A SIMPLE MODEL FOR AFFINE SELF-SIMILARITY OF IMAGES AND ITS APPLICATIONS

Edward R. Vrscay

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

ABSTRACT

Extensive numerical experiments indicate that images, in general, possess a considerable degree of self-similarity, that is, blocks are well approximated (in the L^2 sense) by a number of other blocks – at the same or different scales – when affine greyscale transformations are employed. This paper outlines a simple model of affine image self-similarity which includes the method of fractal image coding (cross-scale, affine greyscale similarity) and the nonlocal-means denoising method (same-scale, translational similarity) as special cases. Indeed, the general self-similarity of images accounts for the effectiveness of these methods.

A complete metric space (Y, d_Y) of measure-valued image functions is introduced. Associated with each particular self-similarity model is an operator $M : Y \to Y$. The representation of image functions in this space may be useful in self-similar as well as other nonlocal image processing schemes.

Self-similarity is also shown to exist in the wavelet domain, where coefficient quadtrees are approximated by other quadtrees from the same level or higher levels.

Finally, the possibility of going beyond L^2 and using other similarity measures to characterize self-similarity is also discussed.

1. INTRODUCTION

The term “image self-similarity” is subject to a number of interpretations which are concerned with how well regions of an image can, in some way, be approximated by other regions of the same image. Two important image processing schemes that are based on different notions of self-similarity are nonlocal-means denoising [6] and fractal image coding [4, 24]. Indeed, as we discuss in this paper, these two methods may be viewed as lying on opposite sides of a spectrum of self-similarity approximation methods.

In NL-means denoising, self-similarity is understood in the strict *translational* sense: Given an image function u and two $n \times n$ pixel blocks R_i and R_j, two image subblocks $u(R_i)$ and $u(R_j)$ are considered to be “close” only if $u(R_i) \approx u(R_j)$, i.e., the distance $\|u(R_i) - u(R_j)\|$ is small.

On the other hand, traditional fractal image coding seeks to approximate a subblock $u(R_i)$ by a greyscale-modified and spatially-contracted (i.e., decimated) image subblock $u(D_j)$, i.e., $u(R_i) \approx \alpha\tilde{u}(D_j) + \beta$, where D_j is larger than R_i and ˜ denotes decimation. This may be viewed as an exploitation of *cross-scale* image self-similarity.

Both NL-means denoising and fractal image coding involve *nonlocal image processing*: the greyscale value of an image $u(x)$ is replaced by a transformed value $Tu(x)$ which is determined by one or more values $u(y_k)$, where the points $y_k = y_k(x)$ lie elsewhere in the image and not necessarily close to x. This is, of course, in contrast to standard image processing methods which are *local* in nature, i.e., the points y_k lie in a neighbourhood of x. Indeed, the exceptional success of the NL-means denoising method has been responsible for a surge of interest in nonlocal image processsing, as witnessed at this workshop.

In this paper, we outline a simple model of *local affine image self-similarity*, introduced in [3, 2], that includes NL-means denoising and fractal image coding as special cases. The model is a *low-level* one since it is based on block similarities and not on patterns or context. The original motivation came from more recent work in fractal image coding that showed its effectiveness in image denoising [1, 15, 16]. We have also been inspired by the increasing interest in nonlocal methods of image processing that exploit self-similarity, for example, restoration [29], denoising [6, 7] and zooming [9, 14] – see also [8].

Our model examines the distributions of errors Δ_{ij} in approximating image blocks $u(R_i)$ by affine greyscale transformations of other image blocks $u(D_j)$. Images with error distributions that are more concentrated near zero error may be viewed as possessing greater degrees of self-similarity. This, in turn, suggests that relative degrees of self-similarity can be characterized quantitatively in terms of the means and variances of the error distributions. Our results provide some explanation of why self-similar-based methods, including fractal image coding, work so well in approximating or denoising images quite effectively.

In the above applications and, indeed, in most practical schemes, images are represented as real-valued functions, $u : X \to \mathbb{R}_g$, where X denotes the *base space* or *pixel space* over which the images are defined and $\mathbb{R}_g \subset$

$\mathbb{R}$ is a suitable *greyscale space*, assumed to be compact. Usually, the space of functions $\mathcal{F}(X)$ employed is $L^2(X)$, the space of square-integrable functions on X. There are, however, situations in which it is useful to consider the greyscale value of an image u at a point x as a random variable. As such, it may be more useful to represent images by *measure-valued* functions, for example, $\mu : X \to \mathcal{M}(\mathbb{R}_g)$, where $\mathcal{M}(\mathbb{R}_g)$ is the set of Borel probability measures supported on $\mathbb{R}_g$. Such a representation of images, to be discussed later in the paper, can be useful in our affine self-similarity model for at least a couple of reasons:

1. As a kind of intermediate step before the final "projection" of values $u(y_k)$ to produce the transformed value $Tu(x)$.

2. Using the measure $\mu(x)$ to characterize the *local self-similarity* of the image I at a point $x \in X$.

2. A SIMPLE CLASS OF MODELS FOR IMAGE SELF-SIMILARITY

In the computations presented below, we work with normalized images, i.e., $u : X \to \mathbb{R}_g$, where $\mathbb{R}_g = [0,1]$. The support X of an image function u is assumed to be an $n_1 \times n_2$-pixel array. The components of our model are as follows:

1. A set $\mathcal{R}$ of $n \times n$-pixel *range* subblocks R_i, $1 \leq i \leq N_R$ such that (i) $R_i \cap R_j = 0$ if $i \neq j$ and (ii) $X = \cup_i R_i$. In other words, $\mathcal{R}$ forms a partition of X. We let $u(R_i)$ denote the portion u that is supported on R_i.

2. A set $\mathcal{D}$ of $m \times m$-pixel *domain* subblocks D_j, where $m \geq n$. The set of blocks $\mathcal{D}$ should cover X, i.e., $\cup_j D_j = X$ but they need not be nonoverlapping.

3. The one-to-one geometric transformations $w_{ij}^{(k)}$ that map a domain block D_j to range block R_i. For simplicity, we consider only *affine* transformations. The 8 possible mappings (inversions, rotations) are are accomodated in the index $1 \leq k \leq 8$. (For notational convenience, however, the k superscripts will be omitted.) In the case that $m > n$, i.e., D_j is larger than R_i, it is also assumed that the *contractive* map w_{ij} includes an appropriate pixel decimation operation.

4. Affine greyscale maps $\phi : \mathbb{R}_g \to \mathbb{R}_g$ having the form $\phi(t) = \alpha t + \beta$.

Given an image function u, we examine how well or poorly the subimages $u(R_i)$ are approximated by subimages $u(D_j)$, to be written symbolically as

$$u(R_i) \approx \phi_{ij}(u(D_j)) = \alpha_{ij} u(D_j) + \beta_{ij}, \qquad (1)$$

for $1 \leq i \leq N_R$ and $1 \leq j \leq N_D$, with the understanding that the relation applies at the single pixel level. The error Δ_{ij} associated with this approximation is given by

$$\Delta_{ij} = \min_{\alpha,\beta \in \Pi} \|u(R_i) - \alpha u(D_j) - \beta\|. \qquad (2)$$

Here, $\|\cdot\|$ denotes the $L^2(X)$ norm. In all calculations reported in this paper, the L^2-distance between two $n \times n$ image subblocks $u(R_i)$ and $v(R_i)$ will be the root-mean-square (RMS) distance. $\Pi \subset \mathbf{R}^2$ denotes the feasible (α, β) parameter space, which is appropriate for the self-similarity scheme being examined (one of the four cases listed below) and which guarantees that $\phi : \mathbb{R}_g \to \mathbb{R}_g$. We assume that Π is compact.

There are four important cases:

1. **Purely translational:** The domain and range blocks have the same size, i.e., $m = n$. As such, the w_{ij} are translations and $\alpha_{ij} = 1$, $\beta_{ij} = 0$. The approximation error is simply

$$\Delta_{ij}^{(1)} = \|u(R_i) - u(D_j)\|. \qquad (3)$$

2. **Translational + greyscale shift:** The w_{ij} are again translations. We set $\alpha_{ij} = 1$ and optimize over β:

$$\beta_{ij} = \bar{u}(R_i) - \bar{u}(D_j), \quad \Delta_{ij}^{(2)} = |\beta_{ij}|. \qquad (4)$$

where the bars denote mean values of the subblocks.

3. **Affine, same-scale:** The w_{ij} are translations and we optimize over α, β. In the unrestricted case, the expressions for α and β are standard regression formulas.

4. **Affine, two-scale:** The w_{ij} are affine spatial contractions (which involve decimations in pixel space). We optimize over α, β.

As emphasized in [2], this model has been made as simple as possible:

1. The use of *nonoverlapping, square* blocks: An effort to standardize the method, with low computational cost. Generally, the same behaviour is observed for larger numbers of overlapping blocks.

2. The use of range blocks of the *same size*: Generally, the smaller a block, the easier it is to approximate it. We are attempting to keep all regions of an image "on the same playing field." That being said, it is certainly possible that different features will appear at various scales, resulting in different self-similarity statistics.

3. The use of *affine* greyscale maps $\phi(t) = \alpha t + \beta$. Such a family of maps is very simple in form yet, with two parameters, sufficiently flexible. More important, the functions are *monotone*.

Of course, there remains the question of how to define, or even characterize, the self-similarity of an image. In [2], we attempted to to characterize the self-similarity of images in terms of the the distribution of errors Δ_{ij} defined in Eq. (2). In general, as we show below, the Δ-error distributions associated with affine greyscale transformations – Cases 3 and 4 – demonstrate significant peaking. A more concentrated peaking near zero error suggests a greater degree of overall self-similarity.

