A Simple, General Model for the Affine Self-Similarity of Images

Simon K. Alexander

Department of Mathematics

University of Houston, Houston, Texas, USA 77204

Edward R. Vrscay

Department of Applied Mathematics

University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

"Waterloo Fractal Coding and Analysis Group:" http://links.uwaterloo.ca

Satoshi Tsurumi

Department of Information and Computer Engineering

Gunma National College of Technology

580 Toribamachi, Maebashi, Gunma 371-8530, Japan

Research supported by Natural Sciences and Engineering Research Council of Canada (NSERC)

Overview

We show that images generally possess a considerable degree of affine self-similarity, that is, their subblocks are well approximated by a number of other subblocks with the help of affine greyscale transformations. In this paper:

1. We introduce a quite simple, yet formal, mathematical model of such affine self-similarity using L^2 norm.
2. We show that such a model has been implicitly used in a number of nonlocal image processing schemes, including:
   (a) Nonlocal-means denoising
   (b) Method of "self-examples" and "examples", in general
   (c) Fractal image coding
3. Examine effects of (additive) noise on self-similarity.
4. Assign relative degrees of self-similarity to a collection of images.

Mathematical setting

An image I will be represented by an image function $u \in B(X)$,

$$u : X \to R_g,$$

X: the support of u, e.g. $[0,1]^2$ or an $n_1 \times n_2$ pixel array,

R_g: the *greyscale range*, e.g., $R_g = [0,1]$,

$B(X)$: a suitable space of functions supported on X, e.g., $L^2(X)$, with metric

$$d(u_1, u_2) = \| u_1 - u_2 \|, \quad u_1, u_2 \in B(X).$$

8 bpp *Lena* image and associated image function $u(x, y)$

Here, $R_g = [0,1]$ (normalized images) and $\| \cdot \|$ is RMSE.

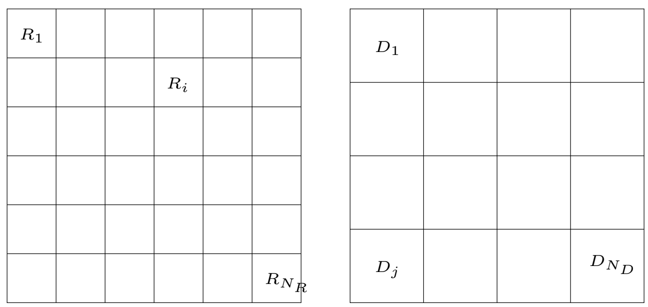
A model of affine image self-similarity

For simplicity, consider the discrete case: X is an $n_1 \times n_2$ pixel array. Then:

1. Let $\mathcal{R}$ be a set of $n \times n$-pixel **range** subblocks R_i, $1 \leq i \leq N_R$, such that $\cup_i R_i = X$. (For convenience, assume that they are nonoverlapping.)

2. Let $\mathcal{D}$ denote a set of $m \times m$-pixel **domain** subblocks D_j, $1 \leq j \leq N_D$, where $m \geq n$ and $\cup_i D_i = X$.

3. Let $w_{ij} : D_j \to R_i$ denote affine geometric transformation (along with decimation, if necessary).



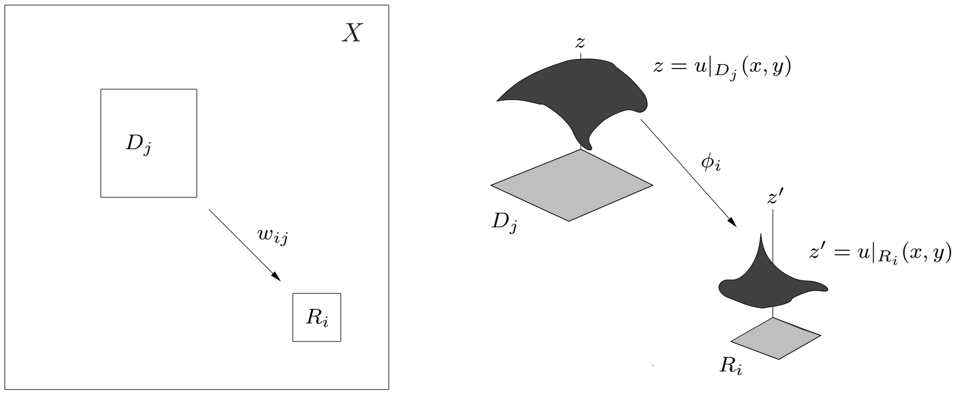
In this report, unless otherwise indicated, we shall use 8×8-pixel range blocks R_j and 8×8- or 16×16-pixel domain blocks.

In general, we may have to examine images at various scales, if new features appear at different scales.

How well are subimages $u(R_i)$ approximated by subimages $u(D_j)$?

$$u(R_i) \approx \phi_i u(w_{ij}^{-1}(R_i)), \qquad 1 \le i \le N_R,$$

where $\phi_i : \mathbf{R} \to \mathbf{R}$ is a **greyscale transformation**.



