



Image Denoising Using Fractal Hierarchical Classification

Swalpa Kumar Roy¹, Nilavra Bhattacharya²(✉), Bhabatosh Chanda³,
Bidyut B. Chaudhuri⁴, and Soumitro Banerjee⁵

¹ Department of CSE, Jalpaiguri Government Engineering College, Jalpaiguri, India
swalpa@cse.jgec.ac.in

² School of Information, The University of Texas at Austin, Austin, USA
nilavra@ieee.org

³ ECS Unit, Indian Statistical Institute, Kolkata, India
chanda@isical.ac.in

⁴ CVPR Unit, Indian Statistical Institute, Kolkata, India
bbc@isical.ac.in

⁵ DPS, Indian Institute of Science Education and Research, Mohanpur Campus,
Kolkata, India
soumitro@iiserkol.ac.in

Abstract. This paper proposes an efficient yet simple fractal-based image denoising technique. Denoising is carried out during fractal coding process. Hierarchical classification is used to increase encoding speed, and avoid a lot of futile mean-square-error (MSE) computations. Quadtree-based image partitioning using dynamic range and domain sizes is used to increase the degree of noise removal. Further denoising is achieved using pyramidal decoding, using non-arbitrary seed image, and additional post processing. Results from experiments show that our proposed scheme improves the structural similarity (SSIM) index of the Lenna image from 44% to 78% for low noise cases, and from 9% to 35% for high noise cases.

Keywords: Denoising · Fractal image coding · Image restoration

1 Introduction

Digital images are highly susceptible to contamination by noise during image acquisition and transmission. The commonest form of background noise found to corrupt digital images is Additive White Gaussian Noise (AWGN). AWGN can get into an image due to imperfect image-capturing device, insufficient illumination of the image subject during image capture, or due to transmission of the image over a noisy network. Background noise hampers both visual quality and object identification. Hence an efficient noise removal algorithm is required to remove or reduce its derogatory effects.

Linear filtering and smoothing operations are simple and popular methods for noise removal and image restoration. But their robustness is less, as they

hypothesize that the image consists of a stationary signal, formed through a linear system. However, real-world images are generally acquired using non-linear techniques, and the images have non-stationary statistical properties. The intensity distribution captured by the image acquiring device is a product of the illumination falling on the scene or object of interest, and its reflectance. Many non-linear and adaptive image denoising methods have been proposed which take these statistical variations into account, and therefore give better output image quality, while maintaining high frequency features of the input image [5,20].

In this paper, we examine the computational problem of estimating the original image f_{orig} from its noisy version f_{noisy} , corrupted by Gaussian noise. The problem may be defined as:

$$f_{noisy} = f_{orig} + G \quad (1)$$

where G is the additive white Gaussian noise. Noise removal, or denoising, is therefore the methodology of estimating the original signal f_{orig} from the noise-degraded signal f_{noisy} .

Proposed by Barnsley in 1988 [3], Fractal Image Compression (FIC) is a lossy image coding technique which exploits local self-similarities present in an image, and represents the image as a collection of affine transformations. Jacquin [13] presented the first automated and practical version of FIC, and it is known as the baseline fractal image compression (BFIC). BFIC uses Partitioned Iterated Function System (PIFS) to find matching image-patches without requiring human assistance. Recently Roy *et al.* [18] proposed a simple and efficient approximation of the scaling parameter which allows us to substitute to the expensive process of matrix multiplication with a simple division of two numbers. Hurtgen and Stiller [11] and Bhattacharya *et al.* [4] modified the Fishers method and obtained some improved performance. Since then, numerous FIC schemes have been proposed [7,9,23]. Apart from being an image compression technique, FIC has also been applied in various fields like image segmentation [6], image indexing and retrieval [17], image encryption [16], image authentication [15], and facial image recognition [21].

In this paper, we propose a fractal-coding based scheme for denoising an image corrupted by AWGN. Basic FIC theory is explained in Sect. 2. The proposed FIC based denoising scheme is discussed in Sect. 3. Experimental results of our scheme are shown in Sect. 4, and concluding discussions are presented in Sect. 5.