3. CASES 1,2 AND 3: SAME-SCALE SELF-SIMILARITY

Here, the domain and range blocks have the same size. We naturally expect that for a given domain-range pairing (D_j, R_i), the approximation errors of Eq. (2) for Cases 1, 2 and 3 will behave as follows:

$$0 \leq \Delta_{ij}^{(3)} \leq \Delta_{ij}^{(2)} \leq \Delta_{ij}^{(1)}, \qquad (5)$$

since one optimizes over more parameters as we move from Case 1 (no parameters) to Case 2 (one parameter) to Case 3 (two parameters). In the numerical experiments reported below, the domain and range blocks were taken from the same set of nonoverlapping 8×8-pixel blocks, i.e., $D_i = R_i$.

We first examine the translational similarity of some test images, i.e., Case 1. In Fig. 1 are plotted histogram distributions of the approximation errors $\Delta_{ij}^{(1)}$, which are simply the L^2 distances between subblocks $u(R_j)$ and $u(R_i)$, for the 512×512-pixel normalized test images *Lena* and *Mandrill*. At first glance, it would appear that the two images are quite translationally self-similar since both distributions have significant peaks at around 0.15, that of the *Mandrill* image being more pronounced.

In Fig. 2 are presented the Δ-error distributions for the *Lena* and *Mandrill* for all three cases. The reduction in approximation errors as one moves from Case 1 (shaded) to Cases 2 and 3 is clearly demonstrated, with the latter two distributions exhibiting greater peaking near zero. We observe that enormous improvements are achieved for the *Lena* image in going even from Case 1 to Case 2, where only the greyscale shift parameter β is employed. (Note that the distributions are plotted over the subinterval $[0, 0.5]$.) From these plots, we would conclude that the two images are more *affinely* self-similar than they are *translationally* self-similar, particularly in the case of the *Lena* image.

In Fig. 3 are plotted the histogram distributions of the standard deviations $\sigma(u(R_i))$ of the 8×8 range blocks. There is a noteworthy similarity between these distributions and the Case 3 distributions of Fig. 1 which can be explained as follows. The standard deviation of an image block $\sigma(u(R_i))$, is the RMSE error in approximating $u(R_i)$ by its mean value, $\bar{u}(R_i)$. This is equivalent to setting the greyscale parameter α to zero and optimizing over β in Eq. (2). Removing the condition $\alpha = 0$ will generally produce better approximations, i.e.,

$$0 \leq \Delta_{ij}^{(3)} \leq \sigma(u(R_i)). \qquad (6)$$

As such, the Case 3 Δ-error distributions will be shifted perturbations of the block variance distributions. That being said, we observe that the distributions of α greyscale coefficients generally demonstrate a significant peaking at zero, as shown in Fig. 4. As such, the perturbations from the $\sigma(u(R_i))$ distributions to the Case 3 Δ-error distributions will be small.

Image	Collage errors		
	mean	stddev	entropy
Lena	0.043	0.044	2.26
San Francisco	0.046	0.057	2.01
Peppers	0.047	0.050	2.32
Goldhill	0.049	0.034	2.46
Boat	0.052	0.052	2.58
Barbara	0.060	0.049	2.69
Mandrill	0.089	0.048	2.85
Zelda	0.126	0.055	3.09

Table 1. Means, standard deviations, and entropies of Case 3 and 4 collage error distributions for some standard test images. From [2].

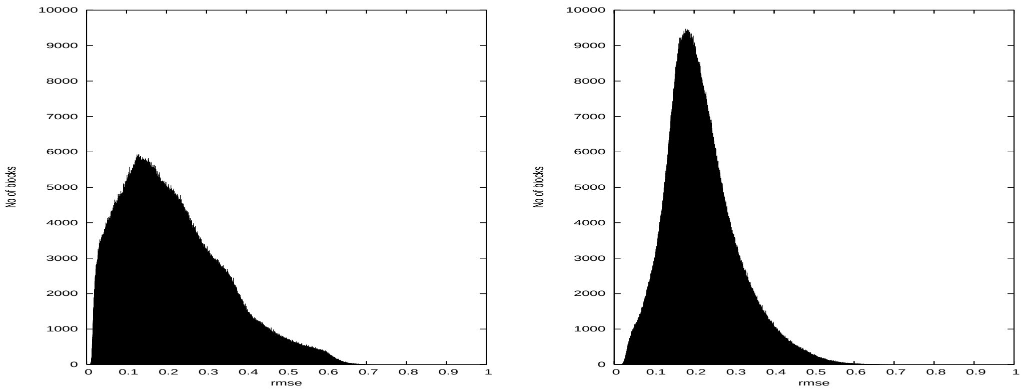
The above discussion, in particular Eq. (6), suggests that the distribution of block variances is the most important factor in how well subblocks of an image I may be approximated by other subblocks, i.e., its degree of "self-similarity." It also explains the significant differences in Δ-error distributions between the *Lena* and *Mandrill* images, especially for Cases 3 and 4. Fig. 3 shows that the *Lena* image contains a significantly higher proportion of "flatter" image subblocks, i.e., blocks of low variance. From Eq. (6), the Case 3 Δ-error distribution for *Lena* will be more concentrated near zero. The benchmark case of self-similarity is the constant image $u = C$. Here, all three $\Delta^{(q)}$-error distributions consist of a single peak at $\Delta = 0$.

In [2], we examined the Case 3 and 4 Δ_{ij} distributions for a number of test images. Some results are presented in Table 1. The entries have been arranged in a rough order of "decreasing self-similarity" based upon increasing mean and, to some extent, increasing width. Estimates of the (natural logarithm) entropies of these distributions have also been presented in this table (third column) – note that with the exception of *San Francisco*, they increase as we proceed down the table. The *Lena* and *Mandrill* images can be viewed as lying roughly on opposite sides of a spectrum of distributions that vary in their mean value and variances.

There may well be concern that the images examined above do not form a suitably broad sampling of "natural images." For this reason, the experiments have been repeated on a much larger set of natural images, namely 700 images from 21 datasets in total taken from the University of Washington 'Groundtruth Database'. The findings were qualitatively similar to those reported above.

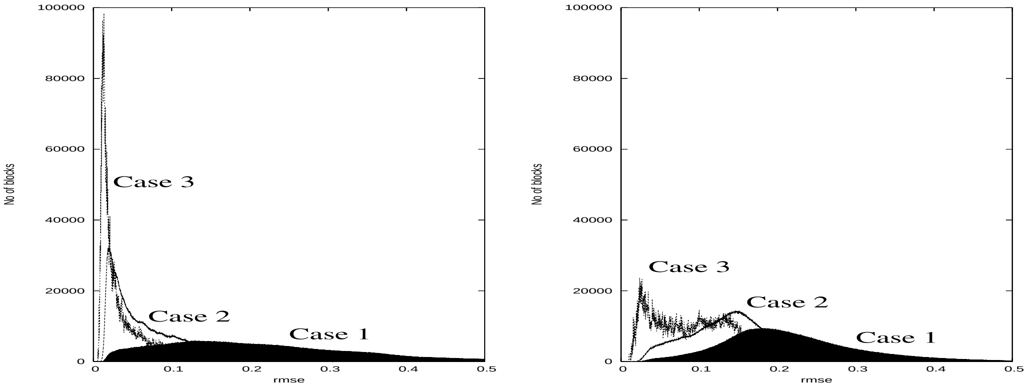
3.1. The effects of noise on Δ-error distributions

The presence of noise in an image will generally decrease the ability of its subblocks to be approximated by other subblocks. As such, we expect that the Δ-error distribu-



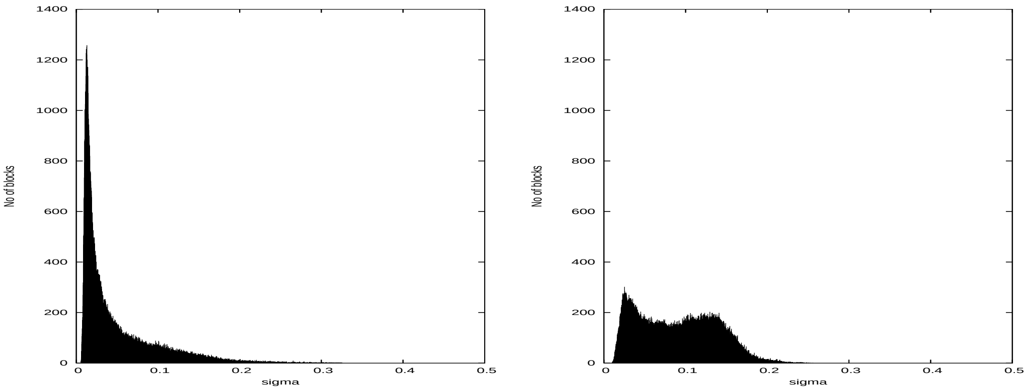
(a) *Lena* (b) *Mandrill*

Figure 1. Case 1 error distributions $\Delta_{ij}^{(1)} = \|u(R_j) - u(R_i)\|$, $i \neq j$, for 512×512-pixel normalized *Lena* and *Mandrill* images, over interval $[0, 1]$. 8×8-pixel blocks R_i.



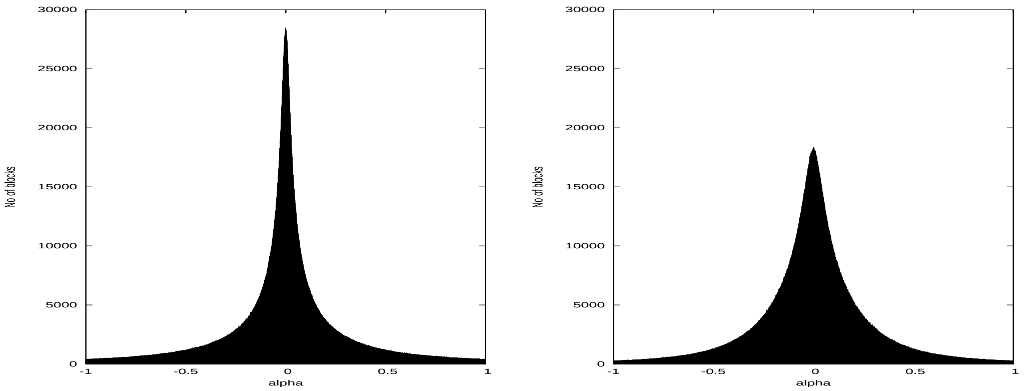
(a) *Lena* (b) *Mandrill*

Figure 2. Same-scale RMS self-similarity error distributions – Cases 1, 2 and 3 – for normalized *Lena* and *Mandrill* images, over interval $[0, 0.5]$. Case 1 distributions from Fig. 1 are shaded. In all cases, 8×8-pixel blocks R_i were used.