Left: Range block R_i and associated domain block D_i. **Right:** Greyscale mapping ϕ_i from $u(D_j)$ to $u(R_i)$.

Consider **affine greyscale maps**, i.e.,

$$\phi(t) = \alpha t + \beta.$$

- Simple in form, yet sufficiently flexible

Then examine the distribution of L^2 (RMS) approximation errors Δ_{ij}, $1 \le i \le N_R$, $1 \le j \le N_D$:

$$\Delta_{ij} = \inf_{\alpha,\beta \in \Pi} \| u(R_i) - \alpha u(w_{ij}^{-1}(R_i)) - \beta \|_2 \quad . \tag{1}$$

$\Pi \subset \mathbf{R}^2$: "feasible parameter space"

We consider four particular cases of self-similarity:

1. **Purely translational:** The w_{ij} are translations and $\alpha_i = 1$, $\beta_i = 0$, i.e.,

$$u(R_i) \approx u(D_j).$$

2. **Translational + greyscale shift:** The w_{ij} are translations and $\alpha_i = 1$, optimize β:

$$u(R_i) \approx u(D_j) + [\bar{u}(R_i) - \bar{u}(D_j)].$$

3. **Affine, same scale:** The w_{ij} are translations but we optimize α and β:

$$u(R_i) \approx \alpha_i u(D_j) + \beta_i.$$

4. **Affine, cross-scale:** The w_{ij} are affine spatial **contractions** (which involve decimations in pixel space).

$$u(R_i) \approx \alpha_i u(w_{ij}^{-1}(R_i)) + \beta_i. \tag{2}$$

- Cases 1-3: Domain D_j and range R_i blocks have *same size.*
- Case 4: Domain blocks D_j are **larger** than range blocks R_i.

Same-scale self-similarity – Cases 1, 2 and 3

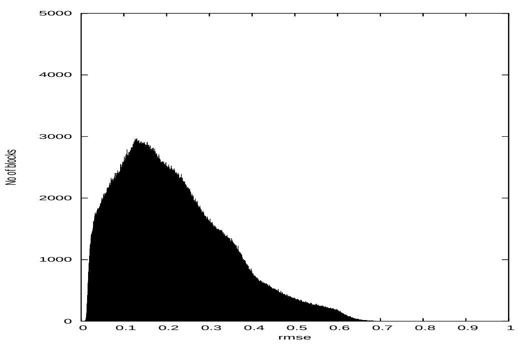
Recall:

- **Case 1**: Purely translational
- **Case 2**: Translational + greyscale shift β
- **Case 3**: Translational + affine greyscale transformation $\alpha t + \beta$.

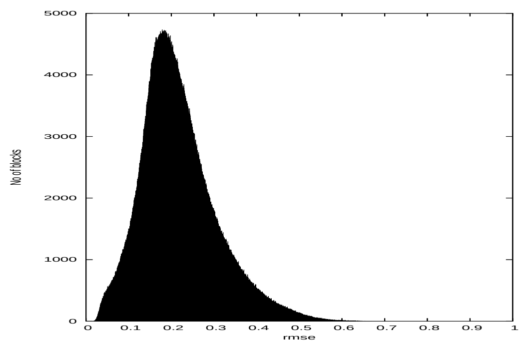
We expect that

$$\Delta_{ij}^{(Case\ 1)} \geq \Delta_{ij}^{(Case\ 2)} \geq \Delta_{ij}^{(Case\ 3)}.$$

Case 1



(a) Lena



(b) Mandrill

Case 1 (same-scale) self-similarity error distributions $\Delta_{ij}^{(Case\ 1)} = \|u(R_j) - u(R_i)\|, \quad i \neq j$, for normalized 512×512-pixel *Lena* and *Mandrill* images. In all cases, 8×8-pixel blocks $R_i = D_i$ were used.

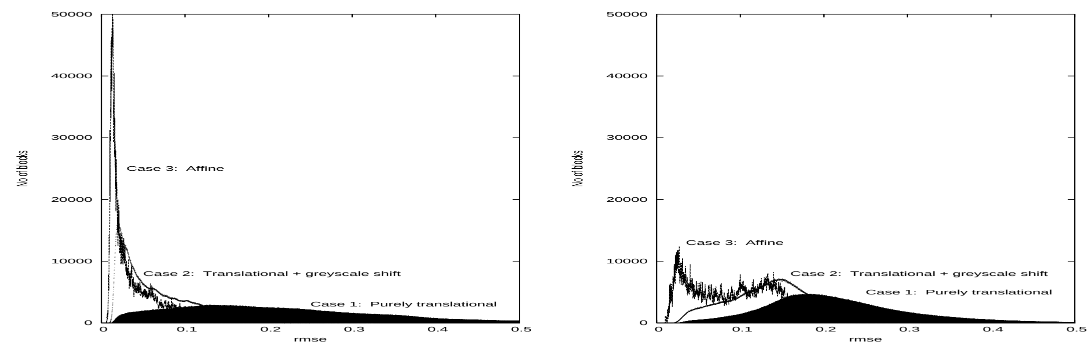
Same-scale self-similarity – Cases 1, 2 and 3

Recall:

- **Case 1**: Purely translational
- **Case 2**: Translational + greyscale shift β
- **Case 3**: Translational + affine greyscale transformation $\alpha t + \beta$.

