On Robustness of Multi Fractal Spectrum to Geometric Transformations and Illumination

Samir H. Abdul Jauwad[1] and Rehmat Ullah[2]

[1] Deptt of Electrical Engg, King Fahd Univ of Petroleum & Minerals Dhahran, Saudi Arabia
[2] Deptt of Computer Systems Engg, Univ of Engg & Technology Peshawar, Pakistan
samara@kfupm.edu.sa, rehmatkttk@nwfpuet.edu.pk

Abstract. Regular region based segmentation approaches utilize color or intensity information to distinguish between different regions. The performance of such procedures is acceptable for man-made objects, since they mainly consist of regular shapes and smooth surfaces. Most natural objects such as mountains, trees or clouds on the other hand are typically formed of complex, rough and irregular surfaces in 3-D, which are transformed into textured regions on the 2-D image plane through the image formation process. If we want to segment or classify images of such surfaces in an automatic fashion, we need to find a way to capture the essence of their structure succinctly. This can be achieved by making use of the 3-D information in combination with the texture of the corresponding region on the image plane. One possible way to model this relationship is through fractal analysis, which has proven to be a good representation of natural objects.

This paper is based on the work of Xu et al. [1]. The aim is to explain a way of efficiently representing natural objects by the use of the multi fractal spectrum (MFS), which is an extension of the regular fractal analysis. Section 1 of this paper will present the concepts of image texture. Then section 2 will briefly introduce the fractal theory and section 3 will elaborate on the concept of fractal dimension (FD). Section 4 is the main part of the paper, which will explain the MFS as a robust and invariant texture descriptor. The final section will present the experimental results obtained by Xu et al. [1] and provide conclusions.

Keywords: Image Texture, Fractal analysis, Fractal dimension (FD), Multi fractal spectrum (MFS), Bi-Lipschitz transformations.

1 Image Texture

Several image properties such as smoothness, coarseness, depth, regularity, etc. can be intuitively associated with textures. However there is no accepted formal definition for texture. Many researchers have described texture as a descriptor of the local brightness variation from pixel to pixel in a small neighbourhood around a central pixel. Alternatively, texture can be described as an attribute representing the spatial arrangement of the gray levels of the pixels in a region. Texture analysis has played an important role in many areas including medical imaging, remote sensing and

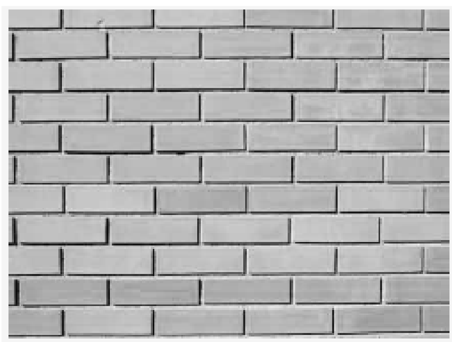
A. Abd Manaf et al. (Eds.): ICIEIS 2011, Part II, CCIS 252, pp. 76–95, 2011.

industrial inspection, and its tasks are mainly classification, segmentation, synthesis, compression and scene description. Human touch is a useful way of interpreting texture, since we naturally relate surface structure to touch. In that sense a texture can be rough, silky, bumpy, etc.

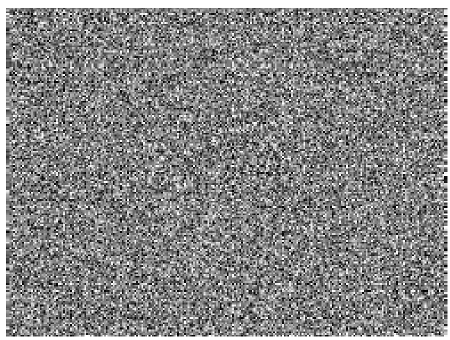
Textures can be categorized into repetitive, stochastic, mixed and fractal patterns. Examples of that are illustrated in Fig. 1. Fig. 1(a) shows a repetitive texture, which is obtained by replicating a texture primitive (brick) in the image X and Y directions. Fig. 1(b) shows a stochastic texture, which is generated by sampling a random process. Fig. 1(c) illustrates that different combinations of texture types can be present in the same image. Fig. 1(d) shows a fractal texture, which will be introduced in further detail in section 2.

A texture descriptor is a discriminant metric to quantify the perceived texture of a surface. A good descriptor should have the following properties:

- Rich informative surface description
- Spatial invariance to geometrical transformations
- Illumination invariance
- Efficient computation (especially for real time systems)

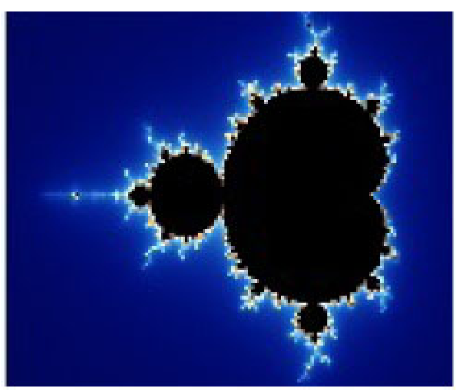


(a) Repetitive texture



(b) Stochastic texture

(c) Mixed texture



(d) Fractal texture

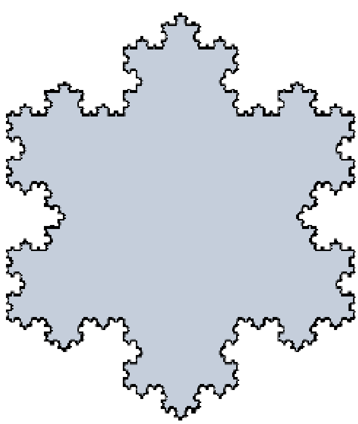
Fig. 1. Texture categories

There are several techniques to obtain a descriptor for a textured image, which can be divided into two main categories. The first one follows a structured approach, which sees an image as a set of primitive texels in a regular or repeated pattern. This approach is particularly useful for describing artificial textures. The second one is a statistical approach, which computes a quantitative measure of the arrangement of the intensities in a region. In general this approach is easier to compute and is more widely used, since natural textures consist of irregular sub-elements. Popular techniques in this category are the *gray level co-occurrence matrices*, *laws masks*, *autocorrelation coefficients* and *fractal analysis*. The latter rather new technique will be discussed in the next section.