(a) *Lena* (b) *Mandrill*

Figure 3. Distributions of $\sigma(u(R_i))$ of 8×8-pixel blocks for normalized *Lena* and *Mandrill* images, over the interval $[0, 0.5]$. Note the similarity to Case 3 distributions of Fig. 2.



(a) *Lena*

(b) *Mandrill*

Figure 4. Distributions of Case 3, same-scale, α greyscale coefficients for normalized *Lena* and *Mandrill* images. The distributions for the Case 4, cross-scale cases are virtually identical.

tions will be shifted away from zero error and possibly broadened as well. This was observed in [2] and will be discussed only briefly here.

In Fig. 5 are shown the Δ-error distributions for the normalized *Lena* image for added Gaussian noise $\mathcal{N}(0, \sigma^2)$ with several variances (right column). In the left column are shown the Δ-error distributions for the "pure noise" images $u = 0.5 + \mathcal{N}(0, \sigma^2)$ for comparision. (In these cases, the Δ-distributions of the noiseless images, $u = 0.5$, consist of a single peak at $\Delta = 0$.) In all cases, the peaks of the distributions move outward as σ increases. Moreover, the peaks for the noisy *Lena* roughly coincide with the peaks of their pure noise counterparts. For Case 3, it can be shown [2] that the Δ-distributions for pure noise images peak at the standard deviation σ. In fact, this is is the basis of the classical block-based noise variance estimation method found in textbooks, e.g., [17].

Finally we note that for sufficiently low σ, i.e., $\sigma \leq 0.1$, the Δ-error distributions of the pure noise images are sharper than those of the noisy *Lena* images. For such low σ-values, blocks of pure noise images are quite "flat", hence well-approximable by other blocks. On the other hand, natural images will have blocks with more structure – these blocks will be less approximable. These observations may provide a partial answer to a question posed by D. Ruderman [26], namely, "How do natural images differ from random ones?"

3.2. Application to "nonlocal-means denoising"

As is well known, a standard technique for the reduction of additive white noise is to average over multiple samples. This is the basis of the very effective "nonlocal-means denoising algorithm" [6], where the multiple samples are provided by the image itself. Very briefly, each pixel $u(i)$ of a noisy image is replaced by a convex combination of other pixel values $u(j)$ from the image. The weights λ_{ij} of this averaging procedure depend upon the similarity between neighbourhoods N_i and N_j centered about pixels i and j, respectively. Neighbourhoods N_k that do not approximate N_i very well, i.e., with high L^2 error $\|N_i - N_k\|$, are assigned low weights. In essence, the nonlocal-means algorithm relies on the translational self-similarity of an image, i.e., Case 1.

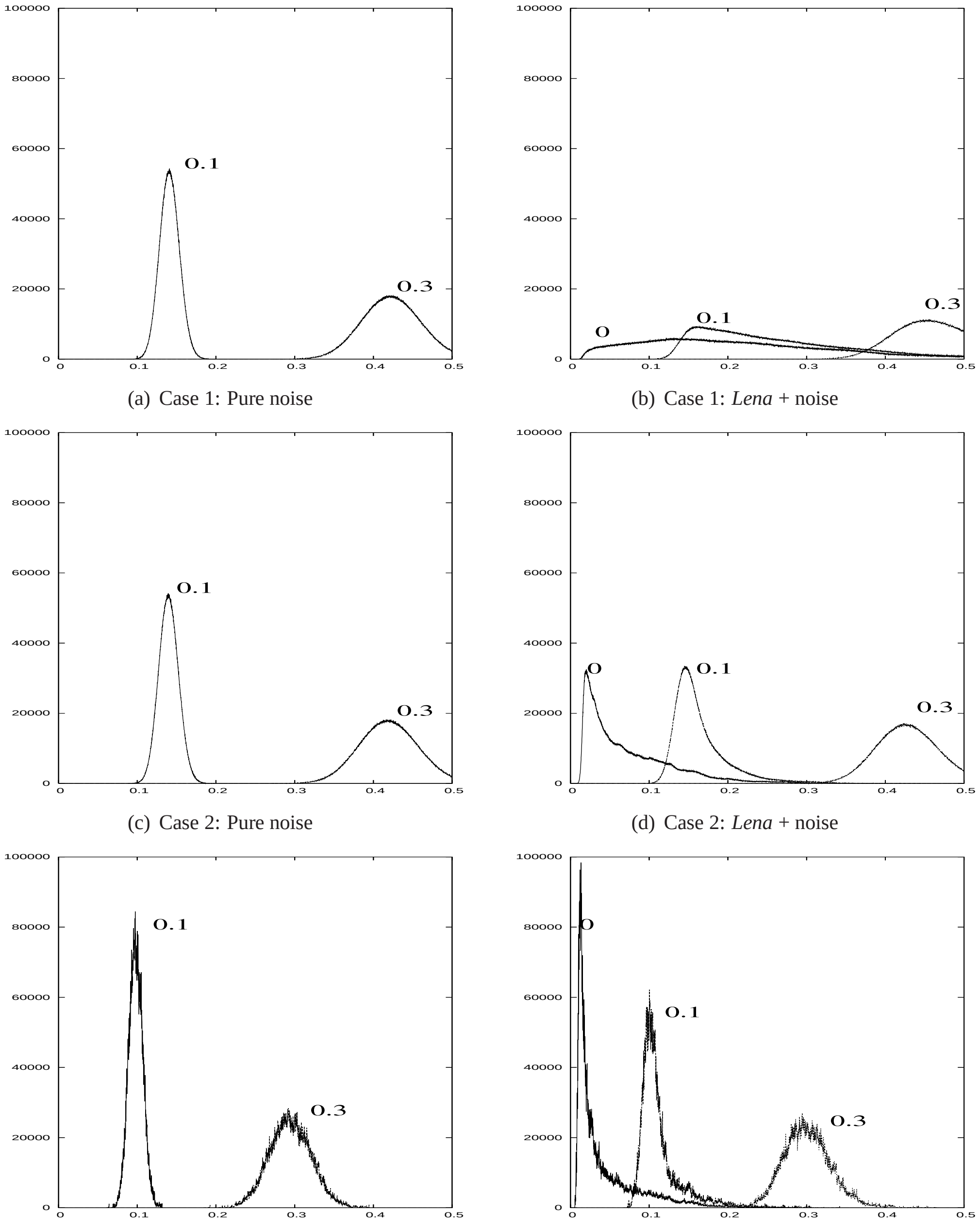
It is remarkable that the NL-means denoising method works so well. Because of the translational symmetry requirement, only a few blocks generally contribute significantly to the denoising of a given pixel. In some applications, it would not seem unreasonable to relax this restriction and allow constant greyscale shifts (Case 2), thereby increasing significantly the number of blocks that could contribute to the denoising. As we shall show later in this paper, such a slight relaxation of the method is observed to improve denoising. Moreover, the computational cost is minimal since the optimal greyscale shifts β are easily computed, cf. Eq. (4).

4. CASE 4: TWO-SCALE, AFFINE SELF-SIMILARITY

The Δ-error distributions for Case 4, cross-scale matching are generally quite similar to their Case 3, same-scale counterparts. In Fig. 6 are presented histogram plots of the approximation errors Δ_{ij} for the *Lena* and *Mandrill* test images, once again for range block partitions $\mathcal{R}$ formed by the set of all 8×8 nonoverlapping pixel blocks of the images ($64^2 = 4096$ in total). For each image, the domain pool $\mathcal{D}$ was formed from the set of $32^2 = 1024$ 16×16 non-overlapping pixel blocks. In addition, for each range/domain block pairing R_j/D_i, we considered all eight square-to-square contractions, for a total of 33,554,432 collage errors. The histogram distributions are very similar to the Case 3 same-scale Δ-error distributions shown in Fig. 2.

The effect of additive Gaussian noise $\mathcal{N}(0, \sigma^2)$ on Case 4, cross-scale Δ-error distributions is the same as for the Case 3, same-scale case: As the variance σ^2 is increased, the Δ-error distributions are pushed away from the zero-error axis. Moreover, these distributions will peak at σ. The Case 4 distributions are virtually identical to those in the bottom row of Fig. 5 and will not be included here.

Case 4, cross-scale self-similarity forms the basis of



(a) Case 1: Pure noise

(b) Case 1: *Lena* + noise

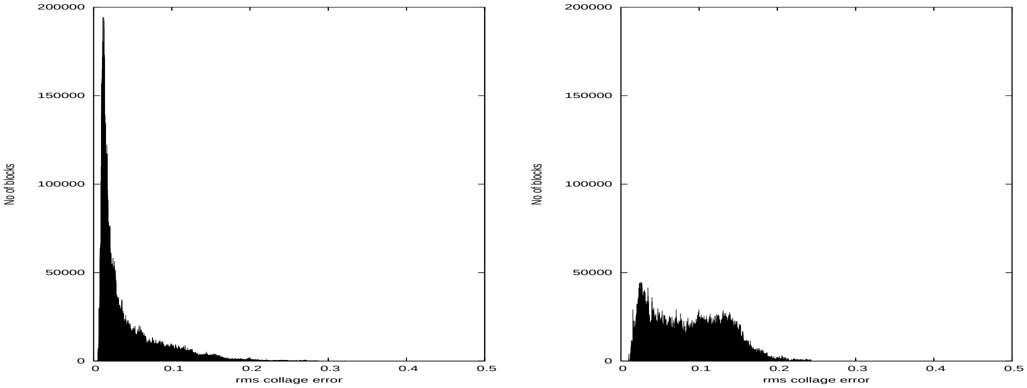
(c) Case 2: Pure noise

(d) Case 2: *Lena* + noise

(e) Case 3: Pure noise

(f) Case 3: *Lena* + noise

Figure 5. Δ-error distributions for constant image $u = 0.5$ (left) and normalized *Lena* image (right) plus independent Gaussian noise $\mathcal{N}(0, \sigma^2)$, Cases 1-3. For $\sigma = 0$, the Δ-distributions of the image $u = 0.5$ (left column) consist of a single peak at $\Delta = 0$.



