SELF-SIMILARITY IN IMAGING, 20 YEARS AFTER “FRACTALS EVERYWHERE”

Mehran Ebrahimi and Edward R.Vrscay

Department of Applied Mathematics
Faculty of Mathematics, University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
m2ebrahi@uwaterloo.ca, ervrscay@uwaterloo.ca

ABSTRACT

Self-similarity has played an important role in mathematics and some areas of science, most notably physics. We briefly examine how it has become an important idea in imaging, with reference to fractal image coding and, more recently, other methods that rely on the local self-similarity of images. These include the more recent nonlocal methods of image processing.

1. INTRODUCTION

Since the appearance of B. Mandelbrot’s classic work, *The Fractal Geometry of Nature* [28], the idea of self-similarity has played a very important role in mathematics and physics. In the late-1980’s, M. Barnsley of Georgia Tech, with coworkers and students, showed that sets of contractive maps with associated probabilities, called “Iterated Function Systems” (IFS), could be used not only to generate fractal sets and measures but also to approximate natural objects and images. This gave birth to *fractal image compression*, which would become a hotbed of research activity over the next decade. Indeed, twenty years have passed since *Fractals Everywhere* [4], Barnsley’s beautiful exposition of IFS theory and its applications, was first published.

Historically, most fractal image coding research focussed on its compression capabilities, i.e., obtaining the best possible accuracy with the smallest possible domain pool. As a result, these investigations would rarely venture beyond observing what “optimal” domain blocks could provide. By taking a step back, however, and examining the statistics of how well image subblocks are approximated by other subblocks, at either the same scale or different scales, one sees that natural images are generally quite self-similar. This actually explains why fractal image coding – a nonlocal image processing method – “works” (with the acknowledgment that it no longer furnishes a competitive method of image compression).

Furthermore, these observations have led to the formulation of a general model of local affine image self similarity [2] that includes, as special cases, cross-scale fractal coding and same-scale nonlocal means denoising. In other words, a number of nonlocal image processing methods may be viewed under a common theme of self-similarity.

In this paper, we outline the evolution of the idea of self-similarity from Mandelbrot’s “generators” to Iterated Function Systems and fractal image coding. We then briefly examine the use of fractal-based methods for tasks other than image compression, e.g., denoising and superresolution. Finally, we examine some extensions and variations of nonlocal image processing that are inspired by fractal-based coding.

2. FROM SELF-SIMILARITY TO FRACTAL IMAGE CODING

2.1. Some history

In *The Fractal Geometry of Nature*, B. Mandelbrot showed how “fractal” sets could be viewed as limits of iterative schemes involving *generators*. In the simplest case, a generator G acts on a set S (for example, a line segment) to produce N finely-contracted copies of S and then arranges these copies in space according to a prescribed rule. Starting with an appropriate “initiator set” S_0, the iteration procedure $S_{n+1} = G(S_n)$ converges, in the limit $n \to \infty$, to a fractal set S.

Last, but certainly not least, the set S is “self-similar,” meaning that arbitrarily small pieces of S are scaled-down copies of S. Consequently, S can be expressed as a union of contracted copies of itself.

Some of Mandelbrot’s examples were certainly not new. For example, in the classical construction of the ternary Cantor set, the usual “middle-thirds” dissection procedure is represented by a generator G that, acting on a line segment of length l, produces two contracted copies of length $l/3$ which are separated by a distance $l/3$. From this, the Cantor set is viewed as a union of two contracted copies of itself. (And, in turn, we obtain its fractal dimension D to be $\ln 2/\ln 3$.) These simple examples, however, led to more general constructions, including random fractals. The result was a rather unified treatment of fractal geometry.

The next leap in fractal analysis and construction came with the work of J. Hutchinson [24]. One of the main ingredients was a set of N contractive maps $w_i : X \to X$ on a complete metric space (X, d). Associated with these maps was a set-valued mapping $\mathbf{w}$, defined as the union of the set-valued actions of the w_i. (As such, $\mathbf{w}$ performed the shrink-and-placement procedure of Mandelbrot's generator G.) Hutchinson showed that $\mathbf{w}$ is contractive in $(\mathcal{H}(X), h)$, the complete metric space of non-empty compact subsets of X with Hausdorff metric h. From Banach's theorem, there exists a unique set $A \in \mathcal{H}(X)$, the *attractor* of the IFS $\mathbf{w}$, which satisfies the fixed point relation

$$A = \mathbf{w}(A) = \bigcup_{n=1}^{N} w_i(A). \tag{1}$$

A is self-similar since it can be expressed as a union of copies of itself.