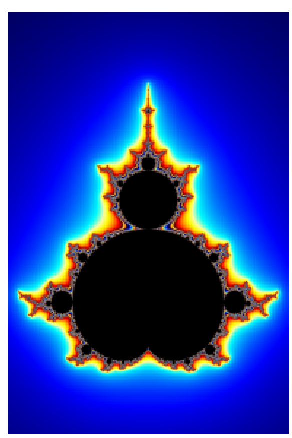
2 Fractal Theory

A fractal is a self-similar geometric shape, i.e. it is made up of reduced size copies of itself. Fractals are defined recursively and can be constructed in an iterative fashion. The roots of the fractal theory go back to the 17th century; however the mathematical concepts were developed one century later. The term fractal was coined by Benoit Mandelbrot in 1975 and was derived from the Latin word fractus, which means "broken" or "fractured". A key property of fractals is that they comprise finite area, but have an infinitely long boundary. No small segment along the boundary of fractal is line-like and the distance between any two distinct points on the boundary is infinite. There are two classes of fractals, depending on the level of self-similarity. Exact self-similar fractals consist exclusively of exact copies of itself. On the other hand quasi self-similar fractals are made up of several structures that are repeated recursively. Two popular examples for the fractal classes are shown in Fig. 2.

As mentioned earlier fractals are constructed iteratively. In order to illustrate this process iterations 0 to 4 of the Koch curve are shown in Fig. 3. The construction starts



(a) Koch curve



(b) Mandelbrot set

Fig. 2. Fractal classes

at iteration 0 with an equilateral triangle. Then in every iteration the middle third of every straight line segment is replaced by a pair of line segments that form an equilateral bump (inclination angles of 60^0 and 120^0). If this process is repeated forever, the boundary will become infinitely long. The Koch curve is a good approximation of the physical formation of snowflakes. Another popular example is the Sierpinski triangle.

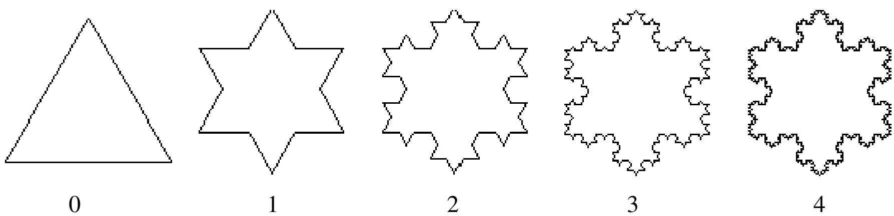


Fig. 3. Koch curve: demonstration of fractal construction

3 Fractal Dimension

The fractal dimension (FD) is a key quantity in the field of fractal texture analysis. It is an indicator of how completely a fractal fills the surrounding space and it is also a measure of the irregularity of the point distribution of an object. The mathematical formulation for computing the FD of a point set $E \in \Re$ is

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

where δ is the measurement scale, $N(\delta, E)$ is the number of copies of the original object when going to a smaller scale and $\dim(E)$ is the fractal dimension of the point set. For each δ, an object is measured in a way that ignores irregularities of size less than δ. The fractal dimension is then obtained by observing how these measurements change as δ approaches 0. One way to illustrate the measurement scale is to think of δ as being the length of a ruler that is used to measure the boundary of the object in question. Fig. 4 shows an example of measuring the length of the coast line of Great Britain using rulers of decreasing sizes. As can be seen the smaller the ruler is, the more detail can be captured, i.e. the better the approximation of the coast line is. No matter at what scale we look at the coast line there will always be scalloping smaller than that scale. This means that the estimated length of the coast line increases as the ruler size decreases. Hence the length of the coast line (or in general the outline of an object) does not just depend on the length of the coast line itself, but also on the length of the measurement tool. In order to obtain a consistent measurement of the coast line, the notion of dimension must be generalized to include fractional dimensions. In this way the unique fractional power that yields consistent estimates of the object's metric properties is that object's fractal dimension (reason for the limit in Eqn. 1).

Fig. 4. Measuring the coast line of Great Britain using a ruler

However when analysing image textures the theoretical way of computing the fractal dimension demonstrated above cannot be applied, since digital images are recorded at a finite resolution and it is not possible to go beyond single pixel resolution. That's why in practice the so-called box-counting dimension is used. Fig. 5 shows the same example of measuring the coast line of Great Britain, using circular disks of decreasing radius *r*. In general the box-counting dimension is computed by covering the space with a mesh of boxes with side length *r*, called the *r* mesh boxes (*r* squares in 2-D) and then counting how many boxes are needed to cover the whole structure.

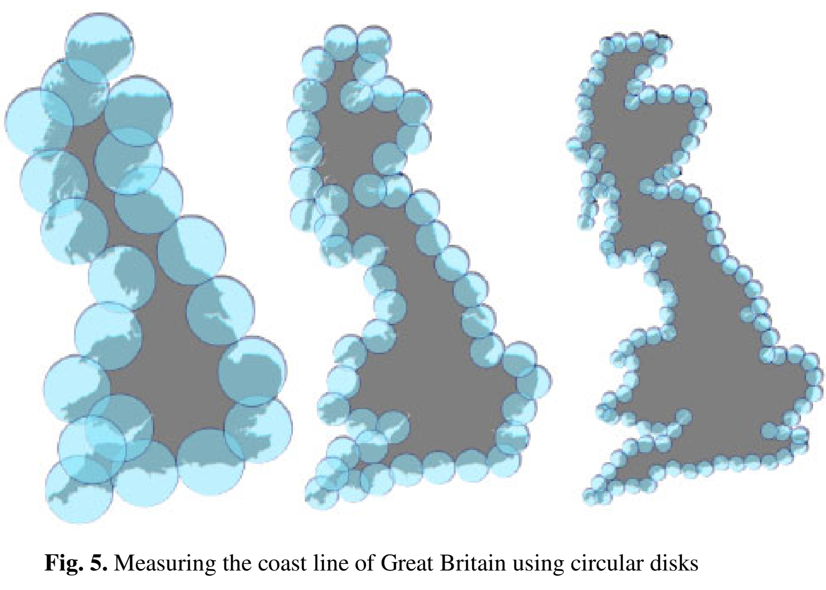


Fig. 5. Measuring the coast line of Great Britain using circular disks

3.1 Computation of the FD

In order to illustrate the concept of FD more clearly, we will show the computation of the FD of the primitive objects that cover up the whole 1-D, 2-D and 3-D space respectively and in contrast to that the FD of a real fractal. Fig. 6-8 have the following structure: the left subfigure shows the original object, the right one shows the result when doubling its extent (scaling factor k of 2) in all dimensions. This scaling factor k determines the number of replications of the object in one dimension. The measurement scale δ on the other hand determines the reduction in length of the ruler. Since they describe the same concept in two opposite ways, their mathematical relation is $k = \frac{1}{\delta}$. In Fig. 6 it can be seen that the number of copies of the original object after the dilation is 2 and hence $\delta = 0.5$. So the FD can be computed using Eqn. 1 as