(a) Case 4: *Lena*

(b) Case 4: *Mandrill*

Figure 6. Histogram distributions of two-scale (Case 4) collage errors $\Delta^{(4)}_{ijk}$ for all 16×16 to 8×8 pixel domain-range block pairings (and 8 square-to-square contractions) for *Lena* and *Mandrill* images.

fractal image coding [11, 24]. Given a "target" image u, each subblock $u(R_i)$ is approximated by a geometrically-contracted and affine greyscale-modified copy of a larger subblock $u(D_j)$. In traditional fractal coding, the range-domain assignments $(i, j(i))$ and associated greyscale parameters (α_i, β_i) define a *fractal transform* operator T. We may write Eq. (1) as follows,

$$u(x) \approx (Tu)(x) = \alpha_i u(w^{-1}_{i,j(i)}(x)) + \beta_i, \quad x \in R_i. \quad (7)$$

Under appropriate conditions [12] involving the α_i and the contraction factors of the spatial maps $w_{i,j(i)}$, T is contractive in $L^2(X)$. From Banach's fixed point theorem, this implies the existence of a unique fixed point function $\bar{u} = T\bar{u}$. Furthermore, $\bar{u}$ may be generated by iteration: Starting with any seed image u_0, let $u_{n+1} = Tu_n$. Then $u_n \to \bar{u}$ as $n \to \infty$. (In the discrete case, convergence is achieved after a finite number of iterations.) Unless all greyscale parameters β_i are zero, which is never the case in practice, the fixed point $\bar{u}$ is nonzero. And from Eq. (7), $\bar{u}$ is an approximation to the target image u, to be discussed in more detail below.

The fractal transform operator T in Eq. (7) is a *non-local* operator since blocks of an image function are replaced by modified copies of blocks from elsewhere in the image. Indeed, the fixed point $\bar{u}$ of T is affinely self-similar since the approximation in Eq. (7) becomes an equality:

$$\bar{u}(x) = (T\bar{u})(x) = \alpha_i \bar{u}(w^{-1}_{i,j(i)}(x)) + \beta_i, \quad x \in R_i. \quad (8)$$

The mathematical basis for fractal-based approximation is provided by the so-called *Collage Theorem* [5], which is a simple consequence of Banach's Theorem:

$$\|u - \bar{u}\| \le \frac{1}{1 - c_T} \|Tu - u\|, \quad (9)$$

where c_T is the contraction factor of T. Given a set of range blocks $\mathcal{R}$ and a *domain pool* $\mathcal{D}$, one tries to to make the approximation error $\|u - \bar{u}\|$ as small as possible by minimizing the *collage error* $\|u - Tu\|$. From Eq. (7) this is generally done as follows: For each range block R_i, we search the domain pool $\mathcal{D}$ for the block $D_{j(i)}$ that yields the lowest approximation error Δ_{ij} in Eq. (2). (In most applications, the range of the α_{ij} greyscale parameters is restricted to $[-\alpha_{\max}, \alpha_{\max}]$, where $\alpha_{\max}$ is typically 1 in value, in order to guarantee contractivity of T.) For this reason, fractal coding has often been refered to as *self-vector quantization* – the codebook used to approximate an image comes from the image itself.

Of course, if the size of the domain pool $\mathcal{D}$ is increased, it may be possible to decrease the collage error. However, an increase in the size of $\mathcal{D}$ implies a greater computational cost because of the searching involved. This is a well known problem in fractal coding [11, 24]. The procedure outlined above, referred to as *collage coding* in the fractal coding literature, was first proposed by A. Jacquin [21] and forms the basis of most, if not all, *block-based fractal coding* procedures.

In Fig. 7 (lower left) is shown the fixed point approximation $\bar{u}$ to the standard 512×512 pixel *Lena* image (8 bits/pixel) obtained from collage coding, once again using a partition $\mathcal{R}$ of nonoverlapping 8×8-pixel blocks and domain pool $\mathcal{D}$ of nonoverlapping 16×16-pixel blocks. (This choice of domain pool is clearly not optimal.) This image was obtained by starting with the seed image $u_0(x) = 255$ (plain white image) and iterating $u_{n+1} = Tu_n$ to $n = 15$. Iterates u_1, u_2 and u_3 are also shown in this figure.

Historically, most fractal image coding research was concerned with compression – obtaining the best possible accuracy with the smallest domain pool. As such, these investigations were primarily concerned with the accuracy provided by the optimal domain blocks in a given domain pool. The fact that most range blocks are well approximated by a number of range blocks does not seem to have been discussed, let alone characterized quanttatively, as done above.

Fractal image denoising: As with any lossy compres-

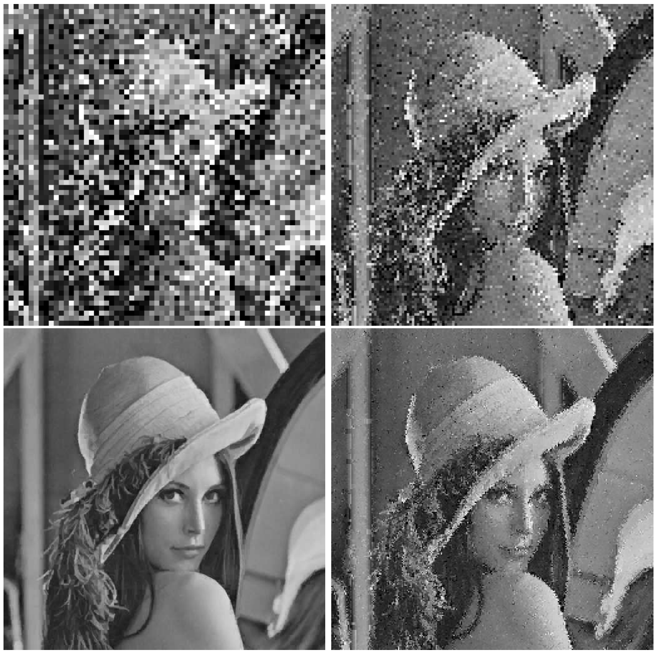


Figure 7. Starting at upper left and moving clockwise: The iterates u_1, u_2 and u_3 along with the fixed point $\bar{u}$ of the fractal transform operator T designed to approximate the standard 512×512 (8 bpp) "Lena" image. The "seed" image was $u_0(x) = 255$ (plain white). The fractal transform T was obtained by "collage coding" using 4096 8×8 nonoverlapping pixel range blocks. The domain pool consisted of the set of 1024 nonoverlapping 16×16 pixel blocks.

sion method, the simple fractal coding of a noisy image produces some denoising [15, 16] . 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 obtained by using estimates of the noise variance to estimate the fractal code of the noiseless image [15].

The fact that each range block is well approximated by a number of domain blocks can be exploited to perform denoising by using multiple copies [1], a cross-scale analog of the nonlocal means denoising method. We shall return to a discussion of such a *multiparent fractal transform* approach later in the paper.

5. MEASURE-VALUED IMAGES AND THEIR APPLICATIONS

We now show how a measure-valued image formulation may be used in this self-similarity model. Mathematical details are to be found in [22].

5.1. A complete metric space of measure-valued images

In what follows, $X = [0,1]^n$ will denote the "base space," i.e., the support of the images. $\mathbb{R}_g \subset \mathbb{R}$ will denote a compact "greyscale space" of values that our images can assume at any $x \in X$. (The following discussion is easily extended to $\mathbb{R}_g \subset \mathbb{R}^m$ to accomodate color images, etc.) Let $\mathcal{M}$ denote the set of all Borel probability measures on $\mathbb{R}_g$ and d_H the Monge-Kantorovich metric on this set:

$$d_H(\mu, \nu) = \sup_{f \in Lip_1(X,\mathbb{R})} \left[\int_X f d\mu - \int_X f d\nu \right]. \qquad (10)$$

Here, $Lip_1(X, \mathbb{R})$ denotes the set of functions $f : X \to \mathbb{R}$ such that $|f(x_1) - f(x_2)| \leq d(x_1, x_2)\ \forall x_1, x_2 \in X$.

For a given $M > 0$, let $\mathcal{M}_1 \subset \mathcal{M}$ be a complete subspace of $\mathcal{M}$ such that $d_H(\mu, \nu) \leq M$ for all $\mu, \nu \in \mathcal{M}_1$. We now define

$$Y = \{\mu(x) : X \to \mathcal{M}_1, \mu(x) \text{ is measurable}\} \qquad (11)$$

and consider on this space the following metric,

$$d_Y(\mu, \nu) = \int_X d_H(\mu(x), \nu(x)) dm(x), \qquad (12)$$

where m denotes Lebesgue measure on X. The metric d_Y is well defined, since μ and ν are measurable functions. Moreover, d_H is bounded so that the function $\xi(x) = d_H(\mu(x), \nu(x))$ is integrable on X. In [22], it was proved that the space (Y, d_Y) is complete.

5.2. Application to same-scale self-similarity

In what follows, we again assume that the image function u is an $n_1 \times n_2$ pixel array with greyscale range $\mathbb{R}_g = [0,1]$. We also assume that the range and domain blocks come from a common pool $\mathcal{R}$. We now consider all possible range-domain block pairs (R_i, R_j), along with the *isometric* affine transformations $w_{ij}^{(k)}$, $1 \leq k \leq 8$, that map R_j to R_i. For simplicity of notation, we omit the k

index. (Indeed, in most of the examples presented in this section, we employ only translations, i.e., $k = 1$.)