Hutchinson also considered a set of probabilities p_i associated with the spatial maps w_i, $1 \leq i \leq N$. These are used to define an operator M on the space $\mathcal{M}(X)$ of Borel probability measures on X: Very briefly, the action of M on a measure μ is to produce N spatially-contracted copies of μ (via the w_i) which are then weighted according to the respective probabilities p_i. The operator M is contractive in $(\mathcal{M}(X), d_M)$, where d_M denotes the Monge-Kantorovich metric. Therefore there exists a unique measure $\bar{\mu} \in \mathcal{M}(X)$ satisfying the fixed point relation

$$\bar{\mu}(S) = M\bar{\mu}(S) = \sum_{i=1}^{N} p_i \bar{\mu}(w_i^{-1}(S), \quad \forall S \in \mathcal{H}(X). \tag{2}$$

Moreover, the support of $\bar{\mu}$ is the IFS attractor A. The *invariant measure* μ satisfies a more generalized self-similarity or *self-tiling* property.

M. Barnsley and S. Demko [5] independently discovered the use of such systems of mappings and associated probabilities for the construction of fractal sets and measures, coining the term "Iterated Function Systems" (IFS). Their analysis was framed in a more probabilistic setting and gave rise to the well-known *chaos game algorithm* for generating pictures of attractors and invariant measures.

Even more significant is the following: In the Barnsley/Demko paper was the first suggestion that IFS might be useful for the *approximation of natural objects*. This was the seed for the *inverse problem of fractal-based approximation*: Given a "target" set S, for example, a leaf, can one find an IFS with attractor A that approximates S to some desired degree of accuracy?

Subsequently, Barnsley and students [6] showed how the inverse problem could be reformulated/simplified in terms of the following "Collage Theorem," a simple consequence of Banach's fixed point theorem: Given a contraction map $T : X \to X$ with contraction factor $c \in [0, 1)$ and fixed point $\bar{x}$, then

$$d(x, \bar{x}) \leq \frac{1}{1-c} d(x, Tx), \quad \forall x \in X. \tag{3}$$

If $x \in X$ is a "target" (set or measure) that we wish to approximate with the fixed point $\bar{x}$ of a contraction map T, then instead of searching for a T (i.e., an IFS) that makes the approximation error $d(x, \bar{x})$ small, one looks for a T such that the *collage distance* $d(x, Tx)$ is small – in other words a transform T that maps x as close as possible to itself. For IFS on sets, cf. Eq. (1), this amounts to looking for ways in which a set S can be approximated by a union of contracted and distorted copies of itself. This procedure was illustrated very nicely for a generic leaf-shape in [6].

In [6] was also presented the infamous "Barnsley fern" – the attractor of a four-map IFS with probabilities (IFSP). But this "fern" was more than an IFS attractor set A – it was an invariant measure μ which could be represented on a computer screen as a shaded image. In other words, the construction of *sets* by IFS now becomes the construction of *images* by IFSP.

Naturally, the next question was: "Can IFS(P) be used to approximate other natural objects?" But an even more ambitious question was: "Can IFS(P) be used to approximate images?" There were various attempts (which will not be reviewed here, for lack of space), and Barnsley himself announced success to the world in the January 1988 issue of *BYTE* magazine [3], claiming that astronomical rates of data compression could be achieved with IFS image coding. However, the details of the IFS compression method were not revealed at that time for proprietary reasons. (Barnsley had been granted a software patent and subsequently founded Iterated Systems Incorporated.)

The thesis of A. Jacquin [25], one of Barnsley's Ph.D. students, however, removed much of the mystery behind IFS image coding. Indeed, Jacquin's seminal paper [26] described in sufficient detail the method of *block-based fractal image coding* which is still the basis of most, if not all, fractal-based image coding methods. It is, of course, overly ambitious to expect that an image can be well approximated with smaller parts of itself. Therefore, in the spirit of standard block-based image coding methods (e.g., DCT, VQ), *range* or *child* subblocks of an image were approximated by affine greyscale transformations of decimated versions of larger *domain* or *parent* subblocks. Although Jacquin's original formulation was in terms of measures, the method is easily interpreted in terms of image functions.

Jacquin's original paper launched an intensive activity in fractal image compression. One of the main drawbacks of fractal coding is the time required to search for optimal domain blocks. As such, there was much investigation on how to achieve the best quality with as little searching as possible. A discussion of some of these methods can be found in [19, 20, 27].

2.2. A closer look at block-based fractal image coding

Here we outline the most important features of fractal image coding. Interested readers will find more detailed treatments in [7, 19, 20, 27].