We expect that

$$\Delta_{ij}^{(Case\ 1)} \geq \Delta_{ij}^{(Case\ 2)} \geq \Delta_{ij}^{(Case\ 3)}.$$

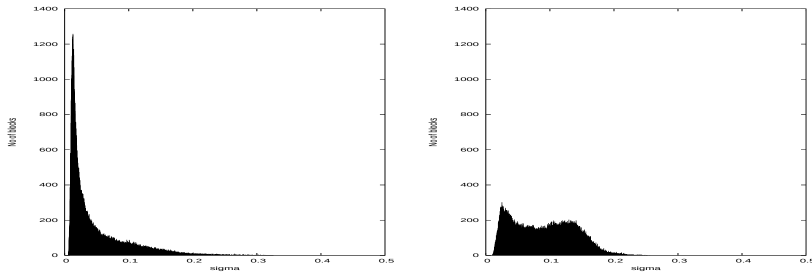


(a) Lena (b) Mandrill

Same-scale (Cases 1,2 and 3) RMS self-similarity error distributions for normalized *Lena* and *Mandrill* images.

Again, 8×8-pixel blocks $R_i = D_i$ were used. Case 1 distributions are shaded.

$\Delta^{(Case\ 3)}$-error and $\sigma(u(R_i))$ distributions are similar



(a) Lena (b) Mandrill

Distributions of $\sigma(u(R_i))$ of 8×8-pixel blocks over the interval $[0, 0.5]$.

$\sigma(R_i)$ is the RMS error of approximation

$$u(R_i) \approx \bar{u}(R_i) \qquad \text{(best } L^2 \text{ fit with a constant).}$$

This corresponds to "clamping" the greyscale parameter, $\alpha = 0$, and optimizing over β.

But in Case 3 we optimize over both α and β. Therefore:

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

Distributions of α parameters peak at zero (later slide). Therefore we expect $\Delta^{(Case\ 3)}$ distribution to be slight perturbation of $\sigma(u(R_i))$ distribution toward zero error.

Extreme example of a perfectly self-similar image

The "flat" image:

$$u = C \qquad \text{(constant)}$$

$\Delta^{(Case\ q)}$-error distributions have single peaks at $\Delta = 0$, for $q = 1, 2, 3$.

Next on the list:

"Ramped" images:

$$u = C + Ax + By$$

Translational symmetry (Case 1) is employed in

"A nonlocal algorithm for image denoising," A. Buades, B. Coll and J.-M. Morel, CVPR (2), 60-65 (2005); Multiscale Mod. Sim. **4**, 490-530 (2005).

"NL-means algorithm: Given noisy image $v = \{v(i), i \in I\}$, replace each pixel value $v(i)$ by estimated value $NL[v](i)$ where

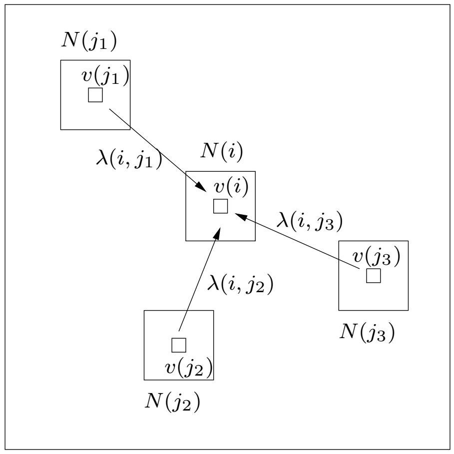
$$NL[v](i) = \sum_{j \in I} \lambda(i,j) v(j).$$

Weights $\lambda(i,j)$ depend upon the "similarity" of greyscale pixel blocks $v(N_i)$ and $v(N_j)$:

$$\lambda(i,j) = \frac{1}{Z(i)} e^{-A\|v(N_i)-v(N_j)\|^2}.$$

1. N_k denotes a square neighbourhood of fixed size and centered at pixel k.
2. $A > 0$: a constant (related to filtering parameter) and
3. $Z(i)$: normalization constant.