For a given same-scale scheme, i.e., Case q, where $q \in \{1, 2, 3\}$, we first compute all possible appropriate approximation errors $\Delta_{ij}^{(q)}$, cf. Eq. (2) and let $\phi_{ij}^{(q)}(t)$ denote the greyscale maps associated with these errors. (Recall that the greyscale maps assume different forms in the three Cases.) For each range-domain pairing (R_i, R_j), we then assign a weighting function p_{ij} which is normalized as follows,

$$\sum_{j=1}^{N_R} p_{ij} = 1, \quad 1 \leq i \leq N_R. \tag{13}$$

An obvious question is the choice of the weighting parameters p_{ij}. It would seem natural to employ higher weights for those domain blocks R_j that yield lower approximation errors Δ_{ij}. Here we consider a weighting scheme that is similar in form to the one used in [6] for NL-means denoising:

$$p_{ij} = \frac{1}{Z_i} \exp\left(-\frac{\Delta_{ij}^P}{h^P}\right), \tag{14}$$

where $P > 0$, $h > 0$ and $Z_i = \sum_j \exp(-\Delta_{ij}^P / h^P)$ is the normalization factor. In practice, P is either 1 or 2. As for the adjustable parameter h, note that:

1. In the limit $h \to 0$, the p_{ij}(s) with the smallest error Δ_{ij} will be selected.

2. In the limit $h \to \infty$, all p_{ij} become equal.

For a given Case q and a set of prescribed weights $\{p_{ij}\}$ we may define an operator $T^{(q)}$, the action of which on an image function $u(x)$ is given as follows: For an $x \in R_i$,

$$\begin{aligned} v(x) &= (T^{(q)}u)(x) \\ &= \sum_{j=1}^{N_R} p_{ij} \phi_{ij}^{(q)}(u(w_{ij}^{-1}(x))). \end{aligned} \tag{15}$$

In other words, the value $u(x)$ at an $x \in R_i$ is replaced by a weighted sum of modified pixel values – the *preimages* of the value $u(x)$ – from all other blocks R_j. The $T^{(q)}$ operators are therefore nonlocal operators.

The following result [22] establishes that the same-scale "collage distance," $\|T^{(q)}u - u\|$ (cf. Eq. (9), reflects the total Case q self-similarity of the image:

$$\|T^{(q)}u - u\| \leq \sum_{i,j}^{N_R} p_{ij} \Delta_{ij}^{(q)}. \tag{16}$$

For the special self-similar image $u = C$, $\Delta_{ij} = 0$ for all i, j, implying that $\|T^{(q)}u - u\| = 0$. In other words, $u = C$ is a fixed point of $T^{(q)}$ for $q \in \{1, 2, 3\}$.

The $T^{(q)}$ operators essentially "collapse" all preimages of an image function value $u(x)$ onto a single value $T^{(q)}u(x)$. The measure-valued image function formalism introduced above allows us to keep track of the *range of values* assumed by these preimages. For this purpose, we associate with the image function $u(x)$ a corresponding measure-valued image function $\mu(x) \in (Y, d_Y)$ as follows:

$$\mu(x) = \delta_{u(x)}, \quad x \in X. \tag{17}$$

Here, δ_t denotes a unit point mass measure at $t \in \mathbb{R}_g$. We now define a measure-valued image $\nu = M\mu \in Y$ as follows: For any measurable set $S \subset \mathbb{R}_g = [0, 1]$ and an $x \in R_i$, we define

$$\begin{aligned} \nu^{(q)}(x)(S) &= (M^{(q)}\mu)(x)(S) \\ &= \sum_{j=1} p_{ij} \mu(w_{ij}^{-1}(x))(\phi_{ij}^{-1(q)}(S)). \end{aligned} \tag{18}$$

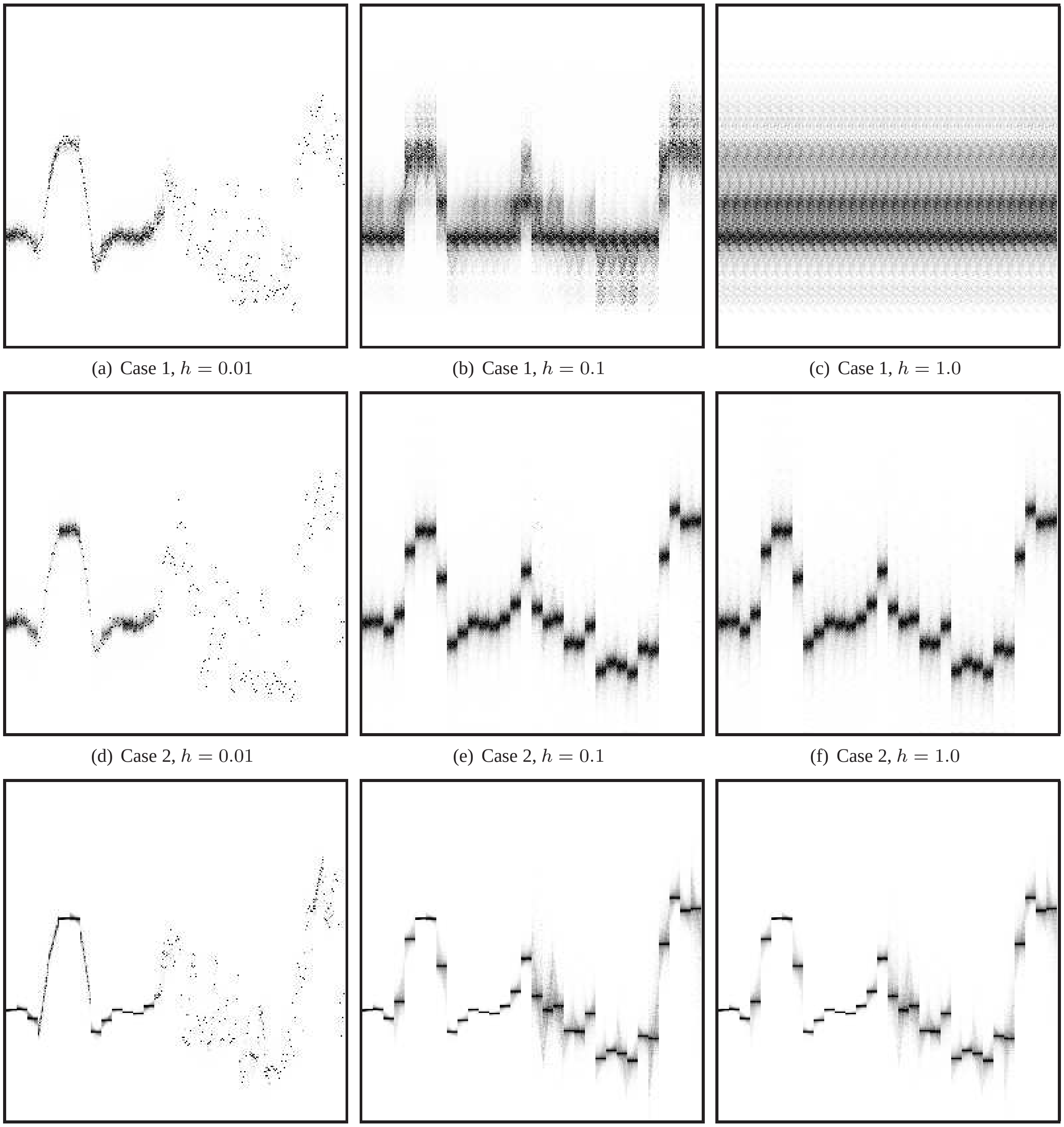
The measure $\nu^{(q)}$, $q \in \{1, 2, 3\}$, is designed to reflect the Case q self-similarity of the image function u: Given a range block $R_i \subset X$, then at each point $x \in R_i$, we keep track of all greyscale values of the image function that are mapped to x by a domain-range mapping w_{ij} and modified by the corresponding Case q greyscale map $\phi_{ij}^{(q)}(t)$. These values are then weighted and "assembled" to define the probability measure $\nu^{(q)}$ at x, i.e., $\nu^{(q)}(x)$.

In Fig. 8 are shown pictorial representations of the measures $\nu^{(q)}(x)$, $q = 1, 2, 3$. For each Case q, we have used three values of the weight parameter h in Eq. (14), along with $P = 2$. In these figures, darker regions have higher associated measures. As expected from the previous figure, the measures become more concentrated about the actual greyscale values $u(256, j)$, $1 \leq j \leq 256$ of the *Lena* image as we move from Case 1 to Case 3. In all three cases, the weight parameter value $h = 0.01$ effectively concentrates the measures close to the *Lena* image values, even for the Case 1 measure. However, as h is increased to 0.1, the Case 1 measure becomes quite diffuse and at $h = 1.0$, quite unrelated to the *Lena* image values. For Cases 2 and 3, there is virtually no change between $h = 0.1$ and $h = 1.0$ and even $h = \infty$ (equal probabilities, not shown in figure).

5.3. Effects of noise on measures and a simple denoising method

In the top row of Fig. 9 are shown pictorial representations of the measures $\nu^{(q)}$ associated with the pixels $u(256, j)$, $1 \leq j \leq 256$, of the normalized *Lena* image with added noise $\mathcal{N}(0, \sigma^2)$, where $\sigma = 0.1$. These pictures should be compared to their noiseless counterparts in the middle column of Fig. 8, As expected, the measures for the noisy case are more diffuse, although the Case 3 measure appears to be as concentrated for the noisy case as for the noiseless one.