For simplicity, the support X of an image function u will be considered as $n \times n$ pixel array, i.e., a discrete support. Now consider a partition of X into nonoverlapping subblocks R_i, $1 \leq i \leq N$, so that $X = \cup_i R_i$. Associated with *range block* R_i is a larger *domain block* $D_i \subset X$ so that $R_i = w_i(D_i)$, where w_i is a 1-1 contraction map. (In the pixel case, it will represent the decimation needed to accomplish the spatial contraction.) Assume that the image function $u(R_i) = u|_{R_i}$ supported on each subblock R_i is well approximated by a spatially-contracted and greyscale-modified copy of $u(D_i) = u|_{D_i}$:

$$u(R_i) \approx \phi_i(u(w_i^{-1}(D_i)), \tag{4}$$

where the $\phi_i : \mathbf{R} \to \mathbf{R}$ are greyscale maps that are usually affine in form, i.e., $\phi_i(t) = \alpha_i t + \beta_i$. Because the range blocks R_i are nonoverlapping, we can write Eq. (4) as

$$u(x) \approx (Tu)(x) = \alpha_i(u(w_i^{-1}(x))) + \beta_i, \quad x \in R_i, \tag{5}$$

where T is the *fractal transform operator* defined by the range-domain assignments and the affine greyscale maps ϕ_i. Under suitable conditions on the α_i greyscale coefficients and the $\mathbf{R}^2$ contraction factors of the w_i, the operator T is contractive in an appropriate image function metric space $\mathcal{F}(X)$ (typically $L^2(X)$). As such, there will exist a unique fixed point $\bar{u} \in \mathcal{F}(X)$ such that $T\bar{u} = \bar{u}$.

It would seem natural that for each range block R_i, we choose the domain block $D_{j(i)}$ that best approximates it, i.e., that yields the lowest approximation error Δ_{ij} as defined below:

$$\Delta_{ij} = \min_{\alpha,\beta \in \Pi} \|u(R_i) - \alpha u(D_j) - \beta\|. \tag{6}$$

Here, $\| \cdot \|$ denotes the $L^2(X)$ norm. $\Pi \subset \mathbf{R}^2$ denotes the feasible (α, β) parameter space, which will be suitably restricted for a given application.

This is, in fact, equivalent to minimizing the collage error $d(u, Tu) = \|u - Tu\|$ in Eq. (3). As such, we expect that the fixed point $\bar{u}$ will be a reasonable approximation to the target image u. Moreover, $\bar{u}$ can be generated by simple iteration: Starting with any seed u_0 (e.g. a blank screen), the iteration procedure $u_{n+1} = Tu_n$ is guaranteed to converge to $\bar{u}$. (In fact, for the discrete, i.e., pixellated, case, convergence is achieved in a finite number of steps.) Note that the fractal transform is an example of a *nonlocal transform* – a portion of the image function u is approximated by another portion that, in general, lies elsewhere.

In the early days of fractal image compression, most people used affine greyscale maps because they seemed to be the best "compromise" in terms of rate-distortion – reasonably good fitting with only two parameters. As well, the coefficients are easy to compute. A few people also looked at higher order polynomials. Perhaps the most significant work in this vein was done by D. Monro and students/coworkers at the University of Bath [30, 31]. They examined higher order polynomials in t, the greyscale variable and also, more significantly, "place-dependent" greyscale maps, e.g., $\phi(x, y, t) = at + b + cx + dy$. They claimed that the computational cost of storing the additional parameters c and d in their "Bath Fractal Transform" was offset by not having to search for parent blocks. For example, in their first implementations of the BFT, domain blocks were simply larger blocks co-centric with their respective range blocks.

That being said, we believe that affine maps capture visual similarities reasonably well. Higher-degree polynomials will provide better approximations but probably at the expense of artificial visual similarities.

2.3. Fractals beyond compression

More recently, the ability of fractal coding to solve problems other than compression has investigated. In [22, 23], it was first observed that fractal-based methods have denoising capabilities. There are two principal reasons: (i) the affine greyscale fitting between domain and range blocks causes some variance reduction in the noise and (ii) the spatial contraction/pixel decimation involved in mapping domain blocks to range blocks provides further variance reduction. Additional denoising can be achieved by using estimates of the noise variance to estimate the fractal code of the noiseless image [22]. And the fact that each range block is well approximated by a number of domain blocks – an important point that will be discussed further below – can be exploited to perform denoising by using multiple parents [1].

Due to the resolution-independent nature of the fractal transform, interpolation algorithms called "fractal zoom" have been also been developed in the literature [19, 20, 21, 27, 32]. Very recently, we have shown that such *super-resolution* can be accomplished in the frequency domain using IFS-type methods [29].

3. NEWER TRENDS IN SELF-SIMILARITY

3.1. Statistical studies on image self-similarity

Recent investigations [1, 2] have shown that images generally possess a great deal of local affine self-similarity: Image subblocks are well approximated by a number of other blocks, at the same or different scales, when affine greyscale transformations are employed (and to a lesser degree when only the identity greyscale transformation is employed). This has led to the introduction of a simple model of affine image self-similarity [2] which includes the method of fractal image coding (cross-scale, affine greyscale similarity) and the nonlocal means denoising method [9, 10] (same-scale, translational similarity) as special cases.