Basic idea: Averaging over noisy samples reduces variance of (additive) noise.



$$NL[v](i) = \frac{1}{Z(i)} \sum_{j \in I} e^{-A\|v(N_i)-v(N_j)\|^2} v(j).$$

Method works surprisingly well, even though neighbourhood-matching requirement is quite restrictive. A computationally inexpensive improvement is obtained when greyscale shifts are allowed:

$$NL[v](i) = \frac{1}{Z(i)} \sum_{j \in I} e^{-A\|v(N_i)-v(N_j)-\beta_j\|^2} [v(j) + \beta_j],$$

where

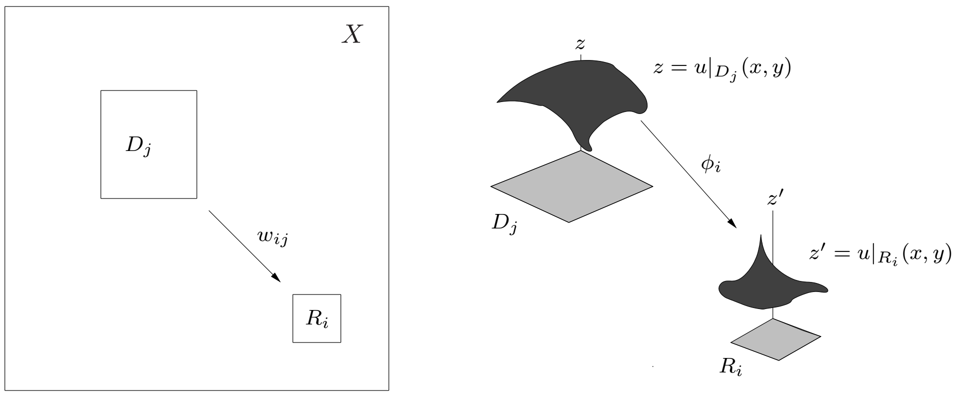
$$\beta_j = \bar{N}_i - \bar{N}_j.$$

Cross-scale self-similarity (Case 4)

Recall

$$u(R_i) \approx \alpha_i u(w_{ij}^{-1}(R_i)) + \beta_i.$$

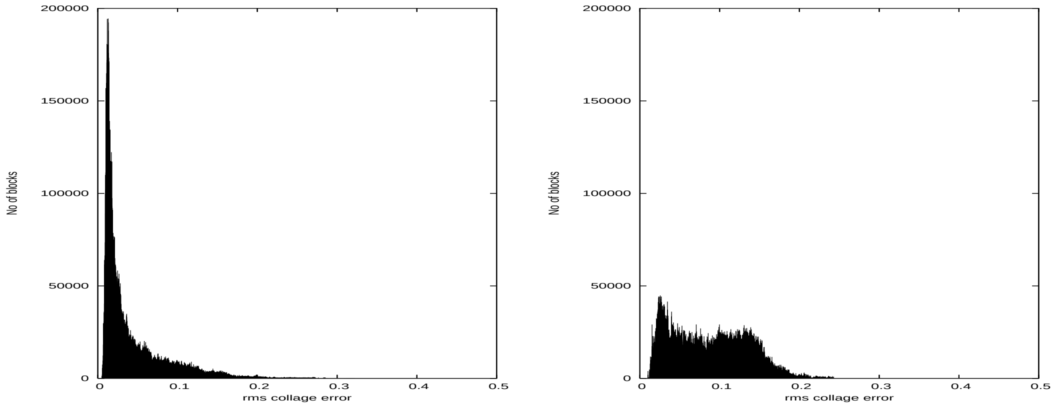
The $w_{ij} : D_j \to R_i$ are **spatial contractions**, mapping **larger** domain blocks onto **smaller** range blocks.



Left: Range block R_i and larger domain block D_i. **Right:** Greyscale mapping ϕ_i from $u(D_j)$ to $u(R_i)$.

Cross-scale self-similarity approximation errors

$$\Delta_{ij} = \inf_{\alpha,\beta \in \Pi} \parallel u(R_i) - \alpha u(w_{ij}^{-1}(R_i)) - \beta \parallel_2 \quad .$$



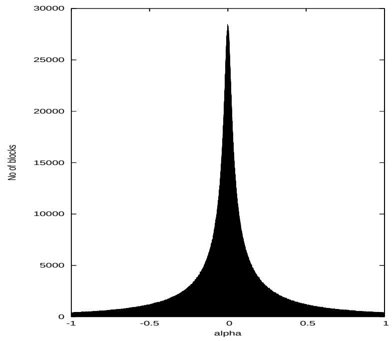
(a) Lena (b) Mandrill

Histograms of approximation errors Δ_{ij}, normalized *Lena* and *Mandrill* images: 8×8-pixel range blocks R_i, 16×16-pixel nonoverlapping domain blocks D_j. All possible domain-range pairs considered along with eight spatial mappings: 33,554,432 comparisons.

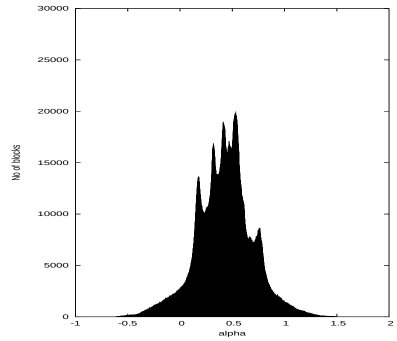
Qualitatively similar results for

- all possible 8×8 range blocks R_i obtained by single-pixel shifts,
- 16×16-pixel range blocks R_i, etc..
- Same-scale, affine (Case 3), $R_i = D_i$.