A natural question is whether the means $\bar{x}$ of these measures estimate the greyscale values of the noiseless *Lena* image. In the middle row of Fig. 9 are plotted the mean values of these measures for the three cases $q = 1, 2, 3$. The root-mean-square error with respect to the noiseless (normalized) image values $u(256, j)$, $1 \leq j \leq$

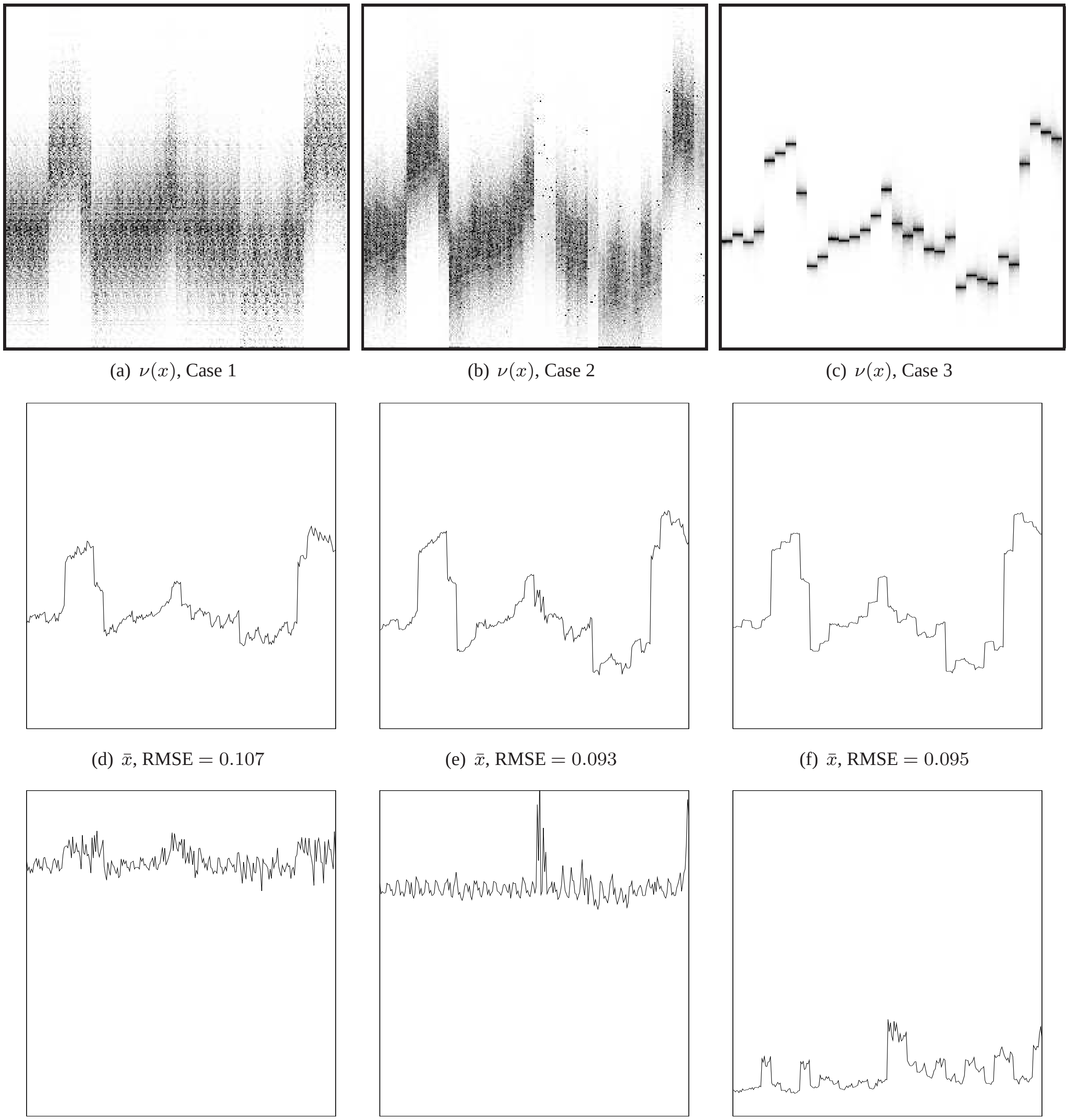


(a) Case 1, $h = 0.01$ (b) Case 1, $h = 0.1$ (c) Case 1, $h = 1.0$

(d) Case 2, $h = 0.01$ (e) Case 2, $h = 0.1$ (f) Case 2, $h = 1.0$

(g) Case 3, $h = 0.01$ (h) Case 3, $h = 0.1$ (i) Case 3, $h = 1.0$

Figure 8. Pictorial representations of measures $\nu(x)^{(q)}$, $q = 1, 2, 3$ for the $u(256, j)$, $1 \leq j \leq 256$ row of pixels for the *Lena* image. Three values of the weighting parameter h in Eq. (14) have been used, with $P = 2$.



(a) $\nu(x)$, Case 1

(b) $\nu(x)$, Case 2

(c) $\nu(x)$, Case 3

(d) $\bar{x}$, RMSE $= 0.107$

(e) $\bar{x}$, RMSE $= 0.093$

(f) $\bar{x}$, RMSE $= 0.095$

(g) $\sigma(x)$

(h) $\sigma(x)$

(i) $\sigma(x)$

Figure 9. Top row: Measures $\nu(x)^{(q)}$ for normalized *Lena* image with added Gaussian noise with $\sigma = 0.1$. In all cases, $h = 0.1$. Vertical axis is $[0, 1]$. Middle row: Mean values $\bar{x}$ of the $\nu(x)^{(q)}$ measures. Vertical axis $[0, 1]$. These results represent denoised values of the noisy *Lena* image and should be compared to the noiseless image values in Fig. 8(a). Bottom row: Variances $\sigma(x)$ of the corresponsding $\nu(x)^{(q)}$ measures. Vertical axis $[0, 0.15]$.

256 are given for each case. As expected, the approximations are poorest in regions of high variance/oscillation, for example, $130 \leq x \leq 200$.

In the bottom row of Fig. 9 are plotted the variances $\sigma(x)$ of the corresponsding $\nu^{(q)}(x)$ measures. In accordance with the plots of the top row, there is a dramatic reduction in variance as we move from Case 2 to Case 3.

Note that, regardless of the comparatively high diffusivity of the Case 1 and Case 2 measures, the accuracy of their means – in terms of RMSE – are virtually identical to that of Case 3. In essence, all three same-scale similarity methods – Cases 1, 2 and 3 – appear to perform denoising almost equally well in this example. This is generally true for the case of coarser approximations. The three methods perform differently when both the value of h and the range block size are decreased to produce finer approximations. For example, Fig. 10 shows the mean value plots that result from $h = 0.05$ and 4×4-pixel range blocks. In this case, there is an improvement in accuracy as we move from Case 1 to Case 3.

The above analysis, however, involves only a single (half) row of pixels in the *Lena* image. When we examine the results of Case 1-3 denoising on a larger portion of the image, as shown in Fig. 11, (upper left quarter of the *Lena* image), the differences between the methods, at least in terms of RMSE values, become more pronounced.

That being said, these few results are not meant to be interpreted as a complete analysis of this approximation problem. A more detailed investigation, which is beyond the scope of this paper, is clearly needed.

The NL-means denoising method [6] may be viewed as an individual pixel-based variation of the Case 1, same-scale operator $T^{(1)}$ of Eq. (15). Because it operates on individual pixels (based on Case 1 similarity between neighbourhoods), the NL-means method will yield better results – of course at greater computational expense. In either procedure, an appropriately-defined operator M and therefore associated measure $\nu(x)$ may be defined as an intermediate step in the computation of the denoised value $v(x)$.

6. A FRACTAL TRANSFORM ON THE SPACE OF MEASURE-VALUED IMAGES

In the same way as was done for Cases 1-3 in the previous section, a cross-scale operator $T^{(4)}$ on image functions that is based on Case 4 self-similarity may be defined. Its functional form will be the same as in Eq. (15) – as such, we forego writing it out explicitly. In this case, however, we must recall that the spatial maps w_{ij} are *contractions*, mapping larger domain blocks D_j onto smaller range blocks R_i.

Such an operator may be called a *multiparent fractal transform*. The use of more than one domain block per range block would certainly be counterproductive from the viewpoint of data compression – one of the original motivations of fractal image coding research. Here, however, were are concerned with image processing and analysis, not compression.

And, following the discussion of the previous section, we may define an operator $M^{(4)}$ on the space of measure-valued functions (Y, d_Y) that correponds to the multiparent fractal transform $T^{(4)}$. This operator will have the same form as in Eq. (18). In [22], a more general, yet "blockless" form of the M operator was introduced and analyzed, namely where the coefficients p_{ij} could be x-dependent, and the greyscale maps ϕ_{ij} were assumed to be Lipschitz. The important point is that under appropriate conditions, the operator M is contractive in the complete metric space (Y, d_Y), implying the existence of a unique measure-valued function $\bar{\mu}$ satisfying the fixed point relation $\bar{\mu}(x) = M\bar{\mu}$. There are some very interesting mathematical consequences, including a recursive structure relating the (place-dependent) moments of the measure $\bar{\mu}$. A discussion of these features, however, is beyond the scope of this paper.

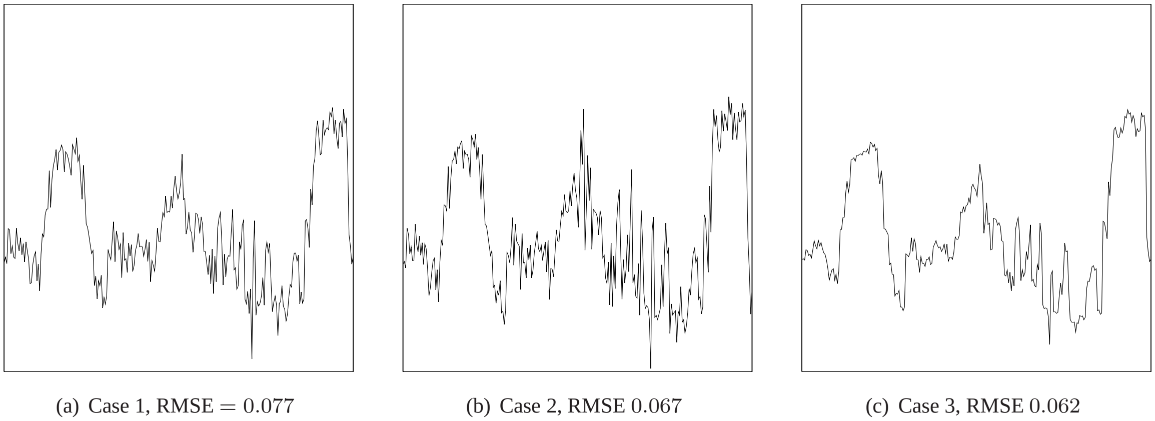
The $T^{(4)}$ operator defines a natural Case 4 counterpart of the Case 1-3 denoising algorithms of the previous section – essentially a cross-scale, block-based, multiparent fractal denoising method, which was investigated in [22]. This method, however, yielded no significant improvement over the same-scale, Case 3 method, either after one application of the $T^{(4)}$ operator after iteration to its fixed point. The results are virtually identical to those of the third column in Fig. 9.