2 Fractal Coding

An Iterated Function System (IFS) is a collection of *contractive* affine transformations $\{w_i : \mathbb{R}^2 \rightarrow \mathbb{R}^2 | i = 1, \dots, n\}$ which map the plane \mathbb{R}^2 to itself. This collection of transformations defines a *map*:

$$W(\cdot) = \bigcup_{i=1}^n w_i(\cdot) \quad (2)$$

A contractive map is one, which brings points closer together. Hutchinson [12] proved that if w_i for all i in an IFS be contractive, then W is contractive. If a

contractive map is iterated from *any* initial point, it will always converge to a unique *fixed point*, which is called the *attractor* of the given contractive map. Given an input image S_0 , we can apply W on it repeatedly in a feedback loop to obtain the fixed point x_W , which is the limit set:

$$x_W \equiv S_\infty = \lim_{n \rightarrow \infty} W^n(S_0) \tag{3}$$

where $W^p = W\{w^{p-1}(\cdot)\}$. Thus the attractor or the fixed point x_W of the map W is not dependent on the choice of S_0 , and W alone can completely determine a unique image. This is called the *Contractive Mapping Fixed-Point Theorem* [1]. Also, if the error-difference between an original image f and the transformation of that image $W(f)$ is less than a certain threshold, the transform W is accepted as an equivalent representation of the image. This is the underlying concept of *Collage Theorem* [2]. FIC tries to find an IFS that maps an image onto itself. However, finding a single IFS that describes an entire image requires human intervention, and cannot be automated effectively. Jacquin [13] suggested that instead of finding transformations that describe a whole image, we can find transforms that apply only to portions of an image. These transformations are called Partitioned IFS (PIFS), which help to automate the encoding process.

2.1 Fractal Encoding

An image f is segmented into non-overlapping *range-blocks* R_i of size $r \times r$ and overlapping *domain-blocks* D_j of size $2r \times 2r$. The collection of all range blocks comprises the range pool R and the collection of all domain blocks comprises the domain pool D . Range blocks are smaller than domain blocks in order to make the transformations contractive. The domain pool is created by sliding a window of size $2r \times 2r$ from the top-left corner to the bottom-right corner of f with an integer step-size d in horizontal and vertical directions.

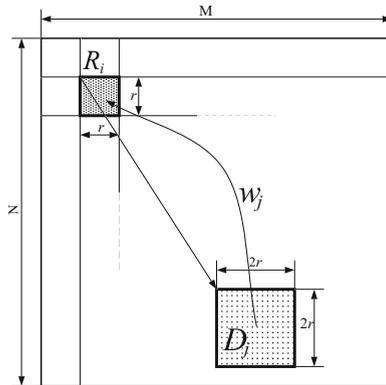


Fig. 1. Fractal encoding: mapping domains to ranges [11]

For each range block $R_i \in R$ an exhaustive search is carried out in the domain pool D to find a domain block D_j , such that D_j can be transformed “closely” to R_i . The contractive affine transformation w_j is then applied to D_j (Fig. 1). The optimal selection of w_j and D_j should ensure that the reconstructed image

$$g = W(D) = \bigcup_{j=1}^n w_j(D_j) \text{ (where } n = (M/r) \times (N/r) \text{: number of range blocks)}$$

has the minimum error-difference from the original image f . The transformation w_j and the location of the domain-block D_j comprise the *fractal code* of the range-block R_i . Fractal codes of all the range blocks collectively form the IFS of the image f . The contractive affine transform w_j is composed of a spatial transform and a grey level transform. The spatial transform is contractive, and it maps the spatial domain D_j (with size $2r \times 2r$) to the spatial range R_i (with size $r \times r$). The grey level transform G is of the form:

$$G(D'_j) = s_j \cdot D'_j + o_j \tag{4}$$

where D'_j is the spatially contracted domain block D_j to match the size of R_i it is mapping to (henceforth for brevity we use D_j to mean D'_j), s_j is the contrast or scaling factor, and o_j is the brightness or the offset of the transform. Using this definition, the FIC problem reduces to searching the domain pool D to find D_j , s_j and o_j for each R_i such that

$$E(R_i, D_j) < \varepsilon \tag{5}$$

where E is the error-metric between range block R_i and transformed domain block D_j , and ε is the target fidelity of the reconstructed image. A lot of image quality measures are available, but the mean square error (**MSE**) metric [8] is generally used as distortion criterion. So Eq. 5 becomes

$$\text{MSE}(R_i, D_j) = \|R_i - (s_j \cdot D_j + o_j)\|_2^2 < \varepsilon \tag{6}$$

where $\|\cdot\|_2$ is the two-norm.