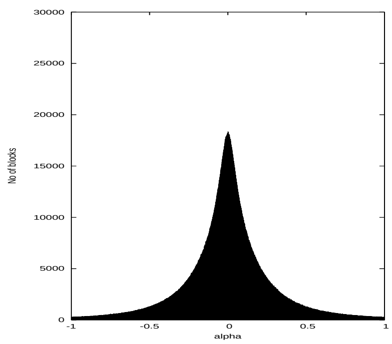
α and β greyscale parameter distributions ($\phi(t) = \alpha t + \beta$)



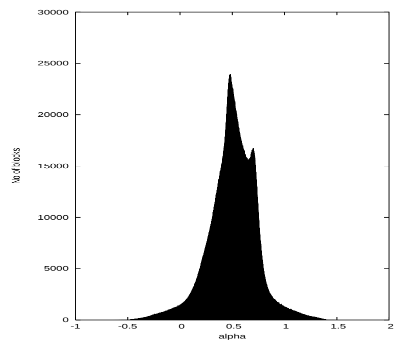
(a) Lena α



(b) Lena β

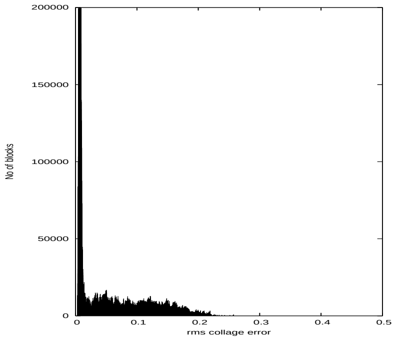


(c) *Mandrill* α

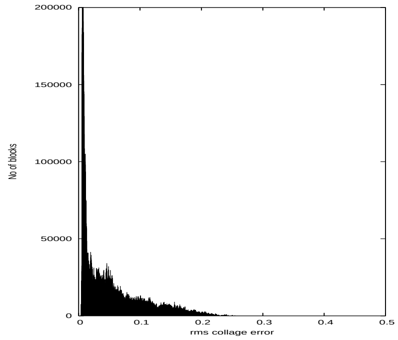


(d) *Mandrill* β

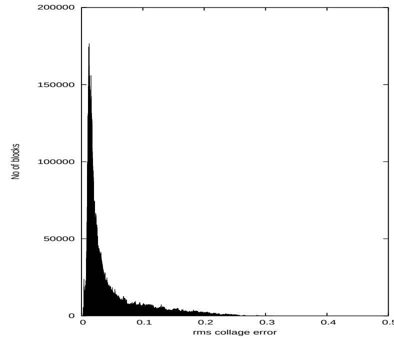
Cross-scale self-similarity approximation errors Δ_{ij} for other images