Stepping back a bit in time, the denoising potential of the multiparent transform was shown by S. Alexander in his Ph.D. thesis [1]. Noting that a range block $u'(R_i)$ of a noisy image is generally well approximated by several domain subblocks $u'(D_{j_k}(i))$ after decimation and greyscale modification, these approximations to $u'(R_i)$ could be viewed as samples which, when averaged, would yield an estimate of the denoised subblock $u(R_i)$ – essentially a cross-scale and block-based version of the yet-to-appear nonlocal-means denoising method. The use of several domain blocks for each range block defined a multiparent fractal transform T'. In general, its fixed point, $\bar{u}'$, was found to be a good approximation to the noiseless image u. Several averaging schemes were investigated.

Indeed, most fractal denoising methods have focussed on approximating the unknown noiseless counterpart u of a noisy image u' by the fixed point $\bar{u}'$ of a contractive fractal transform T' that is obtained by fractally coding u'. In contrast, the Case 4 similarity method, following its Case 1-3 counterparts, relies on the application of the transform $T^{(4)}$, obtained from u', *only once* to u'. A more thorough comparison of the two methods remains to be done.

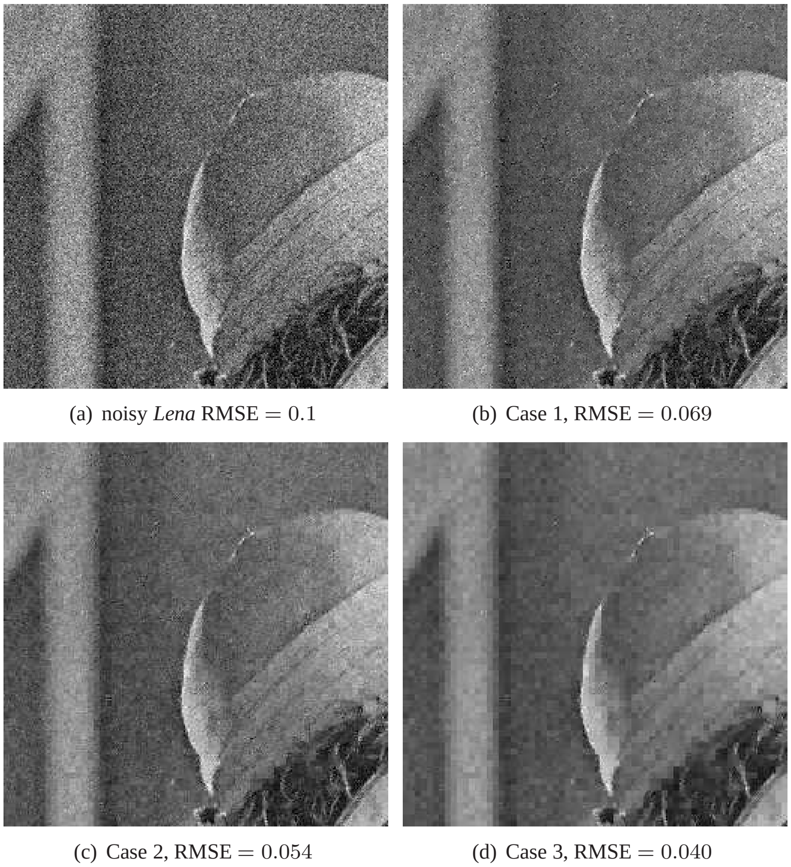
7. SELF-SIMILARITY VS. "APPROXIMABILITY"

Earlier, we showed that the distributions of (Case 3 and 4) Δ-errors of an image function u are virtually identical to the distribution of its block variances $\sigma(u(R_i))$, a consequence of approximating image blocks by their mean values. In this low-level, block-based picture, the self-similarity of an image is essentially determined by the degree of its *flatness*, i.e., the percentage of blocks with



(a) Case 1, RMSE $= 0.077$ (b) Case 2, RMSE 0.067 (c) Case 3, RMSE 0.062

Figure 10. Mean values $\bar{x}$ of the $\nu(x)^{(q)}$ measures obtained from noisy *Lena* image, where parameter $h = 0.05$.



(a) noisy *Lena* RMSE $= 0.1$ (b) Case 1, RMSE $= 0.069$

(c) Case 2, RMSE $= 0.054$ (d) Case 3, RMSE $= 0.040$

Figure 11. Results of Case 1-3 denoising of upper quarter of *Lena* image. In all cases, $h = 0.05$.

near-zero variance.

In [2], however, we observed that very similar (Case 3 and Case 4) Δ-error distributions are obtained if we used blocks from *another* image, e.g., using 8×8-pixel blocks of *Mandrill* to approximate those of *Lena*, and *vice versa*. In this case, the term *approximability* is perhaps more appropriate than *self-similarity*.

The matter of approximability is of paramount importance to vector quantization (VQ). Indeed, the connection between fractal coding and VQ was realized many years ago, for example, [23, 18, 19, 20]. The use of image subblocks from a collection of images to construct codebooks for (cross-scale) fractal coders was nicely illustrated in [23].

Structured vector quantization [10] is a same-scale version of this method: using fixed sets of subblocks to approximate subblocks of the same size, with the help of affine greyscale transformations. In none of the above references, however, are the statistics of domain-range pairings explicitly compiled or analyzed.

8. SELF-SIMILARITY IN THE WAVELET DOMAIN

The ideas of self-similarity developed in Section 1 have straightforward counterparts in the wavelet domain. To date, we have not examined this idea in great detail, and only preliminary results will be presented below. For simplicity, we consider standard tensor-product wavelet basis expansions [25], for which the coefficients are conveniently arranged in a standard matrix form, as shown in Fig. 12. Each of the blocks $\mathbf{A}_k^h$, $\mathbf{A}_k^v$, $\mathbf{A}_k^d$, $0 \leq k \leq K$, for a $K > 0$, contains 2^{2k} wavelet coefficients a_{kij}^h, a_{kij}^v, a_{kij}^d, respectively. (In the case of 512×512-pixel images, $K = 8$.) The three collections of blocks comprise the *horizontal*, *vertical* and *diagonal* quadtrees of the coefficient tree.

For any wavelet coefficient a_{kij}^λ, $\lambda \in \{h, v, d\}$ in this matrix we define A_{kij}^λ to be the unique quadtree with a_{kij}^λ as its root. In the Haar wavelet case, as is well known, for a fixed set of indices $\{k, i, j\}$, the three quadtrees A_{kij}^h, A_{kij}^v and A_{kij}^d correspond to the same spatial block of an image. For generalized wavelets, the correlation remains, although the spatial region is diffused. In Fig. 12 are shown two sets of such quadtree triplets, for two consecutive values of k, i.e., $k = k_1^*$ and $k = k_2^* = k_1^* + 1$.

Given an image function u, with associated wavelet coefficient matrix $\mathbf{A}$, we now examine how well quadtrees A_{kij}^λ are approximated by other quadtrees $A_{k'i'j'}^{\lambda'}$, to be written symbolically as

$$A_{kij}^\lambda \approx \phi(A_{k'i'j'}^{\lambda'}) = \alpha A_{k'i'j'}^{\lambda'}, \quad k' \leq k. \qquad (19)$$

It is certainly possible to consider rotations of the quadtrees in the approximation. In what follows, however, we consider only the rotationless case.

Note that only affine scaling transformations of the form $\phi(t) = \alpha t$ are used. The β shift term is omitted since, at least theoretically for quadtrees of infinite length, their l^2 nature must be preserved. As such, there are only three cases of self-similarity to study:

1. **Translational:** We compare quadtrees at the same scale, i.e., $k = k'$, with $\alpha = 1$.

3. **Affine, same scale:** Again, we compare quadtrees at the same scale, i.e., $k = k'$, but optimize over α.

4. **Affine, two-scale:** Here, $k > k'$, i.e., we approximate higher-scale quadtrees with scaled copies of coarser scale quadtrees that are situated higher in the **A** matrix. In the pixel domain, this corresponds to decimated domain blocks mapped to range blocks.

We have numbered these cases, omitting the nonexistent "Case 2", so that they may be considered as counterparts of the pixel-based self-similarity cases that were studied in earlier sections.

In Fig. 13 we have plotted Δ-error histograms for Cases 1 and 2 of the *Lena* and *Mandrill* test images. Here, $k = k' = 6$. As well, we set $\lambda = \lambda'$ in Eq. (19), i.e., only quadtrees with the same λ value were used to approximate a quadtree A_{kij}^λ. (This restriction can obviously be relaxed.) The approximation errors are simply the l^2 errors between the finite quadtrees. (In this case, they are comprised of $1 + 4 + 16 = 21$ coefficients.) The similarity between these distributions and their pixel-domain counterparts, cf. Fig. 2, is quite striking. The addition of Gaussian noise also pushes these distributions away from zero.

The Case 4 distributions are virtually identical to their Case 2 counterparts and are not plotted here. Indeed, the Case 4 self-similarity exhibited by wavelet subtrees forms the basis of fractal-wavelet image coding [27]. A multiparent fractal-wavelet denoising scheme was investigated in [1]. It is conceivable that wavelet-based NL-means denoising, as well as zooming, could be performed.