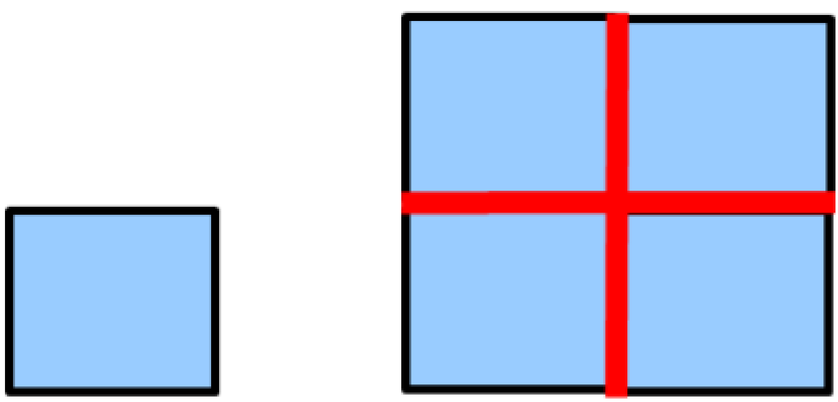
$$\dim(L) = \lim_{\delta \to 0} \frac{\log N(\delta, L)}{-\log \delta} = \frac{\log 2}{-\log 0.5} = 1$$

(a) Original line (b) Dilated line

Fig. 6. Fractal dimension of a 1-D line

Fig. 7 shows that the original square is replicated four times when both dimensions are doubled. So the number of copies $N = 4$. The measurement scale δ is still 0.5 and hence the FD can be computed as

$$\dim(S) = \lim_{\delta \to 0} \frac{\log N(\delta, S)}{-\log \delta} = \frac{\log 4}{-\log 0.5} = 2$$



(a) Original square (b) Dilated square

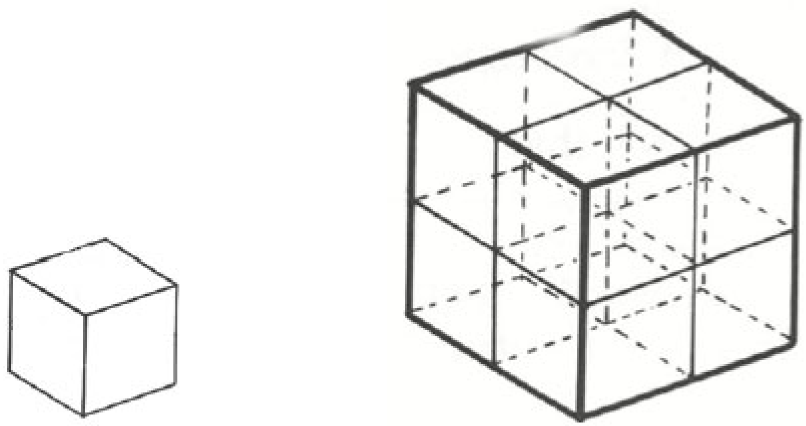
Fig. 7. Fractal dimension of a 2-D square

As can be seen in Fig. 8 the original cube is replicated eight times when all three dimensions are doubled and so the number of copies $N = 8$. The measurement scale did not change and hence the FD is as follows

$$\dim(C) = \lim_{\delta \to 0} \frac{\log N(\delta, C)}{-\log \delta} = \frac{\log 8}{-\log 0.5} = 3$$

From the examples given above we can conclude that regular geometrical objects

- o have an integer fractal dimension
- o have a fractal dimension which is equal to their topological dimension



(a) Original cube (b) Dilated cube

Fig. 8. Fractal dimension of a 3-D cube

As an example of the FD of a real fractal we show the Koch curve. The scaling factor $k = 3$, so $\delta = \frac{1}{3}$. As can be seen in Fig. 9 the number of copies of the original object (shown in black) is $N = 4$. So its fractal dimension can be computed as

$$\dim(K) = \lim_{\delta \to 0} \frac{\log N(\delta, K)}{-\log \delta} = \frac{\log 8}{-\log \frac{1}{3}} \approx 1.2619$$

In contrast to regular objects the fractal dimension of an irregular object is always a fractional number (hence the name fractal dimension).

Another way of illustrating the concept of fractal dimension is to increase gradually the roughness of a 3-D plane from a completely smooth surface to a highly irregular one and observing how its FD changes (as shown in Fig. 10). The 3-D surface irregularities are projected as texture on the 2-D image plane through the image formation process. By computing the FD of the texture, we can see that it is directly proportional to the roughness of the corresponding surface: the higher the roughness, the higher the FD and vice versa.

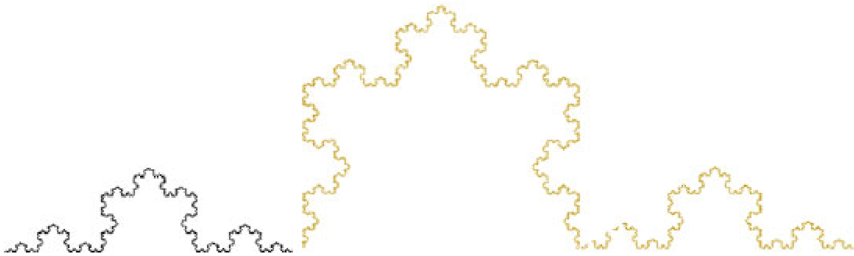
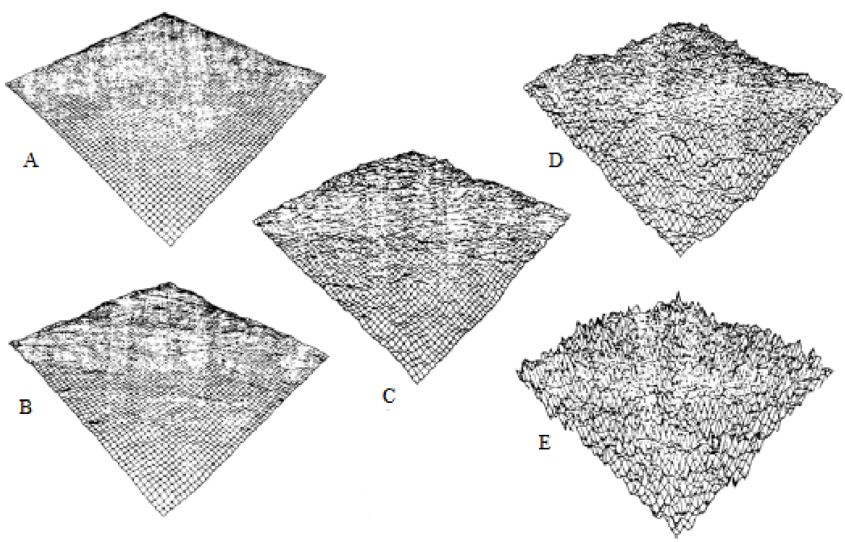


