer/website-texture/texture.htm`.

M. Varma and R. Garg, "Locally invariant fractal features for statistical texture classification", *Proc. IEEE International Conference on Computer Vision*, 2007.

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

M. Varma and A. Zisserman, "Texture classification: Are filter banks necessary?", *CVPR*, Vol. 2, pp. 691-698, 2003.

J. L. Vehel, P. Mignot, and J. P. Berroir, "Multifractals, texture, and image analysis", *CVPR*, pp. 661664, 1992.

I. Weiss, "Geometric Invariants and Object Recognition," *Int'l Journal of Computer Vision*, Vol. 10(3), pp. 207-231, 1993.

H. Wendt, P. Abry and S. Jaffard, "Bootstrap for Empirical Multifractal Analysis," *IEEE Signal Processing Magazine*, 38-48, 2007.

Y. Xia, D. Feng and R. Zhao, "Morphology-based multifractal estimation for texture segmentation", *IEEE Transaction on Image Processing*, Vol. 15 (3), pp. 614-623, 2006.

Y. Xu, H.Ji and C.Fermüller "A projective invariant for textures", *CVPR* Vol. 2, pp. 1932-1939, 2006.

A. Zalesny and L. Van Gool, "Multiview Texture Models", *CVPR*, Vol. 1, 2001.

S.C Zhu, Y. Wu, and D. Mumford, "Filters, random-fields and maximum entropy (FRAME): Towards a unified theory for texture modeling", *Int'l Journal of Computer Vision*, Vol. 27(2), pp. 107-126, 1998.

A Moment-based estimator for the MFS

First, we partition R^2 into r-mesh squares. Then we define the q-th moment of a measure μ as

$$M_r(q) = \sum \mu(B(\boldsymbol{x}, r))^q, \tag{9}$$

where the sum is over the r-mesh squares $\{B(\boldsymbol{x}, r)\}$ for which $\mu(B(\boldsymbol{x}, r)) > 0$. For a series of $r = 1, 2, \cdots, n$, we have corresponding measurements $M_r(q)$. Then the power law behavior of $M_r(q)$ is identified by the number $\beta(q)$ which is defined as:

$$\beta(q) = \text{Slope of } \log(M_r(q)) \text{vs } \log(r). \tag{10}$$

In other word, $\beta(q)$ is the slope of the line which best fits the data sets:

$$\{\log(M_r(q)), \log(r)\}.$$

It has been shown in (Falconer 1997) that the MFS and $\beta(q)$ are related to each other by a Legendre transform as

$$f(\alpha(q)) = q\alpha(q) + \beta(q), \tag{11}$$

where

$$\alpha(q) = -\frac{d\beta(q)}{dq}. \tag{12}$$

Using equations (11) and (12) we estimate the MFS. It can be seen that we could obtain a measure of the reliability of $\beta(q)$ from the fitting residual in (10). However, after the Legendre transform, we don't have any idea how to estimate the reliability of $f(\alpha)$, anymore.

B Proof of Theorem 1.

Let g denote a bi-Lipschitz transform on the spatial variable $\boldsymbol{x}$. Then the new image after applying such a geometrical transform on the original image $I(\boldsymbol{x})$ can be written as $I'(\boldsymbol{x}) = I(g(\boldsymbol{x}))$. In order to prove the invariance of the MFS to bi-Lipschitz geometrical transforms, we first prove that for each α,

$$E'_\alpha = g(E_\alpha),$$

where E'_α denotes the corresponding level set of $I'(x)$. Following, we prove that $g(E_\alpha)$ has the same fractal dimension as E_α.

For a constant α, let $\boldsymbol{x}$ be any point in the set of E_α, then from (3) we have

$$\lim_{r \to 0} \frac{\log \mu(B(x,r))}{\log r} = \alpha.$$

Thus, for the point $g(\boldsymbol{x})$ in the new image $I'(\boldsymbol{x}) = I(g(\boldsymbol{x}))$, we have

$$\frac{\log \mu((B(g(\boldsymbol{x}), r)))}{\log r} \leq \frac{\log c_2^2 \mu(B(\boldsymbol{x}, r))}{\log r}$$
$$= \frac{2\log c_2 + \log \mu(B(\boldsymbol{x}, r))}{\log r}.$$

Similarly, we have

$$\frac{\log \mu(B(g(\boldsymbol{x}), r)))}{\log r} \geq \frac{2\log c_1 + \log \mu(B(\boldsymbol{x}, r))}{\log r}.$$

Taking the limit on both sides of the two inequalities above, we obtain

$$\lim_{r \to 0} \frac{\log \mu(B(g(\boldsymbol{x}), r))}{\log r} = \lim_{r \to 0} \frac{\log \mu(B(\boldsymbol{x}, r))}{\log r} = \alpha.$$

Thus, we conclude that $E'_\alpha = g(E_\alpha)$.

To complete the proof, we need to show that the fractal dimension of $g(E_\alpha)$ and the fractal dimension of E_α are the same. Suppose E_α is covered by $N(\delta, E_\alpha)$ sets, which is the smallest number of sets with diameter less than δ. Then the $N(\delta, E_\alpha)$ images of these sets under g form a cover of $g(E_\alpha)$ by sets of diameter less than $c_2\delta$. By the definition the fractal dimension, we have

$$\dim(g(E_\alpha)) = \lim_{c_2\delta\to 0} \frac{\log N(c_2\delta, g(E_\alpha))}{-\log c_2\delta} \le \lim_{\delta\to 0} \frac{\log N(\delta, E_\alpha)}{-\log c_2\delta}$$

$$= \lim_{\delta\to 0} \frac{\log N(\delta, E_\alpha)}{-\log c_2 - \log \delta} = \lim_{\delta\to 0} \frac{\log N(\delta, E_\alpha)}{-\log \delta}$$

$$= \dim(E_\alpha).$$

On the other hand, suppose $g(E_\alpha)$ is covered by $N(\delta, g(E_\alpha))$ sets, which is the smallest number of sets of diameter at most δ. The same argument yields

$$\dim(E_\alpha) \le \lim_{\delta\to 0} \frac{\log N(\delta, g(E_\alpha))}{-\log c_1^{-1} - \log \delta} = \dim(g(E_\alpha)).$$

Thus, $\dim(E_\alpha) = \dim(g(E_\alpha))$. The MFS of the new image $I(g(\boldsymbol{x}))$ equals the MFS of the original image $I(\boldsymbol{x})$. $\qquad\square$