In [2] was presented numerical evidence that the distributions of domain-range approximation errors Δ_{ij} defined in Eq. (6) can be used to characterize the relative degree of self-similarity of images. For many natural images, the distributions of errors Δ_{ij} demonstrate significant peaking near zero error. For others, the distributions are broader and peak farther away from zero. The latter images are judged to be less self-similar than the former. As noise of increasing variance is added to an image, the distributions broaden and move outward as well. In [15], we have also examined the distribution of neighborhood distances at various scales for various natural images. Figure 1 shows some of the results. It is observed that for each test image, the similarity between the same-scale and cross-scale histograms is striking. The details in producing the histograms are discussed in [15].

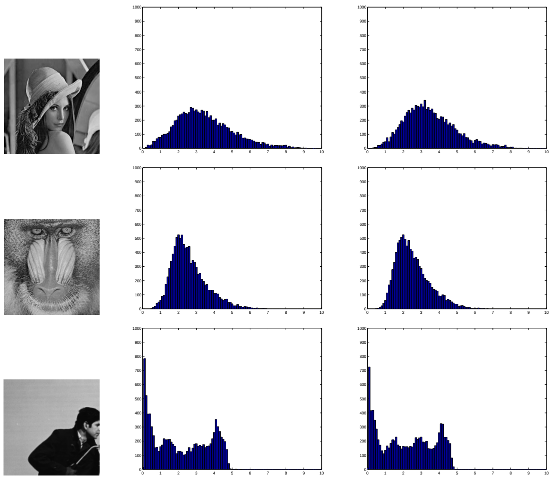


Figure 1. Histograms of approximation errors Δ_{ij} for same-scale (middle) and cross-scale (right) cases for various images [15].

3.2. Including *a priori* information in the fractal code

Fractal imaging methods have relied almost exclusively on the Collage Theorem for coding and on Banach's contraction mapping theorem for decoding, i.e., to generate the attractor of the fractal transform. This can be viewed as a major drawback for the traditional fractal-based methods in that the output is quite restricted – no prior knowledge or additional regularization can be combined with these methods. One may well ask whether it is absolutely necessary to employ Banach's contraction mapping principle.

Recently, we have examined various possibilities of including *a priori* information with the fractal code [11, 12]. In [12] an algebraic formulation is presented, in which prior information may be incorporated in terms of penalty functions. The problem is then solved by minimization.

In [11], it is shown that fractal image coding can be viewed and generalized in terms of the method of projections onto convex sets (POCS). In the approach, the fractal code is considered as a set of spatial domain similarity constraints. As a result, POCS provides an opportunity to apply the fractal code of an image along with additional constraints at the decoding stage.

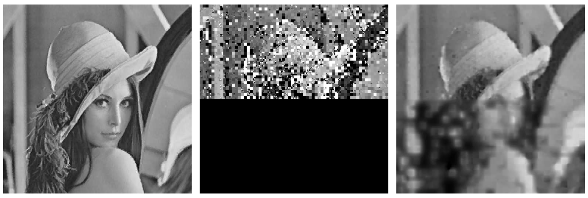


Figure 2. *Lena* attractor 4x4 range block size, decoded in the case that the bottom half of the fractal code is missing. Seed image is black. POCS scheme [11]

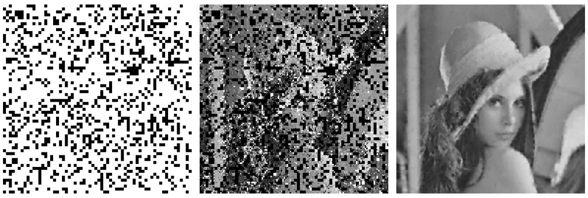


Figure 3. A quarter of the code related to the range blocks in black is missing, Decoded *Lena* attractor image starting from black seed, where one quarter of the fractal code is randomly missing. POCS scheme [11]

As an illustration, assume that we have an incomplete fractal code of an image, i.e., some range-domain block assignments along with their corresponding greyscale coefficients are missing. In this case, the usual fractal decoding scheme employing an arbitrary "seed" image will collapse since the range blocks of the image for which the fractal code is missing cannot be modified. These blocks will simply remain stationary, identical to the corresponding subblocks of the seed image. This is demonstrated in the middle images of Figures 2 and 3, which are the results of normal fractal decoding. In Figure 2, the fractal codes corresponding to all range blocks in the bottom half of an image are missing. In Figure 3, the fractal codes corresponding to some randomly selected range blocks are missing. Such a missing-code problem is, in fact, underdetermined. As such, we consider an additional low pass-filtering operation which corresponds to a smoothness condition. When this condition is used along with the POCS model of fractal coding, we obtain the images on the right side of Figures 2 and 3. The results represent a significant improvement over the middle images of Figures 2 and 3, respectively.

3.3. Non-local means image and film denoising

A new and important denoising method which involves self-similarity is nonlocal-means (NL-means) image denoising [9, 10]. The authors have demonstrated that this algorithm has the ability to outperform classical denoising methods, including Gaussian smoothing, Wiener filter, TV filter, wavelet thresholding, and anisotropic diffusion. Furthermore, an extension of the algorithm has been developed [8] to address the problem of image denoising of image sequences.