Fig. 9. Fractal Dimension of the Koch Curve

3.2 Fractal Dimension of Images

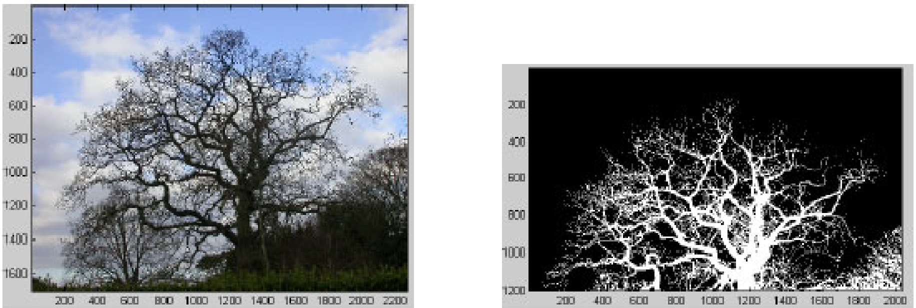
Up till now we have only discussed how to compute the fractal dimension of differently shaped objects. In order to utilize the FD for texture analysis we need to extend the concept to the space of images. A crucial property of shapes is that for every point in space, we know if it belongs to the shape or not. Hence it is possible to compute its fractal dimension. However images consist of a dense matrix of RGB or intensity values. So a pre-processing step is necessary to binarize the images. Then the FD of the binary images can be computed, where white means that a pixel belongs to the shape (logical true) and black means that it does not (logical false). This binarization (or categorization) can be achieved in various ways, depending on the task at hand and the input image



A: FD ≈ 2.0, B: FD ≈ 2.1, C: FD ≈ 2.5, E: FD ≈ 2.8

Fig. 10. Surfaces of increasing fractal dimension

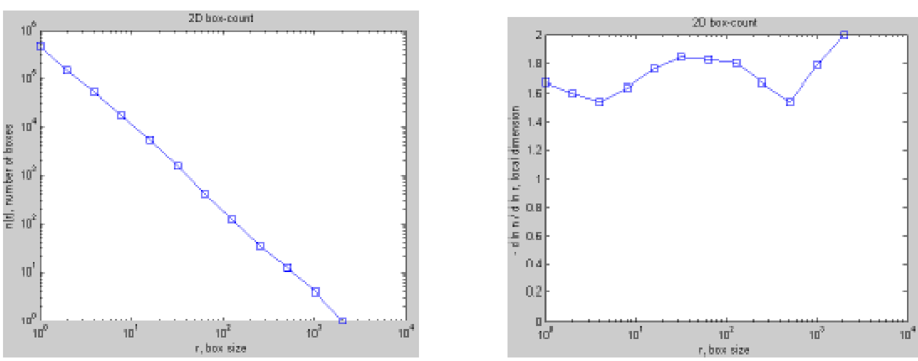
class. In order to illustrate the concept we will start by simple thresholding. Fig. 11 shows the original color image of a tree and its binarized version obtained by thresholding the blue channel[1]. All pixels with blue channel intensities below 80 are set to white, the rest of the pixels are set to black. The binary version clearly shows the shape of the object of interest (tree) in relation to the background.



(a) Original tree image (b) Binarized tree image

Fig. 11. Binarization of an RGB image for computation of its FD

Fig. 12 shows the fractal dimension of the binarized tree image from Fig. 11(b) as a function of the box size used to cover the shape. The first thing to observe is that the fractal dimension for a box size of 2000 pixels or more is 2, i.e. that the full image is covered by a single box and hence the FD corresponds to the topological dimension. But we know from Eqn. 1 that the FD is defined for the box size *r* going to zero. As the box size approaches zero, we get different values for the FD. Since an image is



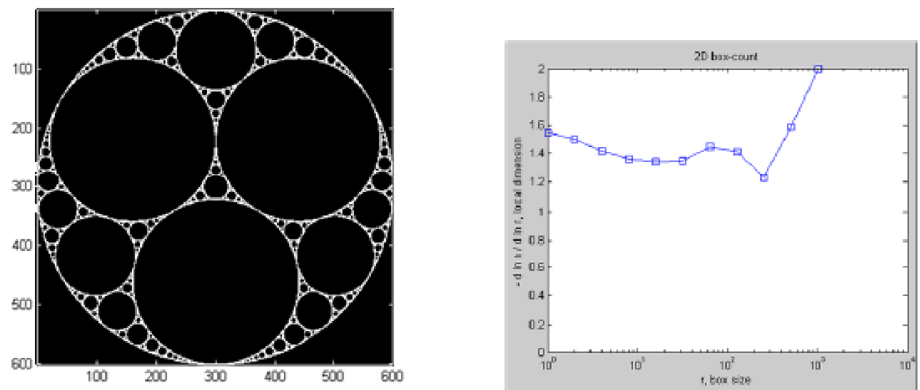
(a) Number of boxes (b) Fractal dimension

Fig. 12. Functions w.r.t. the box size of the tree image

[1] The binary image just contains a centred, reduced size version of the original image, because it represents the most descriptive part of the tree shape.

recorded at a final resolution the minimum box size is one pixel (at the left border of the graph). The final FD is obtained by taking the average of the finite differences of the number of boxes with respect to the box size. For the tree image it is approximately 1.801 ± 0.06394. The FD can just be estimated, because of two reasons: firstly, the finite resolution of the image just allows for finite differencing and not for an analytical derivative and secondly, if the image size is not an integer power of 2, the boxcount[2] function pads the remaining space with zeros (e.g. a 320x200 image is padded to 512x512) and hence that region is not considered to be part of the shape.

Fig. 13 shows the computation of the fractal dimension for a binary image that contains a fractal-like shape. We can observe that the line evolves differently w.r.t. the previous figure.



(a) Original fractal image

(b) Fractal dimension

Fig. 13. FD of a fractal gray scale image

3.3 Issues with the FD

The fractal dimension analysed up to now has several advantages. It is insensitive to image scaling, it is close to the human perception of surface roughness and it is a good representation of natural objects. But there are also two main drawbacks, namely that the FD just gives a single value for a texture and that different textures can have the same FD. Hence it is not a unique and very rich descriptor. A possible solution to overcome the mentioned limitations is the extension to the multi fractal spectrum (MFS), which will be covered in the following section.