2.2 Fractal Decoding

Fractal decoding is fast and recursive. An arbitrary initial “seed” image (typically a blank image) is chosen, and the encoded affine transforms are recursively applied on the image. By the principles of the *Contractive Mapping Fixed-Point Theorem* [1] and the *Collage Theorem* [2], the arbitrary image converges to the final decoded image within a few iterations.

3 Proposed Algorithm

In this section, we discuss the various components of our fast image denoising technique. Of the five components, “hierarchical classification”, “optimal denoised image”, and “non-arbitrary seed image” are our original contributions, while the rest have been adapted from their corresponding sources.

3.1 Fractal Image Denoising

Fractal coding exploits the presence of self-similarity in different magnification levels of an image. Fractal image denoising is based on the idea that since natural images contain self-similar structures, they can easily be approximated by contractive affine transformations. However, approximating random noisy components occurring in an image using such affine transforms is impossible [10].

Fractal coding and decoding schemes involve finding the scaling and offset parameters s_j and o_j for a given range R_i and a best matched domain D_j such that Eq. 6 is minimized. In order to store the parameters in the fractal file, they are quantized in the encoder and dequantized in the decoder. Hence post quantization of the parameters often leads to some degree of information loss as compared with pre-quantization [9]. Since noise is high degree of random information, this information is lost due to quantization, which aids the denoising process in fractal coding.

3.2 Hierarchical Classification

FIC is an asymmetric technology. Exhaustive searching of the domain pool to find matching block pairs is computationally intensive, and makes the encoding algorithm slower than most of the existing algorithms. A lot of research has gone into solving the lengthy encoding-time problem, and a popular solution is to perform domain pool classification [19, 26].

Fisher [9] proposed a classification scheme to illustrate the advantages of domain classification. A domain or range is partitioned into four quadrants. For each quadrant, values proportional to mean pixel intensity and to the variance of the pixel intensities are computed. Classes are identified based on the permutational orderings of these values.

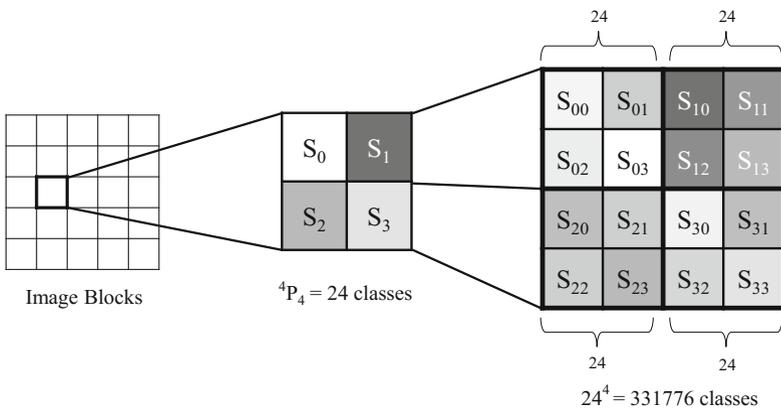


Fig. 2. Hierarchical classification.

The Hierarchical Classification scheme [4] is an improvement of Fisher's Mean-Variance classification scheme. A sub-image is divided into four quadrants (Fig. 2). Sum of pixel values (luminance) of each quadrant is calculated as S_0 , S_1 , S_2 and S_3 . According to the ordering of the luminance sums, there are ${}^4P_4 = 24$ classes. This gives the Level - I classification. In Level - II, each quadrant is further subdivided into 4 sub-quadrants (sixteen subquadrants in total), and sum of pixel values over each sub-quadrant is calculated as S_{i0} , S_{i1} , S_{i2} and S_{i3} ($i = 0, 1, 2, 3$). Based on the ordering of S_{ij} in quadrant i , each quadrant gives 24 classes, totalling to $24^4 = 331776$ classes, as shown in Fig. 2.

Our proposed denoising technique incorporates the hierarchical classification scheme with the baseline FIC to speed up the encoding process in comparison to Baseline FIC.

3.3 Quadtree Partitioning

Some regions (range blocks) in an image can be defined well with domain range blocks, while some others are difficult. *Quadtree Partitioning* breaks up a range block into 4 equally sized sub-quadrants, when it cannot be approximated well enough by a domain block. The process repeats recursively, starting at the initial image and iterating until partitioned blocks are small enough to be approximated within some specified MSE threshold. Small blocks can be approximated better than large blocks because pixels within a small neighbourhood tend to show high correlation (Fig. 3).