As mentioned ealier, the denoising properties of the fractal transform operator have been analyzed in [22, 23, 1]. Here we mention some of the main differences between fractal denoising and NL-means denoising:

- The blocks in fractal-based methods are typically taken at two different scales (coarse and fine), while the NL-means algorithm introduced in [9, 10] considers only same-scale similarity.

- In the NL-means algorithm, blocks are compared solely in terms of the L^2 distance between them. In fractal coding, an additional affine greyscale transformation is used.

- The NL-means is not an iterative scheme. One application of the operator completes the denoising task. However, the fractal denoising scheme [22] is iterative. In [13] it is shown how one may construct a contractive operator along with an iterative scheme for NL-means denoising. However, the iteration procedure is not computationally feasible since the matrix representation of the NL-means operator is not sparse, unlike the case for fractal transforms.

- The NL-means scheme involves the replacement of one pixel at a time. This produces a kind of continuity across pixels, in contrast to standard fractal decoding, which generally suffers from blockiness across range block boundaries, since entire range blocks are being replaced.
- In the NL-means scheme, each pixel is replaced by a weighted average of all pixel intensities in the image, where blocks of greater similarity have larger weights. Such a counterpart exists in fractal image coding in the form of multiparent transforms [1], but has not been extensively employed.

3.3.1. *Same-scale vs. cross-scale approximations*

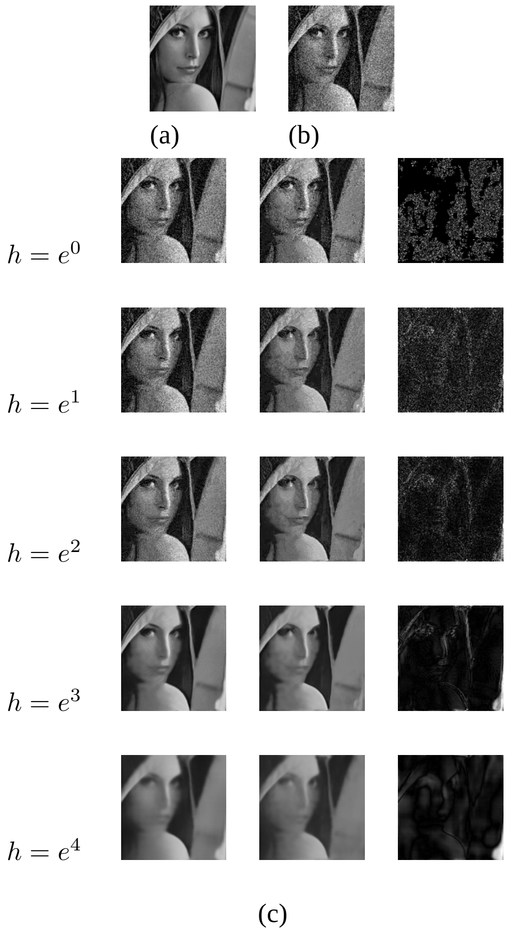


Figure 4. (a) The original image, (b) A noisy observation, In (c): For each value of h are shown: Traditional NL-means (left), NL-means using cross-scale approximations (middle), their difference (right). [15]

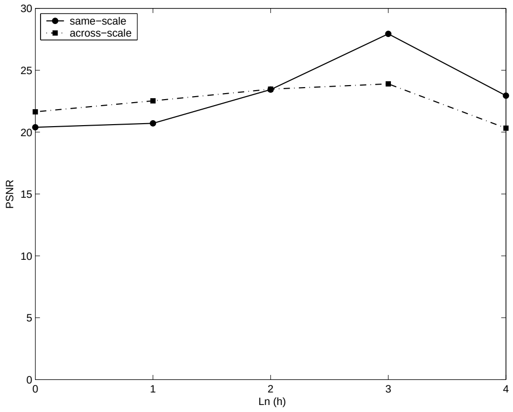


Figure 5. PSNR vs $\ln\ h$ for images plotted on left and in middle of Figure 4(c). [15]

As mentioned above in the first item above, a fundamental difference between NL-means and fractal coding is same-scale vs. cross-scale similarity exploitation. The statistical experiments performed in [15] hint that an appropriately defined cross-scale version of the NL-means filter should behave in a similar fashion to the traditional same-scale NL-means filter. Figure 4 demonstrates a result of a denoising experiment reported in [15] using the same-scale and cross-scale counter-part of the NL-means algorithm for different values smoothing parameter h [See [9, 10] for the definition of h]. Figure 5 compares the PSNR of the output for these cases.

3.4. Non-local means image and video zooming

3.4.1. *Single frame image zooming*

In [14] is presented a single-frame image zooming technique based on so-called “self-examples”. The method combines the ideas of fractal-based image zooming and nonlocal-means image denoising in a consistent framework. In Figure 6, we present some results of this algorithm. It can be seen that the algorithm performs denoising in parallel with zooming. For purposes of comparison, the results of pixel replication and bilinear interpolation are also presented.