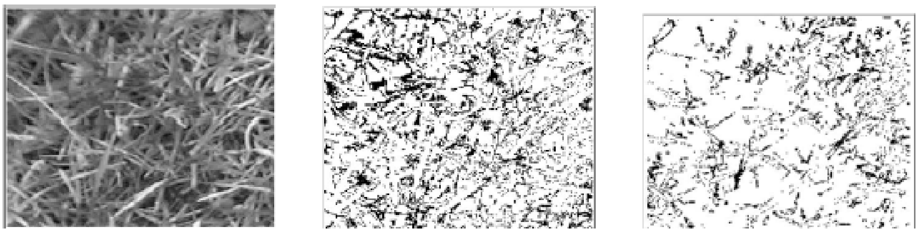
4 Multi Fractal Spectrum

The MFS is a vector of the fractal dimensions of some categorization of the image. Here, a categorization is a way of creating binary sub images from the original image

---

[2] A Matlab function.

using some thresholding criteria (as shown in the tree example in the previous section). A more general way to do this is intensity thresholding: divide the overall range (e.g. 0-255) in *N* bins, and for every bin compute a binary image by setting all the pixels in the bin to black and all others to white. In this way *N* binary images are obtained. Fig. 14 shows this process for a grass texture.



(a) Original Image (b) Thresholded between 100 and 120 (c) Thresholded between 80 and 100

Fig. 14. Binarization of grass texture

The computation of the MFS is a three step process, which is illustrated in Fig. 15. It consists of the following stages:

I. Extract N binary images from the original image
II. Compute the fractal dimension of all the binary images
III. Concatenate all the computed FDs into one vector, called the Multi Fractal Spectrum

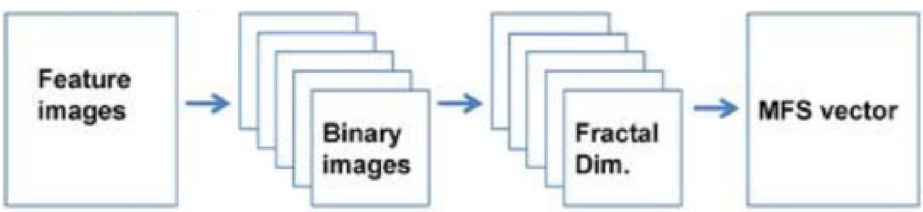


Fig. 15. Overview of the FSM computation

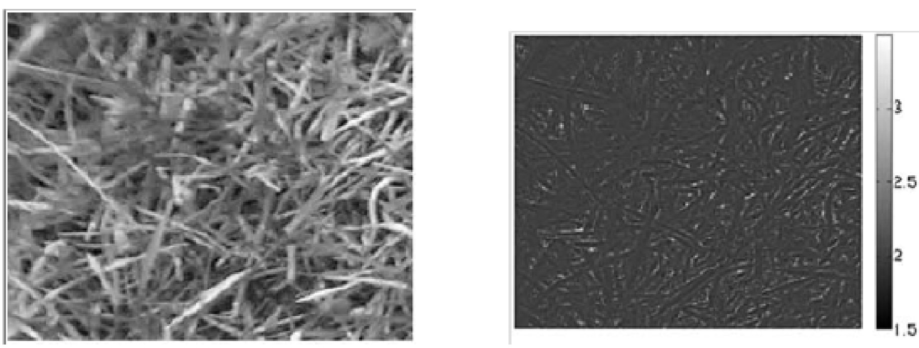
An important aspect of this process is the way how the categorization is achieved. The classical approach in the MFS literature (which was also followed in [1]) introduces a local density function for this task. It is also called Hölder exponent or local fractal dimension, since its formula resembles the one for computing the normal FD. The crucial difference is that instead of having the number of boxes in the numerator, a measurement function is used.

4.1 Local Density Function

The local density function for an image pixel x is defined as:

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

where $\mathrm{d}(x)$ is the local density at the image pixel x; x is a two element vector containing the *(x, y)* position of the central pixel, r is the radius of the circle or the side length of the square used to define the neighbourhood around the central pixel, $B(x,r)$ is the area with center x and radius r, μ is a measurement function that returns a single, representative value for the given area. The density function describes how locally the measurement function μ satisfies the power law behaviour. It measures the "non-uniformness" of the intensity distribution in the region neighbouring the measured point. In practice, the local density is obtained as the slope of the line fitted to the data $\{\log r, \log \mu(B(x,r))\}$, where r takes on several discrete values in a fixed range. When the local density is computed for every pixel in the input image a density image is obtained. Fig. 16 shows the original grass texture and the corresponding density image.



(a) Original grass texture (b) Density image

Fig. 16. Computing the local density image from a texture

There are several possibilities to define the measurement function μ. The first approach is to work directly on the intensity domain. Denoting with $I(x)$ the intensity of pixels x, $\mu(B(x,r))$ is defined as:

$$\mu_1(B(x,r)) = \iint_{B(x,r)} (G_r * I)\,dx \tag{3}$$

where "*" is the 2-D convolution operator and G_r is a Gaussian smoothing kernel with variance r.

$$G_r = \frac{1}{r\sigma\sqrt{2\pi}} e^{\frac{-\|x\|^2}{2\sigma^2 r^2}} \tag{4}$$

where σ is a predefined parameter. In other words, $\mu(B(x,r))$ is the sum of average intensity values inside the disk $B(x,r)$. Since the variance of the Gaussian kernel depends on the neighbourhood size r, it encodes how the intensity at a point changes over scale. When using this measurement function, the final MFS vector contains the fractal dimension for multiple values of the density of the intensity.

The measurement function shown in Eqn. 3 is simple to compute but not robust to large illumination changes. To overcome this problem, several meaningful definitions of μ can be introduced. One possibility is to take the differential operators along the four main directions (horizontal, vertical, diagonal, anti-diagonal). The new measurement function

$$\mu_2(B(x,r)) = \left(\iint_{B(x,r)} \sum_K (f_k(G_r * I))^2 \right)^{\frac{1}{2}} dx \tag{5}$$

is illumination invariant, since the finite differencing due to the differential operators removes the effects of linear additive illumination changes. Another variant for the measurement function is

$$\mu_3(B(x,r)) = \int\int_{B(x,r)} \left| \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) (G_r * I) \right| dx \tag{6}$$

which is the sum of the Laplacians of the image inside $B(x,r)$. The definition of the measurement function is important, since it directly affects the final MFS vector. A combined MFS vector obtained by using different measurement functions leads to a better representation of the texture. Fig. 17 shows the output images obtained by using different definitions of the measurement functions and a graph illustrating the three corresponding MFS vectors. As can be seen the vectors are substantially different from each other. The graph shows the fractal dimension $f(\alpha)$, where $\alpha \in \Re$ is a level of local density.