Fig. 3. Quadtree partitioning with dynamic range sizes.

In our implementation, we have created domain pools of sizes 8×8 (D_{min}), 16×16 and 32×32 (D_{max}); corresponding range sizes are 4×4 (R_{min}), 8×8 and 16×16 (R_{max}) respectively. The program first tries with a 16×16 range block. If a suitable domain is found, an affine map is generated, else the range is quadtree partitioned into 4 smaller quadrants as range blocks, and domain blocks of appropriate size are searched for each of them. High detail areas are usually mapped by smaller ranges, while less-detailed areas are mapped by larger ones. The distance between two successive domains, called *domain step-size*, is set to 4 pixels in the default case.

3.4 Optimal Denoised Image

In our proposed technique, we obtain an optimal denoised image by averaging and post processing. Image decoding is done by iterating through the fractal code, where we approximate the i_{th} range block R_{apr}^i of k^{th} approximate image from an arbitrary image using

$$R_{apr}^i = s \cdot \bar{D}^{k-1} + o \quad (7)$$

where \bar{D} is obtained by averaging i.e. taking the mean value of every 2×2 pixel block of the $(k-1)^{th}$ image. Since noise is distributed randomly over the pixels, averaging the pixel values also averages the noise levels, resulting in low noise value. As ranges are encoded independently, the block boundaries may not be smooth. The human eye is sensitive to such discontinuities, however small. A post-processing step [9] has been applied to smooth-out the block boundaries and improve decoded image quality. The pixels at the block-boundaries are smoothed by a weighted average technique. The pixel values a and b (on either side of the block-boundary) are replaced by $w_1a + w_2b$ and $w_2a + w_1b$ respectively, with $w_1 + w_2 = 1$. For smallest ranges (4×4), the weights are $w_1 = 5/6$ and $w_2 = 1/6$, while for the larger ranges, the weights are $w_1 = 1/3$ and $w_2 = 2/3$. Though the weights are heuristic, they give satisfactory results (Fig. 7d).

3.5 Non-arbitrary Seed Image

Conventional fractal decoding uses an arbitrary seed image as the starting image of the recursive decoding process. For easy implementation, this image is chosen as a plain black image i.e. a matrix filled with zeroes. Since the first decoding iteration is dependant on the seed image, the plain black image may not always give the best results for denoising purposes. For images having self-similar structures, the final attractor image (fixed point) is not dependent on the choice of seed image. However, a noisy image has lots of random pixel intensity variations, and is therefore not very self similar. So for denoising, we have used some

non-arbitrary noise free images as the starting seed image for fractal decoding (Fig. 4). The idea is to “help” the first decoding iteration by providing noise-free domains to map to ranges. Using these images we gained some small yet significant improvement in decoded image quality (Fig. 5).

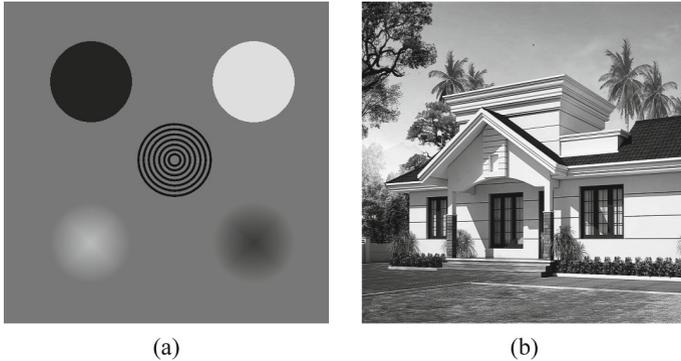


Fig. 4. Non-arbitrary noise-free seed images chosen to further improve denoised output.

4 Experimental Results

We tested our technique on seven benchmark 8-bit images (Aerial, Baboon, Boat, Bridge, Lenna, Man and Peppers) taken from the USC-SIPI Image Database. Image sizes are either 512×512 or 1024×1024 . We have implemented the algorithm in C++ using OpenCV library, and have run the tests in Ubuntu 14.04 running on Intel i7 2630QM 2.0 GHz processor and 4GB DDR3 RAM. We compared the performance of our proposed technique to two popular adaptive image restoration schemes: Lee filter [14] and Bilateral filter [22]. For this comparison, we have run the tests on images – Lenna, Aerial, Baboon and Boat.