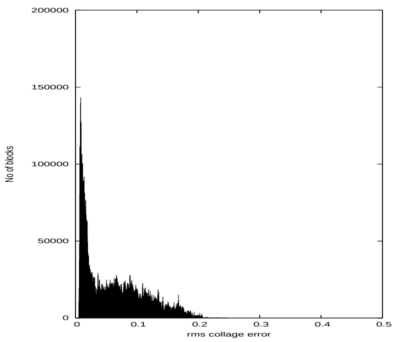
(a) San Francisco



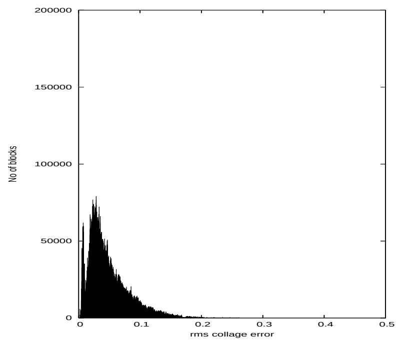
(b) Boat



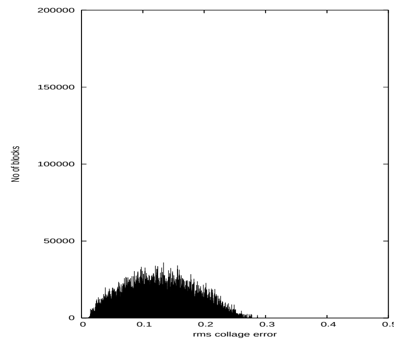
(c) Peppers



(d) Barbara



(e) Goldhill



(f) Zelda

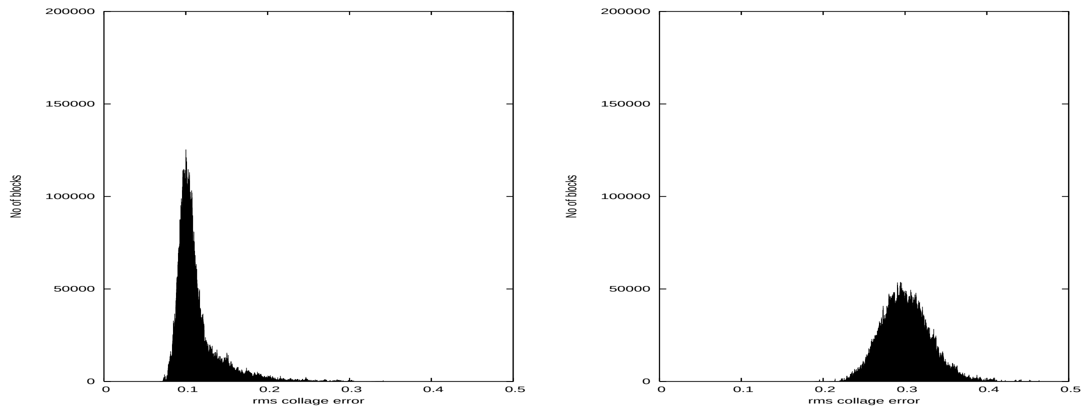
Special case: Constant image $u = C$ is perfectly self-similar (Cases 1-4).

$$\Delta_{ij} = 0 \quad \forall i, j.$$

Histogram has single peak at $\Delta = 0$ and is zero everywhere else.

Effects of noise

As noise of increasing variance σ is added to an image, the distribution of approximation errors Δ_{ij} moves outward and spreads.

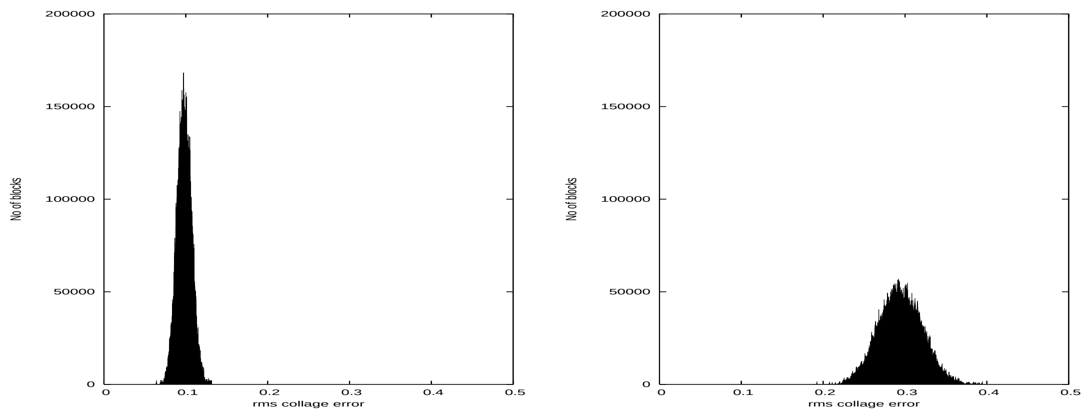


(a) Lena + noise ($\sigma = 0.1$)

(b) Lena + noise ($\sigma = 0.3$)

Distributions of Δ_{ij} approximation errors for two cases of *Lena* + zero-mean, Gaussian noise. The peaks of these distributions lie roughly at the σ value of the noise.

Pure noise



(a) $\sigma = 0.1$ (b) $\sigma = 0.3$

Distribution of Δ_{ij} approximation errors for pure noise images, $u = 0.5 + \mathcal{N}(0, \sigma)$. 512×512-pixel images, 8×8-pixel range blocks. The peaks of these distributions lie at the σ value of the noise.

(This is the basis of standard block-based estimation of noise STD found in image processing textbooks.)

Can we use these distributions to assign relative self-similarity?

Are *Lena* and *San Francisco* more self-similar than *Mandrill* and *Zelda*?

Are the latter more noise-like?

Examine the means and variances of the Δ_{ij}-distributions...

Characterizing self-similarity quantitatively from Δ_{ij} distributions

Image	Δ_{ij}			$\sigma(R_j)$	
	mean	stddev	entropy	mean	stddev
Lena	0.043	0.044	2.26	0.046	0.046
San Francisco	0.046	0.057	2.01	0.048	0.059
Peppers	0.047	0.050	2.32	0.049	0.052
Goldhill	0.049	0.034	2.46	0.052	0.036
Boat	0.052	0.052	2.58	0.055	0.055
Barbara	0.060	0.049	2.69	0.064	0.051
Mandrill	0.089	0.048	2.85	0.089	0.048
Zelda	0.126	0.055	3.09	0.141	0.054

Columns 1-3: Means, standard deviations, and entropies of Δ_{ij} distributions for a number of test images. Columns 4 and 5: Means and standard deviations of σ-distributions of these images, to show their agreement with Columns 1 and 2, respectively.

This may provide a partial answer to the question

"In what way do natural images differ from random images?"