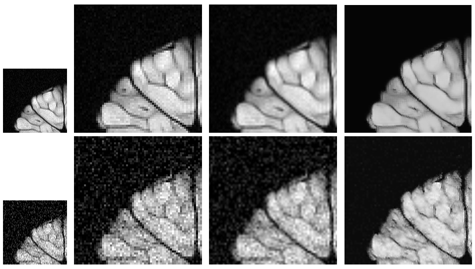


Figure 6. Original, Pixel replication, Bilinear, Self-examples. For both rows $h = 37.5$ is applied. In the second row the standard deviation of noise is $\sigma = 25$. [14]

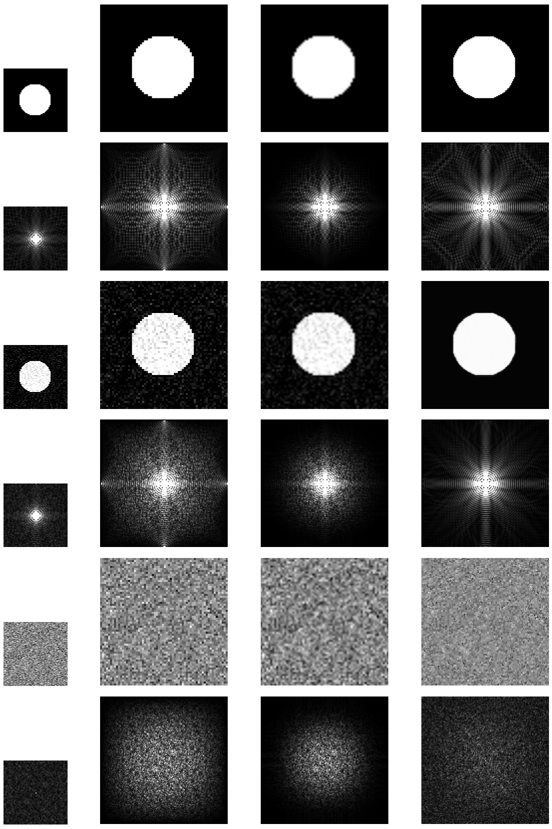
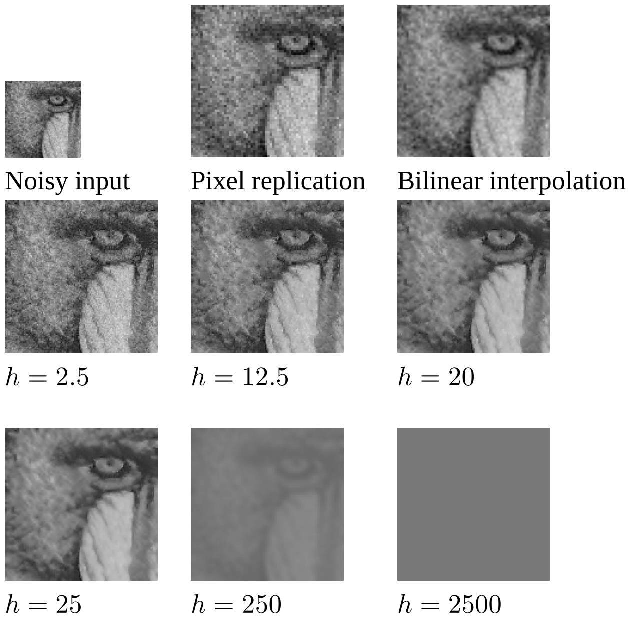


Figure 7. A comparison of the image zooming using self-examples. Odd rows: Original, Pixel replication, Bilinear, Self-examples. Even rows: The corresponding Fourier spectra of the corresponding image above image. [14]



Noisy input

Pixel replication

Bilinear interpolation

$h = 2.5$

$h = 12.5$

$h = 20$

$h = 25$

$h = 250$

$h = 2500$

Figure 8. A comparison of the results obtained by image zooming using self-examples for different values of the filter parameter h. A noisy input image is considered for which the standard deviation of noise is $\sigma = 12.5$. [14]

In Figure 7, rows 1 and 3, are shown the results of this algorithm when applied to a simple circular shape in both noiseless and noisy situations, respectively. Below each of these rows are displayed the Fourier spectra of each image. In the bottom two rows of this figure are shown the effect of this algorithm on a Gaussian white noise sample, along with corresponding Fourier spectra. The value of the filter parameter h was set in a way that the variance of noise was three-quarters of the variance of the input noise. However, the output is still noise-like, as opposed to the results of bilinear and pixel replication. It can be seen that the Fourier spectrum of the output is also distributed rather uniformly over the frequency domain, as opposed to the pixel replication and bilinear interpolation cases.