For all the images, varying levels of AWGN was incorporated by varying the standard deviation σ from 0.03 to 0.30. A higher σ means more noise. Below $\sigma = 0.03$, the image appeared practically noiseless, while above $\sigma = 0.30$, the image was not recognizable. Then each image was fractally encoded and decoded. The decoded image was compared to the noisy image and the original image.

We have measured the decoded image quality with respect to the original noise free image using Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [24,25].

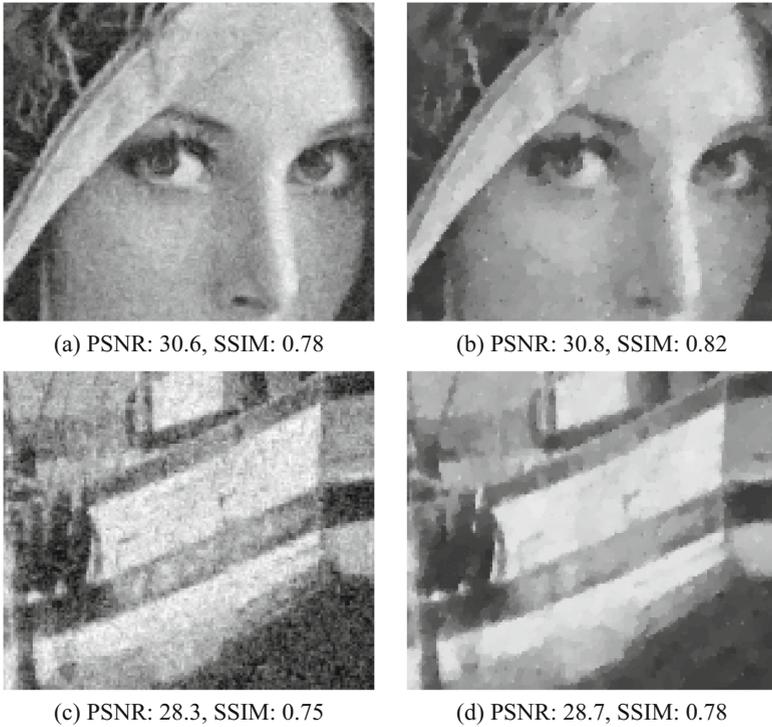


Fig. 5. Output after using (a, c) random seed image and (b, d) Fig. 4a as seed image, with AWGN $\sigma = 0.06$.

Figure 5 shows the output of denoising operation after using a random seed image versus using a noise-free seed image, as discussed in Sect. 3.5. Metric-wise the improvements are small. However to the human eye, object recognition and visual clarity is increased in Figs. 5b and d.

Table 1 lists the denoised image quality (in PSNR and SSIM) obtained using the proposed method, and compares the results with that of Lee filter [14] and Bilateral filter [22]. Figures 6a and d show the comparison in the variation of noisy and denoised image quality with increase in image noise using PSNR and SSIM metrics, respectively, using the proposed denoising scheme. In both graphs, the denoised images have better quality than their noisy counterparts, indicating that denoising was indeed achieved. However, for the 'bridge' image, at very low levels of noise, our proposed scheme actually degraded the image quality.

Figures 6b, c, e and f show the comparison of our scheme with the Lee filter denoising method and the Bilateral filter method, for Aerial and Baboon images. Here again, our proposed scheme does degrade the image quality for very low noise levels. A possible explanation is that the fractal scheme acts more like a lossy compressor than an image restorator at these very low levels of noise. As

Table 1. Some highlights of experimental results obtained, showing denoised image quality (PSNR and SSIM) obtained using proposed denoising method, and comparing with Lee filter [14] and Bilateral filter [22]. Bold figures indicate cases where our proposed scheme performs the best denoising across varying noise values, compared to other methods.