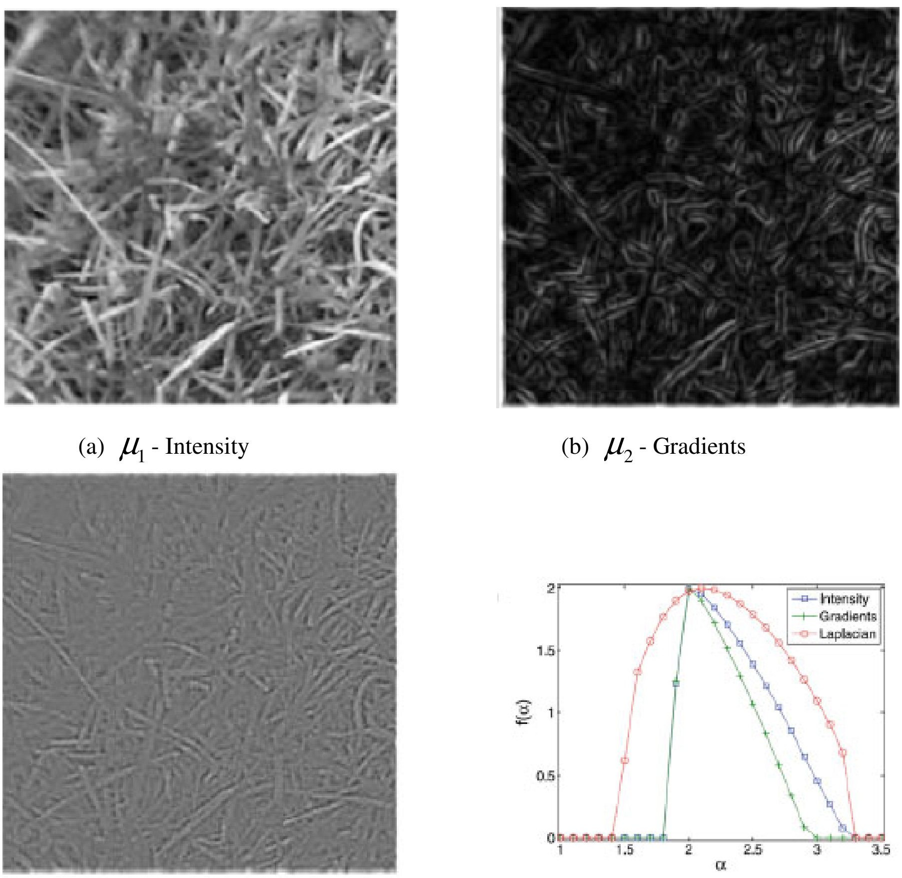
Fig. 18 shows the four different textures and a graph with their corresponding MFS. It demonstrates that the MFS vectors of different textures are significantly different. The MFS of the carpet texture is completely different from the other MFS vectors. This is due to the fact that the carpet texture is stochastic (is comprised by a random pattern), whereas the other three textures show natural images of plants that are more similar to each other. We can also observe that as the surface roughness decreases from 18(a) to 18(d), the shape of the main bump of the MFS vectors becomes gradually narrower.

4.2 Properties of the MFS

The most relevant property of the MFS texture descriptor is its invariance to geometrical transformations under the Bi-Lipschitz map. A Bi-Lipschitz transform is a general class of spatial transformation including translation, rotation, projective transformation and texture warping on regular surfaces. It is defined as a function $g : \Re^2 \rightarrow \Re^2$, if there exists two constants $c_1 > 0, c_2 > 0$ such that for any $x, y \in \Re^2$ the following relation holds:

$$c_1 \|x - y\| < \|g(x) - g(y)\| < c_2 \|x - y\| \tag{7}$$

This spatial invariance is a very powerful feature and it has been mathematically proven for images with infinite resolution. In practice we are dealing with finite resolution images, but the MFS has shown to be very robust to perspective transformations and warping of the surface. Usually four to five levels of resolution (radius *r* used to compute the FD) are sufficient for a good estimation of the fractal dimension. Fig. 19 shows an example of the grass texture, shown earlier in Fig. 14(a),