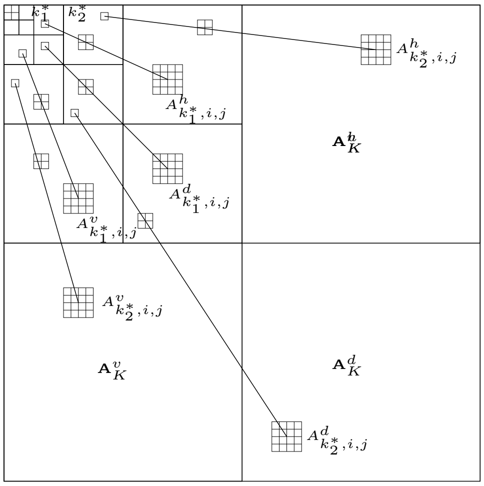
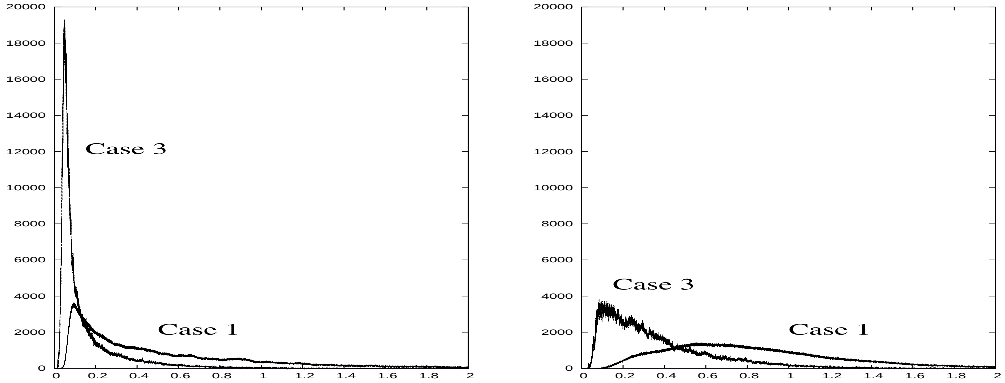


Figure 12. Wavelet coefficient table, showing some particular quadtrees.



(a) *Lena* (b) *Mandrill*

Figure 13. Same-scale RMS self-similarity error distributions for wavelet coefficient subtrees – Cases 1 and 3 – for normalized *Lena* and *Mandrill* images, over interval $[0, 2]$. The same vertical scale was employed in both plots for comparison. Here $k = k' = 6$ and $\lambda = \lambda'$ in Eq. (19).

9. GETTING AWAY FROM L^2: EXPLORING THE USE OF SIMILARITY MEASURES

In [3], we explored the use of mutual information (MI) to characterize self-similarity of image blocks. A motivation lay in the widespread use of MI for the purpose of registering images from different modalities, since each modality captures its own unique information. A primary difference between the usual L^2 metric approach and MI is that the former involves a comparison of greyscale *values*, whereas the latter involves the comparison of greyscale *distributions* and their correlations. As such, it does not necessarily follow that there will always be *visual* self-similarity.

Unfortunately, there are several practical difficulties associated with this approach. The computation of MI relies on an accurate estimation of probability distributions and their entropies. These estimates naturally suffer when the number of quantized greyscale values is too large, or when block sizes are too small. In [3], better results were obtained with six bit-per-pixel (bpp) images, i.e., 64 quantized greyscale values. In Fig. 14 are presented histogram distributions of the MI for 6 bpp *Lena* and *Mandrill* test images. Here, 64×64-pixel subblocks were used – insufficient differentiation between the images was obtained with smaller block sizes. This, of course, is unfortunate since similarities at smaller scales, e.g., 8×8-pixel blocks, cannot be ascertained.

The qualitative difference between the *Lena* and *Mandrill* distributions is opposite to what was observed in the L^2-error case. Here, it is the MI distribution of the *Mandrill* image that is more concentrated toward zero – recall that lower MI values imply greater independence between blocks. And as noise is added, the distributions become sharper, with greater shifting toward zero MI.

We are now in the process of investigating other similarity measures/perceptual metrics, for example, *structural similarity* [28].

Acknowledgements

This research has been supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) in the forms of a Discovery Grant (ERV), which is gratefully acknowledged. The author would like to thank several people for stimulating and helpful discussions during the course of this work, namely, S.K. Alexander (U Houston), D. La Torre (U Milan) and Z. Wang (U Waterloo).

10. REFERENCES

[1] S.K. Alexander, *Multiscale Methods in Image Modelling and Image Processing*, Ph.D. 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 **5112**, 192-203 (2008).

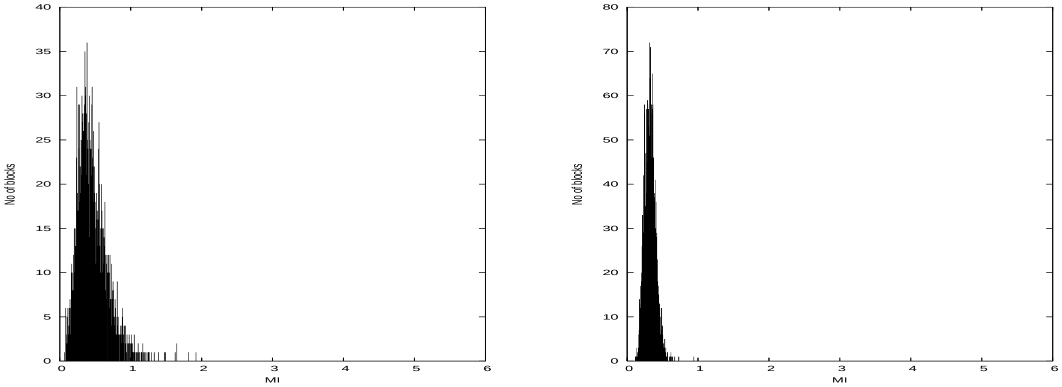
[3] S.K. Alexander, S. Kovačič and E.R. Vrscay, A simple, general model for the affine self-similarity of images, with the possible use of mutual information, Proceedings of the EUSIPCO 2007 Conference, Poznan, Poland, August 2007, p. 975-979.

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

[5] M.F. Barnsley, V. Ervin, D. Hardin and J. Lancaster, Solution of an inverse problem for fractals and other sets, Proc. Nat. Acad. Sci. USA **83**, 1975-1977 (1985).

[6] A. Buades, B. Coll and J.M. Morel, A review of image denoising algorithms, with a new one, Multiscale Modelling and Simulation, **4**, 490-530 (2005).

[7] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering, IEEE Trans. Image Proc. **16**, 2080-2095 (2007).



(a) Lena

(b) Mandrill

Figure 14. Histogram distributions of mutual information $I(R_i, R_j)$ for 512×512, 6 bpp *Lena* and *Mandrill* images using 64×64-pixel blocks.

[8] M. Ebrahimi and E.R. Vrscay, Solving the inverse problem of image zooming using “self-examples”, in *Image Analysis and Recognition*, LNCS 4633, Proceedings of ICIAR 2007, pp. 117-130 (Springer Verlag, Berlin-Heidelberg).

[9] M. Elad and D. Datsenko, Example-based regularization deployed to super-resolution reconstruction of a single image, The Computer Journal, **50**, 1-16 (2007).

[10] C. Etemoglu, V. Cuperman, Structured vector quantization using linear transforms, IEEE Trans. Sig. Proc. **51**, 1625-1631 (2003).

[11] Y. Fisher, Ed., *Fractal Image Compression: Theory and Application*, Springer Verlag, New York (1995).

[12] B. Forte and E.R. Vrscay, Theory of generalized fractal transforms in *Fractal Image Encoding and Analysis*, NATO ASI Series F, Vol 159, ed. Y. Fisher, Springer Verlag, New York (1998).

[13] B. Forte and E.R. Vrscay, Inverse problem methods for generalized fractal transforms, in *Fractal Image Encoding and Analysis*, *ibid*.

[14] W.T. Freeman, T.R. Jones and E.C. Pasztor, Example-based super-resolution, IEEE Comp. Graphics Appl., **22**, 56-65 (2002).

[15] M. Ghazel, G. Freeman and E.R. Vrscay, Fractal image denoising, IEEE Trans. Image Proc. **12**, 1560-1578 (2003).

[16] M. Ghazel, G. Freeman and E.R. Vrscay, Fractal-wavelet image denoising, preprint. IEEE Trans. Image Proc. **15**, 2669-2675 (2006).

[17] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Prentice-Hall, New Jersey (2002).

[18] R. Hamzaoui and D. Saupe, Combining fractal image compression and vector quantization, IEEE Trans. Image Proc. **9**, 197-207 (2000).

[19] R. Hamzaoui, M. Müller and D. Saupe, VQ-enhanced fractal image compression, ICIP 1996.

[20] R. Hamzaoui, Encoding and decoding complexity reduction and VQ asspects of fractal image compression, Ph.D. Thesis, University of Freiburg, 1998.

[21] A. Jacquin, Image coding based on a fractal theory of iterated contractive image transformations, IEEE Trans. Image Proc. **1**, 18-30 (1992).

[22] D. La Torre, E.R. Vrscay, M. Ebrahimi and M.F. Barnsley, Measure-valued images, associated fractal transforms and the affine self-similarity of images, preprint (2008).

[23] S. Lepsoy, P. Carlini and G.E. Oien, On fractal compression and vector quantization, in *Fractal Image Encoding and Analysis*, *op. cit.*.

[24] N. Lu, *Fractal Imaging*, Academic Press, New York (1997).

[25] S. Mallat, *A wavelet tour of signal processing*, Academic Press, New York (1998).

[26] D.L. Ruderman, The statistics of natural images, Network: Computation in Neural Systems **5**, 517-548 (1994).

[27] E.R. Vrscay, A generalized class of fractal-wavelet transforms for image representation and compression, Can. J. Elec. Comp. Eng. **23**, 69-83 (1998).

[28] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli, Image quality assessment: From error visibility to structural similarity, IEEE Trans. Image Proc. **13** (4), 600-612 (2004).

[29] D. Zhang and Z. Wang, Image information restoration based on long-range correlation, IEEE Trans. Cir. Syst. Video Tech. **12**, 331-341 (2002).