Image	Sigma	Noisy image		Proposed method		Lee filter		Bilateral filter	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lenna	0.03	30.5	0.72	32.42	0.87	32.76	0.86	31.93	0.88
	0.12	18.6	0.22	26.90	0.58	24.03	0.43	20.82	0.26
	0.21	14.1	0.11	22.99	0.39	20.57	0.27	15.08	0.10
	0.30	11.7	0.07	20.51	0.29	18.63	0.20	12.16	0.06
Baboon	0.03	30.5	0.89	22.35	0.61	24.98	0.79	22.97	0.85
	0.12	18.5	0.47	21.88	0.55	21.42	0.53	18.77	0.41
	0.21	14.0	0.28	20.21	0.43	19.15	0.38	14.38	0.21
	0.30	11.6	0.18	18.76	0.34	17.76	0.30	11.74	0.13
Aerial	0.03	30.5	0.84	26.99	0.85	28.67	0.87	29.40	0.87
	0.12	18.7	0.40	24.62	0.67	22.15	0.54	20.48	0.33
	0.21	14.5	0.24	21.96	0.52	18.45	0.39	15.31	0.16
	0.30	12.0	0.16	19.71	0.41	16.32	0.29	12.46	0.11
Man	0.03	30.7	0.77	31.78	0.83	30.76	0.81	30.76	0.81
	0.12	19.0	0.25	22.84	0.41	21.19	0.30	21.19	0.30
	0.21	14.5	0.12	19.44	0.26	15.48	0.13	15.48	0.13
	0.30	12.0	0.08	17.32	0.19	12.44	0.08	12.44	0.08

noise levels gradually increase, our scheme starts performing well as a denoiser. When comparing to Lee filter, our scheme performs well for the baboon image, but not so well for the aerial image. It is interesting to note that Bilateral filtering actually degrades the image (according to the metrics) when noise level increases. An observation from the curves show that the proposed scheme performs best when σ is around 0.05. The downward slope is least around this region of the curve, and for baboon it even shows an upward trend.

Figures 7 and 8 show the actual output images of our proposed scheme compared with the noisy images, the images restored with Lee-filter and the images restored with Bilateral filter. The SSIM and PSNR values of each image, compared to the original, noise-free image is given below each image. Lee filter makes images look blurry. Bilateral filter makes the image look sharper, but noise particles are not effectively removed. Our scheme maintains the contrast and edges.

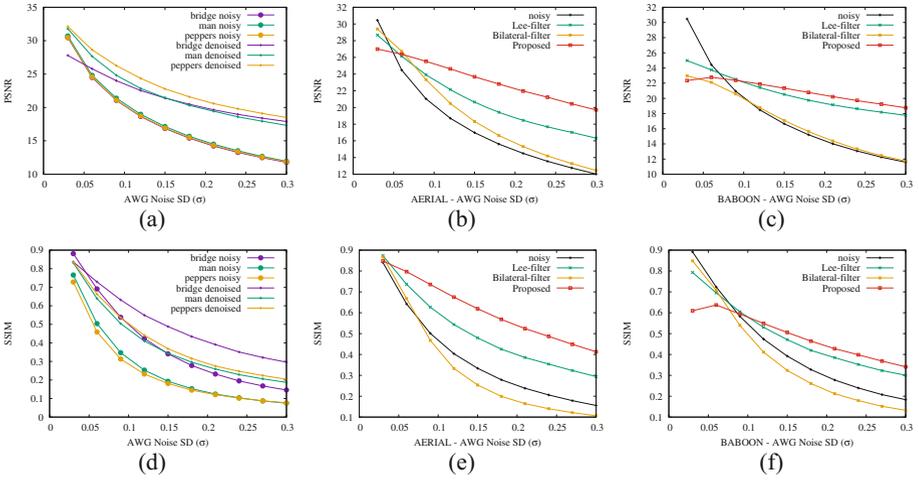


Fig. 6. (a) Comparison of level of restoration for different images using PSNR metric. (b) - (c) Comparison of performance of proposed method with Lee filter [14] and Bilateral filter [22] for Aerial and Baboon images using PSNR. (d) Comparison of level of restoration for different images using SSIM metric. (e) - (f) Comparison of performance of proposed method with Lee filter [14] and Bilateral filter [22] for Aerial and Baboon images using SSIM.