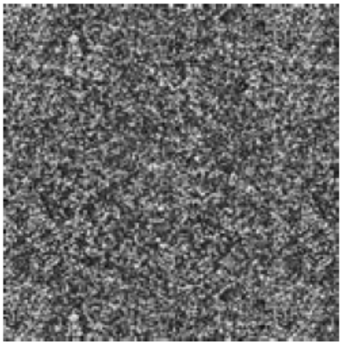
(a) μ_1 - Intensity

(b) μ_2 - Gradients

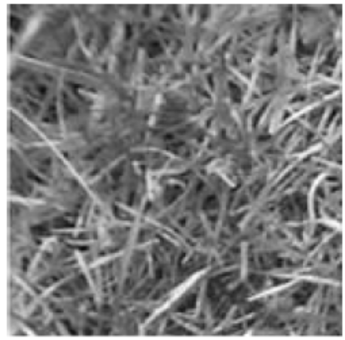
(c) μ_3 - Laplacians

(d) Resulting MFS Vectors

Fig. 17. Comparison of different measurement functions



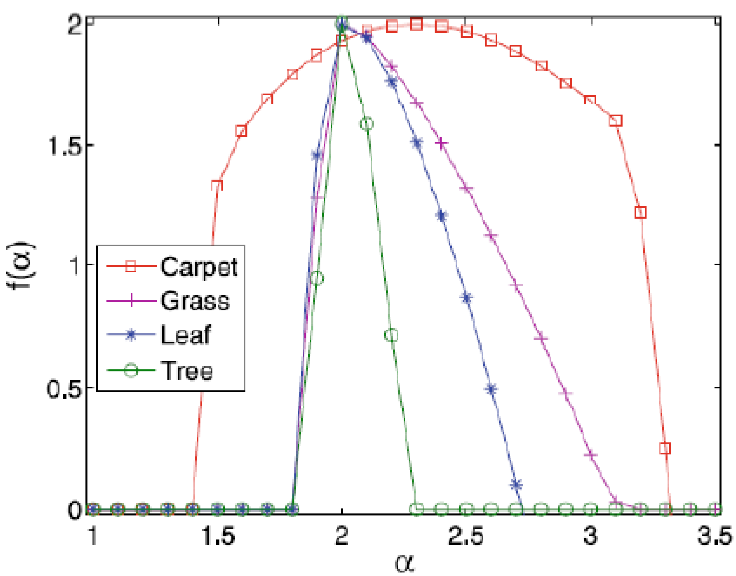
(a) Carpet



(b) Grass

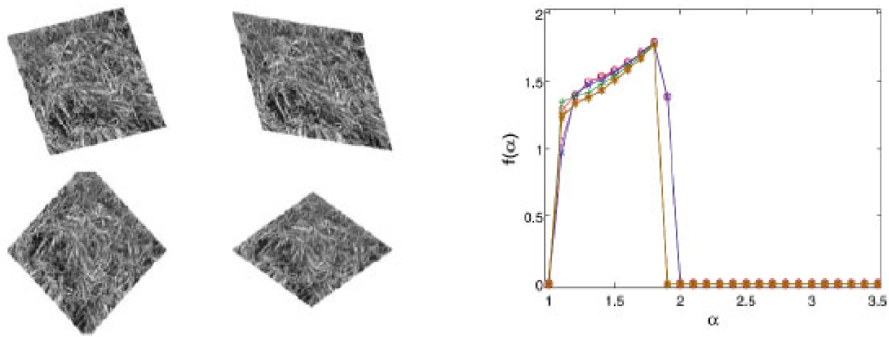
(c) Leaf

(d) Tree



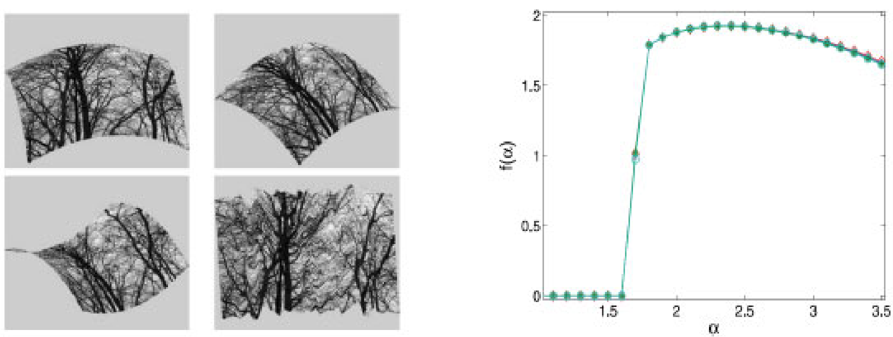
(e) Multi Fractal Spectra of the density of the four textures above

Fig. 18. Comparison of the MFS vectors of different textures



(a) Bi-Lipschitz transformations (b) Resulting MFS vectors

Fig. 19. Four perspective views of a foliage texture and the corresponding MFS vectors of the density of intensity



(a) Warping (b) Resulting MFS vectors

Fig. 20. Perspective images of tree texture on different general smooth surfaces and the corresponding MFS vectors

which has been deformed using Bi- Lipschitz transformations. The resulting MFS vectors are displayed next to it. As can be seen they are very similar and hence the different images would be matched to the same texture prototype.

Fig. 20 shows another type of Bi-Lipschitz transform, namely the warping of a perspective transformation of the texture onto a smooth surface. As can be observed the resulting MFS vectors are nearly identical and hence the MFS is very robust to this type of transformations.

In addition to the spatial invariance the MFS descriptor is invariant to local multiplicative changes in image illumination, if the intensity measurement function μ_1 is used, and also to local linear changes in illumination if μ_2 or μ_3 are used. The first invariance is due to the fact that the ratio of logarithms in Eqn. 1 does not change due to multiplicative changes (transformed into additive changes). The second invariance is due to the principle of finite differencing, namely that image derivatives always compensate for additive illumination changes.

4.3 Comparison of MFS and Histogram

The most used statistical texture descriptor is the histogram. It is efficient and simple to compute, since it just counts the number of elements in a bin according to a categorization of the image. However in this process all the information about the spatial distribution of the elements is lost. Hence it is not invariant to perspective transformations. It is also very sensitive to changes in illumination.

The multi fractal spectrum on the other side estimates the exponential changing ratio of the number of elements in a bin over multiple resolutions (different radii r in the computation of the local density). Accordingly it incorporates more geometrical information and is, as mentioned before, invariant to perspective transformations.

5 Experimental Results

In the paper of Xu et al. [1] the performance of the MFS descriptor was evaluated using classical texture retrieval and classification tasks. The MFS is compared with three other methods:

> **LSP:** was presented by Lazebnik et al. [2] in 2005 and is a sophisticated interest-point based texture representation. The main concept is to use elliptic clusters for characterizing a texture. It represents an image texture by its frequency of texture elements. It is robust to geometric transformations; however it is affected by changes in scale and viewpoint. Additional drawbacks of this method are: sophisticated pre-processing, K-means clustering and a large number of parameters. Hence it is complex and computationally expensive.
>
> **VZ-J:** was presented by Varma and Zisserman [3] in 2003 and uses a dense set of textons for the description task. It is very simple and non-invariant, but it has shown to outperform complex texton descriptors, based on the output of certain filters.
>
> **VG-F:** was presented by Varma and Garg [4] in 2007 and also makes use of the local density function. They apply the MR8 filter bank in a pre-processing step and their final feature vector contains 13 values. The descriptor for each image is a normalized histogram of the pixel texton labellings. The main difference to MFS is that they only use fractal geometry locally and do not integrate the standard global fractal dimension.

The experimental configuration of the MFS descriptor was obtained by using a combination of SVM cross-validation and the Fisher-score. The final vector had 33 dimensions and consisted of 13 intensity, 10 gradient and 10 Laplacian values (obtained by computing the local density images using μ_1, μ_2 and μ_3 respectively). A weighing of $\frac{1}{5}.(1,2,2)$ was applied to compose the vector. The distance between the vectors was computed using the absolute sum (L_1 norm). A nearest neighbour algorithm was used to classify the images, which were fetched from the UIUC

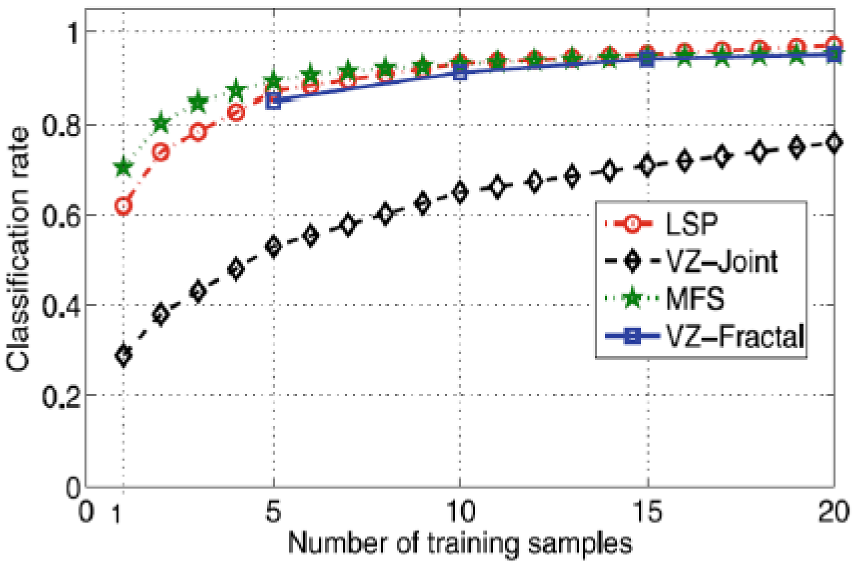


Fig. 21. Mean classification rate of 25 texture classes of MFS in comparison to three other methods