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

Cross-scale self-similarity is the basis of fractal image coding

Recall

$$u(R_i) \approx \alpha_i u(w_{ij}^{-1}(R_i)) + \beta_i.$$

The $w_{ij} : D_j \to R_i$ are **spatial contractions**, mapping **larger** domain blocks onto **smaller** range blocks.

Standard block-based fractal image coding:

For each range block R_i, choose the domain block $D_{j(i)}$ that yields the **lowest** approximation error Δ_{ij}.

The range-domain assignments $(i, j(i))$ along with associated greyscale map parameters (α_i, β_i) define a **fractal transform** operator T:

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

The fractal transform is a *nonlocal* operator

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

- The image function u is approximated by a union of **spatially contracted** and **greyscale modified** copies of its subblocks.
- For this reason, fractal coding has often been referred to as **"self-VQ"**
- perhaps more aptly as **"self-structured VQ using linear transforms"**, cf. C.O. Etemoglu and V. Cuperman, IEEE Trans. Sig. Proc. **51**, 1625-1631 (2003).

Under suitable conditions on α_i and C_{ij} (contractivity factors of w_{ij}), T is a **contractive operator** on the space $B(X)$. Therefore,

Banach Contraction Mapping Theorem (1922)

There exists a unique $\bar{u} \in B(X)$ such that

1. $T\bar{u} = \bar{u}$
2. For any $u_0 \in B(X)$, define the sequence $u_{n+1} = Tu_n$, $n = 0, 1, 2, \cdots$. Then

$$\| u_n - u \| \to 0 \quad \text{as} \quad n \to \infty. \tag{4}$$

T possesses a unique and globally attractive fixed point $\bar{u}$.

A simple consequence of Banach's theorem is the "Collage Theorem":

$$\| u - \bar{u} \| \;\; \leq \; \frac{1}{1-c} \, \| u - Tu \|,$$

where c is the contraction factor of T.

Inverse problem of fractal image coding: Given a "target" image u, we try to find an operator T (in terms of domain-range assignments) that makes the approximation error $\| u - Tu \|$ as small as possible, so that u is well approximated by $\bar{u}$.

Why? Because we can store parameters that define T and then generate $\bar{u}$.

This was a major motivation for work in fractal image coding/compression initiated by M. Barnsley and students/coworkers at Georgia Tech in late 1980's.

Iteration of fractal transform operator

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, 8bpp *Lena* image. 8×8-pixel nonoverlapping range blocks. 16×16-pixel nonoverlapping domain blocks. The “seed” image was $u_0(x) = 255$ (plain white).

Further insights into fractal image coding

- From cross-scale Δ_{ij} (approximation error) distributions, we conclude that an image range subblock $u(R_i)$ is generally well approximated by a number of (decimated) domain subblocks $u(D_j)$.
- Traditional fractal coding research, which was concerned with *compression*, usually focussed on using the **best** domain block.
- But a number of other range blocks will also do very well. This suggests:

Fractal image coding with multiple parent blocks

Each image range block $u(R_i)$ is expressed as a weighted sum of spatially-contracted and greyscale modified copies of a number of image domain blocks $u(D_{ij})$:

$$u(x) \approx (Tu)(x) = \sum_j \lambda_{ij} \left[\alpha_{ij} u(w_{ij}^{-1}(x)) + \beta_{ij} \right], \quad x \in R_i, \quad 1 \leq i \leq N_R,$$

where

$$\sum_j \lambda_{ij} = 1.$$

Recent applications: Obviously not compression! But rather

- image denoising (S.K. Alexander, Ph.D. Thesis, Waterloo, 2005) – a cross-scale version of NL-means denoising


(a) Lena original (b) + $\mathcal{N}(0.01, 0)$, PSNR = 20.05


(c) Single parent, PSNR = 26.67 (d) Multiparent, PSNR = 29.03

Denoising of *Lena* image: single parent (traditional fractal coding) and multiparent fractal coding (10 parents).

Recent applications (cont'd):

- image zooming (pixel domain) "using examples" (M. Ebrahimi)
  - look for larger domain blocks that provide good fits - use these to construct higher resolution image
- super-resolution in the frequency domain, with particular application to MRI (G. Mayer)
  - block-based coding of the raw frequency signal, to (i) extrapolate, (ii) denoise

These are **nonlocal** methods of super-resolution/denoising.

"Nonlocal image processing" has recently become a subject of great interest:

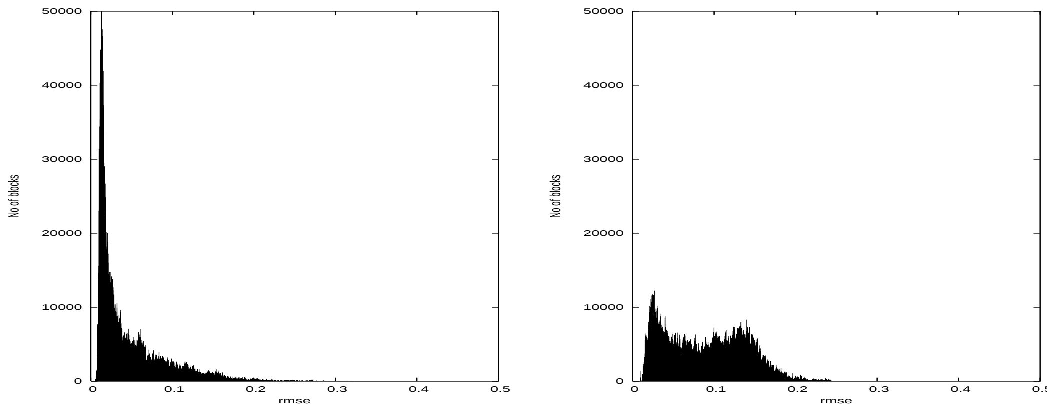
"LNLA 2008:" Workshop on Local and Nonlocal Approximation in Images,

August 23-24, 2008, Lausanne, Switzerland

(Prelude to EUSIPCO 2008)

Approximating subblocks of one image with those of another

Same scale:



(a) Approximating *Lena* with *Mandrill*

(b) Approximating *Mandrill* with *Lena*

Figure 1: Error distributions associated with approximating 8×8-pixel subblocks of image A by affine transformations of 8×8-pixel subblocks of image B.

Same qualitative behaviour for *cross-scale* approximation, i.e., fractal coding.

“Self-similarity” vs. “Approximability”?

Are images self-similar or are they simply approximable?

We’ll simply say that **images that are approximable are also self-similar** – a range block R_j is generally well approximated by many domain blocks D_{j_i}.

This property may be exploited for purposes of **denoising** and **example-based zooming**.

This is the spirit of **nonlocal, data-driven methods**.

Possible objection: The computational price – the **search times** to find good domain blocks.

This is offset to a good degree by **restricted searching** – the methods still work very well.

Other avenues and recent progress

1. **"Going beyond L^2":** Exploring the use of other **similarity measures** to characterize self-simililarity.

   - S.K. Alexander, S. Kovačič and E.R. Vrscay, "A simple model for image self-similarity and the possible use of mutual information," EUSIPCO 07.

2. **Measure-valued images:** Associated with each pixel is a measure/distribution of greyscale-modified values from all other parts of the image.

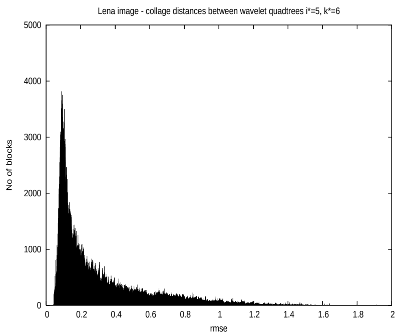
   - A kind of **preprocessing** before the final projection to a single value $u(x)$.
   - Possible characterization of **pointwise self-similarity properties** of an image.

   (D. La Torre, E.R.V., M. Ebrahimi and M.F. Barnsley, to appear in *SIAM Journal of Imaging Science*)

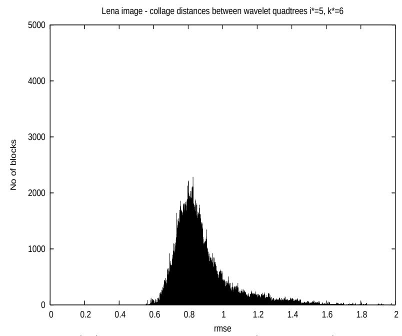
3. **Self-similarity in wavelet domain:** Wavelet coefficient subtrees demonstrate both same-scale (Cases 1,2) and cross-scale similarity (Case 4, with $\beta = 0$):

Cross-scale affine self-similarity of wavelet sub-quadtrees

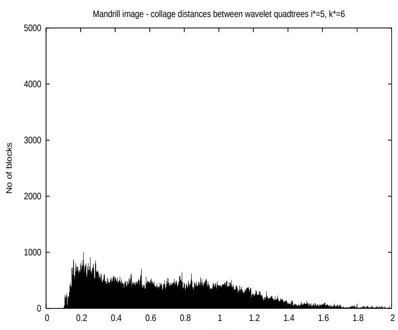
$$B_{k_1+1,i,j} = \alpha B_{k_1,i',j'}.$$



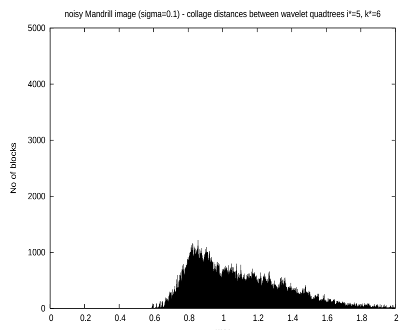
(a) Lena



(b) Lena + noise ($\sigma = 0.1$)



(c) Mandrill



(d) Mandrill + noise ($\sigma = 0.1$)

Distributions of Δ_{ij} approximation errors for cross-scale affine self-similarity.

This is the basis for **fractal-wavelet image coding**.