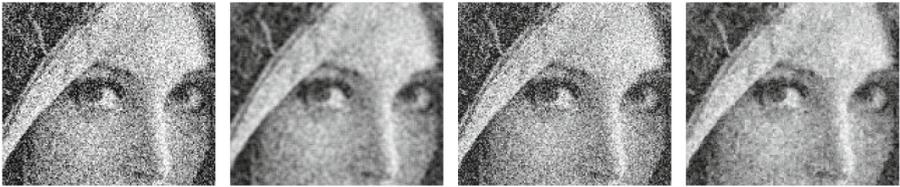
Size of range blocks chosen also influence the degree of denoising achieved. If only large range block sizes are chosen (like 32×32 or 64×64), then all noisy components disappear, but the quality of the output image is lowered as well. On the other hand, if only very small range blocks (like 2×2 or 4×4) are chosen, all the details from the noisy image will get approximated accurately, including the noisy components, thereby bringing back the noise in the decoded image, and leading to very little or no denoising. This illustrates the importance of properly choosing the fractal encoding parameters for reducing noise. Our proposed scheme uses the quadtree partitioning scheme (Sect. 3.3), instead of using fixed size range blocks to tackle this situation. If the tolerance fidelity for a range block is exceeded, then it is partitioned into quadrants, and each quadrant is processed in a similar fashion as the parent block.



(a) $\sigma = 0.06$ (PSNR: 24.4, SSIM: 0.44) (b) Lee-filter (PSNR: 28.6, SSIM: 0.65) (c) Bilateral filter (PSNR: 29.3, SSIM: 0.66) (d) Proposed (PSNR: 30.6, SSIM: 0.78)



(e) $\sigma = 0.12$ (PSNR: 18.6, SSIM: 0.22) (f) Lee-filter (PSNR: 24.0, SSIM: 0.43) (g) Bilateral filter (PSNR: 20.8, SSIM: 0.26) (h) Proposed (PSNR: 26.9, SSIM: 0.58)



(i) $\sigma = 0.18$ (PSNR: 15.3, SSIM: 0.13) (j) Lee-filter (PSNR: 21.5, SSIM: 0.31) (k) Bilateral filter (PSNR: 16.5, SSIM: 0.13) (l) Proposed (PSNR: 24.1, SSIM: 0.44)



(m) $\sigma = 0.24$ (PSNR: 13.2, SSIM: 0.09) (n) Lee-filter (PSNR: 19.8, SSIM: 0.24) (o) Bilateral filter (PSNR: 13.9, SSIM: 0.08) (p) Proposed (PSNR: 22.1, SSIM: 0.35)

Fig. 7. Output for Lena image. First column is the noisy image. Second column uses Lee filter [14]. Third column uses Bilateral Filter [22]. Fourth column has denoised image using proposed Fractal scheme.



(a) $\sigma = 0.06$ (PSNR: 24.5, SSIM: 0.53)



(b) Lee-filter (PSNR: 27.6, SSIM: 0.67)



(c) Bilateral filter (PSNR: 27.7, SSIM: 0.66)



(d) Proposed (PSNR: 28.3, SSIM: 0.75)



(e) $\sigma = 0.12$ (PSNR: 18.6, SSIM: 0.28)



(f) Lee-filter (PSNR: 23.6, SSIM: 0.46)



(g) Bilateral filter (PSNR: 20.7, SSIM: 0.29)



(h) Proposed (PSNR: 25.7, SSIM: 0.59)



(i) $\sigma = 0.18$ (PSNR: 15.2, SSIM: 0.18)



(j) Lee-filter (PSNR: 21.1, SSIM: 0.34)



(k) Bilateral filter (PSNR: 16.5, SSIM: 0.16)



(l) Proposed (PSNR: 23.4, SSIM: 0.46)



(m) $\sigma = 0.24$ (PSNR: 13.1, SSIM: 0.12)



(n) Lee-filter (PSNR: 19.6, SSIM: 0.27)



(o) Bilateral filter (PSNR: 13.8, SSIM: 0.10)



(p) Proposed (PSNR: 21.6, SSIM: 0.37)



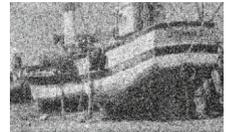
(q) $\sigma = 0.30$ (PSNR: 11.6, SSIM: 0.09)



(r) Lee-filter (PSNR: 18.4, SSIM: 0.23)



(s) Bilateral filter (PSNR: 12.1, SSIM: 0.07)



(t) Proposed (PSNR: 20.1, SSIM: 0.31)

Fig. 8. Output for Boat image. First column is the noisy image. Second column uses Lee filter [14]. Third column uses Bilateral Filter [22]. Fourth column has denoised image using proposed Fractal scheme.