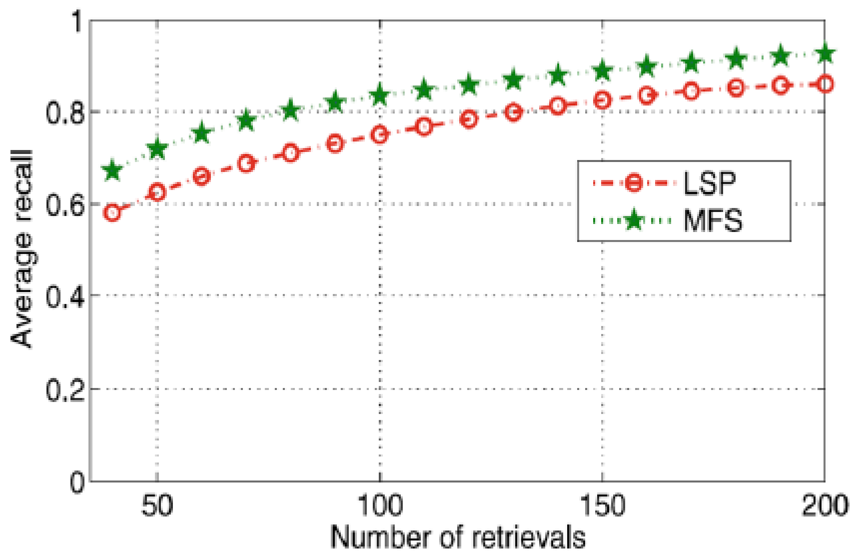


Fig. 22. Retrieval curves for the UIUC dataset for the MFS and LSP methods

repository. Fig. 21 shows the classification outcome of the MFS classifier as described before in comparison to the other three methods. It can be observed that the MFS classifier is clearly better than the VZ-Joint and comparable to the other two methods. The MFS performs better for a small number of training samples; however as the number of training samples (X-axis) increases the LSP method is more accurate. This is due to the fact that the LSP method is more robust to large illumination changes as compared to MFS.

Fig. 22 shows the retrieval curves on the UIUC dataset. Here just the performance of the LSP method is shown as a reference, since it is the most competitive one to the MFS. As can be seen the MFS accuracy is better by a noticeable margin. It is also necessary to mention that MFS uses a very small number of parameters as compared to LSP.

Fig. 23 shows a comparison between the performance of MFS and LSP on high resolution images taken from the UIUC dataset. The overall performance is very similar; however as the number of retrievals increases the performance of MFS is slightly better than LSP.

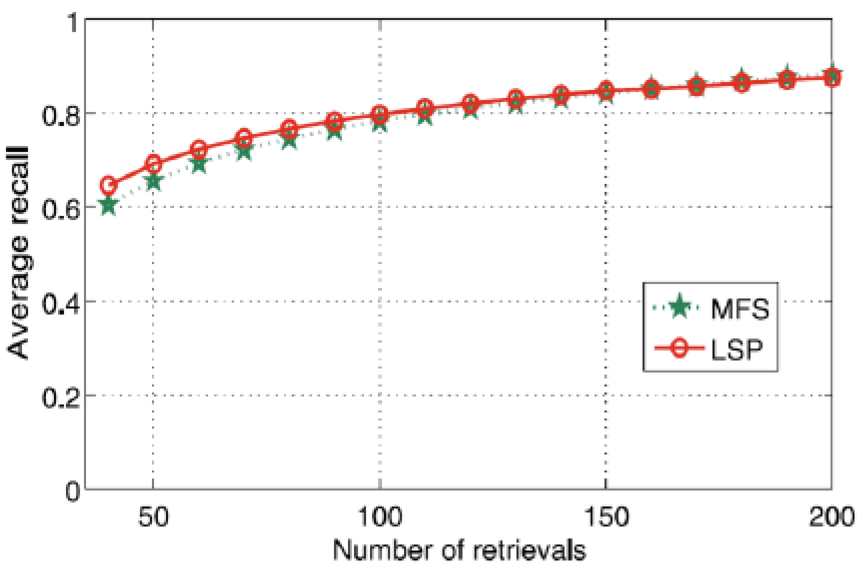


Fig. 23. Retrieval curves for high resolution images of the UIUC dataset for the MFS and LSP methods

6 Summary and Conclusions

In conclusion it can be said that the MFS is a promising texture descriptor. Its invariance to transformations under the Bi-Lipschitz map and illumination changes is proven mathematically and results in a robust descriptor. As shown in the experimental results it can be efficiently employed for texture retrieval and classification and its performance is similar or in some cases even better than comparable state-of-the-art texture classifiers.

A noticeable advantage w.r.t. to the other methods is that no feature detection, clustering or other pre-processing is required. Hence its computation is very efficient. It was also shown that the MFS is capable of utilizing the extra details that are present in high resolution images to create a more accurate texture description.

Acknowledgment. The authors wish to gratefully acknowledge *King Fahd University of Petroleum & Minerals Dhahran, Saudi Arabia,* for providing the funds to complete this work.

References

1. Xu, Y., Ji, H., Fermüller, C.: Viewpoint Invariant Texture Description Using Fractal Analysis. International Journal on Computer Vision (83), 85–100 (2009)
2. Lazebnik, S., Schmid, C., Ponce, J.: A Sparse Texture Representation Using Local Affine Regions. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(8), 1265–1278 (2005)
3. Varma, M., Zisserman, A.: Texture Classification: Are Filter Banks Necessary? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, vol. 2, pp. 691–698 (June 2003)
4. Varma, M., Garg, R.: Locally Invariant Fractal Features for Statistical Texture Classification. In: Proceedings of the IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil (October 2007)
5. Chaudhuri, B.B., Sarkar, N.: Texture segmentation using Fractal Dimension. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(1), 72–77 (1995)
6. Pentland, A.P.: Fractal-based description of natural scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence 6(6), 661–674 (1984)