In Figure 8 are presented the results of this algorithm as applied to a noisy input image, with various values of the filter parameter h. It can be seen that small values of h produce noisy outputs. As h increases, the output is increasingly smoothed and will approach a constant as $h \to \infty$.

3.4.2. *Multiframe super-resolution*

Super-resolution (see e.g., [17] for an introduction) is another challenging inverse problem which has received much interest over the last few decades. In [16], a novel super-resolution scheme is introduced to increase the resolution of multi-frame image sequences. The method is closely associated with the NL-means [8]. In the proposed algorithm, no explicit motion estimation is performed, unlike the case in many other methods. Furthermore, the results are comparable, if not superior, to many existing approaches (e.g., [18]), especially in the case of low signal-to noise ratio. The computational burden of the scheme is a formidable challenge, however, which precludes any iteration scheme to improve the results.

Figure 9 shows some results obtained when this method is applied to an image sequence taken from the data-set library of MDSP at the University of California Santa Cruz.

We have taken the first 20 frames, of size 32×32 from this sequence, and have added independent additive white Gaussian noise of standard deviation $\sigma = 12.5$ to the data set. The top image in Figure 9 is obtained when nearest neighborhood interpolation is applied to the second frame of this sequence, with a zooming factor of 3. The middle image of Figure 9 is the result of bilinear interpolation applied to the same frame. The bottom image of Figure 9 shows result of our algorithm. Details of the algorithm and parameters used can be found in [16].

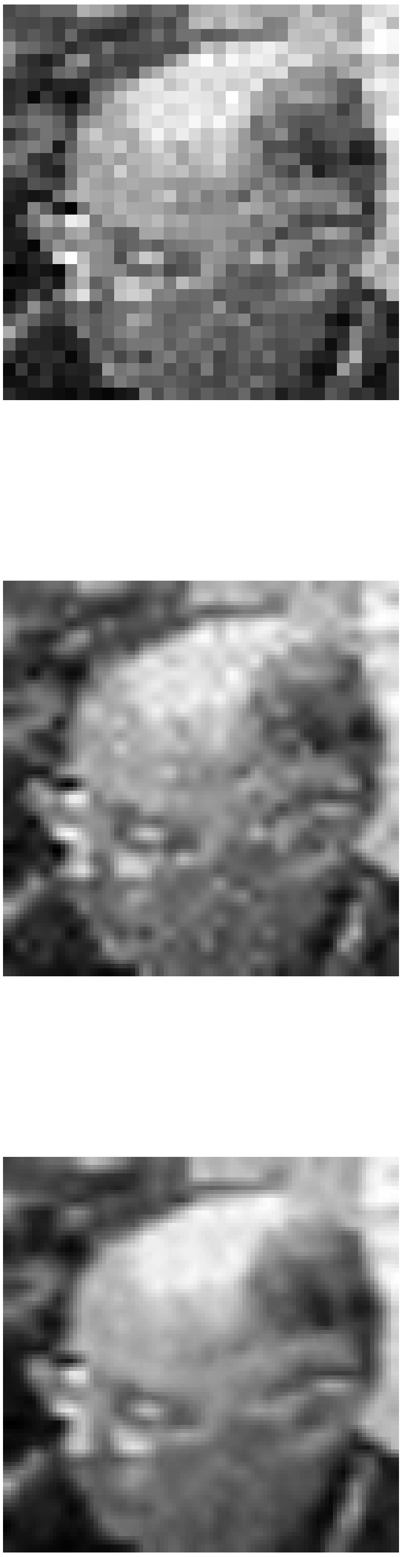


Figure 9. From top to bottom, Nearest neighbohood, Bilinear interpolation, and Proposed multi-frame super-resolution in [16]

Concluding Remarks

In this paper, we have briefly examined the implications of self-similarity in various imaging problems. One goal was to examine the relationships between traditional fractal-based methods and some more recent imaging inverse problems which employ self-similarity properties as a regularizer. Because of space limitations, many details were skipped – the interested reader can consult the references for more information.

Acknowledgements

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada in the form of a Discovery Grant (ERV). ME also gratefully acknowledges financial support from the Province of Ontario (Ontario Graduate Scholarship) as well as from the Faculty of Mathematics, University of Waterloo.

4. REFERENCES

[1] S. K. Alexander. *Multiscale methods in image modelling and image processing*. PhD thesis, Dept. of Applied Mathematics, University of Waterloo, 2005.

[2] S. K. Alexander, E. R. Vrscay, and S. Tsurumi. A simple, general model for the affine self-similarity of images. In *International Conference on Image Analysis and Recognition, ICIAR 2008, Lecture Notes in Computer Science*, volume 5112, pages 192–203, Berlin-Heidelberg, 2008. Springer.

[3] M. Barnsley and A. Sloan. A better way to compress images. *BYTE Magazine*, pages 215–223, January 1988.

[4] M. F. Barnsley. *Fractals Everywhere*. Academic Press, New York, 1988.