5 Conclusions

Our proposed fractal based denoising scheme has an advantage: as fractal transformations were primarily developed for image compression, our proposed scheme can also be optimized for performing image denoising and image compression simultaneously. Our scheme can remove noise effectively and also preserving edges, in order to reduce visual artifacts and distortions. The denoising is performed mainly during the fractal encoding process, where hierarchical classification has been employed to accelerate the encoder. Quadtree based image partitioning, pyramidal decoding and post processing has been used to enhance the degree of noise removal and improve the output image quality.

References

1. Banach, S.: Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fund. Math* **3**(1), 133–181 (1922)
2. Barnsley, M.F.: *Fractals Everywhere*. Academic Press, Cambridge (1988)
3. Barnsley, M.F., Jacquin, A.E.: Application of recurrent iterated function systems to images. In: *Visual Communications and Image Processing 1988*. Third in a Series, pp. 122–131. International Society for Optics and Photonics (1988)
4. Bhattacharya, N., Roy, S.K., Nandi, U., Banerjee, S.: Fractal image compression using hierarchical classification of sub-images. In: *Proceedings of the 10th International Conference on Computer Vision Theory and Applications*, pp. 46–53 (2015)
5. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.* **4**(2), 490–530 (2005)
6. Chaudhuri, B.B., Sarkar, N.: Texture segmentation using fractal dimension. *IEEE Trans. Pattern Anal. Mach. Intell.* **17**(1), 72–77 (1995)
7. Duh, D.J., Jeng, J., Chen, S.Y.: DCT based simple classification scheme for fractal image compression. *Image Vis. Comput.* **23**(13), 1115–1121 (2005)
8. Eskicioglu, A.M., Fisher, P.S.: Image quality measures and their performance. *IEEE Trans. Commun.* **43**(12), 2959–2965 (1995)
9. Fisher, Y. (ed.): *Fractal Image Compression: Theory Appl.* Springer, New York (1994)
10. Ghazel, M., Freeman, G.H., Vrscaj, E.R.: Fractal image denoising. *IEEE Trans. Image Process.* **12**(12), 1560–1578 (2003)
11. Hürtgen, B., Stiller, C.: Fast hierarchical codebook search for fractal coding of still images. In: *Berlin-DL tentative*, pp. 397–408. International Society for Optics and Photonics (1993)
12. Hutchinson, J.E.: *Fractals and self similarity*. University of Melbourne. [Department of Mathematics] (1979)
13. Jacquin, A.E.: Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Trans. Image Process.* **1**(1), 18–30 (1992)
14. Lee, J.S.: Digital image enhancement and noise filtering by use of local statistics. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI* **2**(2), 165–168 (1980)
15. Lian, S.: Image authentication based on fractal features. *Fractals* **16**(04), 287–297 (2008)
16. Lin, K.T., Yeh, S.L.: Encrypting image by assembling the fractal-image addition method and the binary encoding method. *Opt. Commun.* **285**(9), 2335–2342 (2012)

17. Pi, M., Mandal, M.K., Basu, A.: Image retrieval based on histogram of fractal parameters. *IEEE Trans. Multimedia* **7**(4), 597–605 (2005)
18. Roy, S.K., Kumar, S., Chanda, B., Chaudhuri, B.B., Banerjee, S.: Fractal image compression using upper bound on scaling parameter. *Chaos Solitons Fractals* **106**, 16–22 (2018)
19. Saupe, D., Hamzaoui, R.: A review of the fractal image compression literature. *ACM SIGGRAPH Comput. Graph.* **28**(4), 268–276 (1994)
20. Shao, L., Yan, R., Li, X., Liu, Y.: From heuristic optimization to dictionary learning: a review and comprehensive comparison of image denoising algorithms. *IEEE Trans. Cybern.* **44**(7), 1001–1013 (2014)
21. Tang, X., Qu, C.: Facial image recognition based on fractal image encoding. *Bell Labs Techn. J.* **15**(1), 209–214 (2010)
22. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: *Sixth International Conference on Computer Vision*, pp. 839–846. IEEE (1998)
23. Wang, J., Zheng, N.: A novel fractal image compression scheme with block classification and sorting based on Pearson’s correlation coefficient. *IEEE Trans. Image Process.* **22**, 3690–3702 (2013)
24. Wang, Z., Bovik, A.C.: Mean squared error: love it or leave it? A new look at signal fidelity measures. *IEEE Sig. Process. Mag.* **26**(1), 98–117 (2009)
25. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
26. Wohlberg, B., De Jager, G.: A review of the fractal image coding literature. *IEEE Trans. Image Process* **8**(12), 1716–1729 (1999)