[5] M. F. Barnsley and S. Demko. Iterated function systems and the global construction of fractals. In *Proc. Roy. Soc. London*, number A399, pages 243–275, 1985.

[6] M. F. Barnsley, V. Ervin, D. Hardin, and J. Lancaster. Solution of an inverse problem for fractals and other sets. In *Proc. Nat. Acad. Sci. USA*, volume 83, pages 1975–1977, 1985.

[7] M. F. Barnsley and L. P. Hurd. *Fractal Image Compression*. Wellesley, Massachusetts, 1993.

[8] A. Buades, B. Coll, and J. M. Morel. Denoising image sequences does not require motion estimation. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 70–74, 2005.

[9] A. Buades, B. Coll, and J.M. Morel. A nonlocal algorithm for image denoising. In *IEEE International conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 60–65, San-Diego, California, June 2005.

[10] A. Buades, B. Coll, and J.M. Morel. A review of image denoising algorithms, with a new one. *SIAM Journal on Multiscale Modeling and Simulation (MMS)*, 4(2):490–530, 2005.

[11] M. Ebrahimi and E. R. Vrscay. Fractal image coding as projections onto convex sets. In *Image analysis and Recognition*, volume 4141, pages 493–506, Berlin/Heidelberg, 2006. Springer.

[12] M. Ebrahimi and E. R. Vrscay. Regularized fractal image decoding. In *Proceedings of CCECE '06*, pages 1933–1938, Ottawa, Canada, May 2006.

[13] M. Ebrahimi and E. R. Vrscay. Regularization schemes involving self-similarity in imaging inverse problems. In *Proceedings of Applied Inverse Problems (AIP) 2007*, University of British Columbia, Vancouver, Canada, June 2007.

[14] M. Ebrahimi and E. R. Vrscay. Solving the inverse problem of image zooming using "self-examples". In *Image analysis and Recognition*, volume 4633, pages 117–130, Berlin/Heidelberg, 2007. Springer.

[15] M. Ebrahimi and E. R. Vrscay. Examining the role of scale in the context of the non-local-means filter. In *Image Analysis and Recognition*, volume 5112, pages 170–181, Berlin/Heidelberg, 2008. Springer.

[16] M. Ebrahimi and E. R. Vrscay. Multi-frame super-resolution with no explicit motion estimation. In *Proceedings of The 2008 International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV 2008*, Las Vegas, Nevada, USA, July 2008.

[17] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Advances and challenges in super-resolution. *International Journal of Imaging Systems and Technology*, 14(2):47–57, August 2004.

[18] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Fast and robust multi-frame super-resolution. *IEEE Transactions on Image Processing*, 13(10):1327–1344, October 2004.

[19] Y. Fisher. *Fractal Image Compression, Theory and Application*. Springer-Verlag, New York, 1995.

[20] Y. Fisher, editor. *Fractal image encoding and analysis*. NATO ASI Series F 159. Springer-Verlag, New York, 1998.

[21] M. Gharavi-Al., R. DeNardo amd Y. Tenda, and T. S. Huang. Resolution enhancement of images using fractal coding. In *Proc. of SPIE Visual Commun. and Image Proc.*, volume 3024, pages 1089–1100, San Jose, CA, 1997.

[22] M. Ghazel, G. Freeman, and E. R. Vrscay. Fractal image denoising. *IEEE Transactions on Image Processing*, 12(12):1560–1578, December 2003.

[23] M. Ghazel, G. Freeman, and E. R. Vrscay. Fractal-wavelet image denoising revisited. *IEEE Trans. Image Proc*, 15:2669–2675, 2006.

[24] J. Hutchinson. Fractals and self-similarity. *Indiana Univ. J. Math.*, 30:713–747, 1981.

[25] A. Jacquin. *A Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding*. PhD thesis, Georgia Institute of Technology, 1989.

[26] A. Jacquin. Image coding based on a theory of iterated contractive image transformations. In *IEEE Trans. Image Proc.*, volume 1, pages 18–30, 1992.

[27] N. Lu. *Fractal Imaging*. Academic Press, New York, 1997.

[28] B. Mandelbrot. *The Fractal Geometry of Nature*. W.H. Freeman, New York, 1983.

[29] G.S. Mayer and E. R. Vrscay. Iterated fourier transform systems: A method for frequency extrapolation. In *Image Analysis and Recognition, Lecture Notes in Computer Science 4633, Proceedings of ICIAR 07*, pages 728–739, Berlin-Heidelberg, 2007. Springer.

[30] D. M. Monro. A hybrid fractal transform. In *Proceedings of ICASSP*, volume 5, pages 162–172, 1993.

[31] D. M. Monro and F. Dudbridge. Fractal block coding of images. *Electron. Lett.*, 28:1053–1054, 1992.

[32] E. Polidori and J. L. Dugelay. Zooming using iterated function. *Fractals*, 5 (Supplementary Issue):111–123, April 1997.