Hyperspectral images as function-valued mappings, their self-similarity and a class of fractal transforms

E.R. Vrscay\textsuperscript{1}  D. Otero\textsuperscript{1}  Davide La Torre\textsuperscript{2}

Department of Applied Mathematics, Faculty of Mathematics, University of Waterloo, Waterloo, ON, Canada

Department of Economics, Business and Statistics, University of Milan, Milan Italy

ervrscay@uwaterloo.ca, dotero@uwaterloo.ca, davide.latorre@unimi.it

1 Introduction

2 A complete metric space \((Y, d_Y)\) of function-valued images

3 Self-similarity of greyscale images

4 Self-similarity of hyperspectral images

5 A class of block fractal transforms on hyperspectral images
Outline

1. Introduction
2. A complete metric space \((Y, d_Y)\) of function-valued images
3. Self-similarity of greyscale images
4. Self-similarity of hyperspectral images
5. A class of block fractal transforms on hyperspectral images
This study represents ongoing work on the development of multifunction representations of images, in particular,

- **Measure-valued image functions:**

- **Function-valued image functions:**
Our work is involved with generalizations of the usual mathematical representation of a (greyscale/colour) image, i.e.,
\[ u : X \rightarrow R_g, \]
where
- \( X \): base or pixel space, the support of the image, \( X \subset \mathbb{R}^n, n = 1, 2, 3, \)
- \( R_g \subset \mathbb{R} \) (or \( \mathbb{R}^3 \)): the greyscale (or colour) range.
Representations of image functions

**Greyscale-valued image function**
At each pixel $x \in X$, $u(x)$ is a **real value** (or a vector of real values, i.e., “RGB”)

**Function-valued image function**
At each pixel $x \in X$, $u(x)$ is a **real-valued function**, i.e., $u(x; t)$

**Example:** In multispectral/hyperspectral imaging, $u$ represents the **spectral density function**. The values $u(x, t_k)$, $t_1 < t_2 < \cdots < t_M$ represent intensities of reflected radiation from point $x$ on ground, as captured by satellite reading, at a discrete set of wavelengths, $t_k$. 

---

**Diagram:**
- **(a) Greyscale-valued image function**
- **(b) Function-valued image function**
Hyperspectral imaging
In practical situations, multispectral/hyperspectral images may be represented by vector-valued functions,

\[ u : X \rightarrow \mathbb{R}^M, \]

e.g.,

\[ u(x) = (u_1(x), u_2(x), \cdots, u_M(x)), \]

where

\[ u_k : X \rightarrow \mathbb{R}, \quad 1 \leq k \leq M \]

are the usual real-valued image functions. (Of course, RGB images are special, low-dimensional, cases.)

That being said, it is instructive to start with the continuous, multifunction approach, from which definitions over vector-valued image functions naturally follow.
Introduction

A complete metric space $(Y, d_Y)$ of function-valued images

Self-similarity of greyscale images

Self-similarity of hyperspectral images

A class of block fractal transforms on hyperspectral images
Introduction A complete metric space \((Y, d_Y)\) of function-valued images

Self-similarity of greyscale images Self-similarity of hyperspectral images A class of block fractal transforms on hyperspectral images

\[ x(t), u(x, t) \]

\[ y(x, t), v(x, t) \]

(a) (b)

Linear space: For \( u, v : X \rightarrow L^2(R_{g}) \), define

\[(c_1 u + c_2 v)(x, t) = c_1 u(x, t) + c_2 v(x, t), \quad \text{etc. (linear space)}\]

Normed linear space \( Y \): For \( u : X \rightarrow L^2(R_{g}) \), norm of \( u(x) \) is given by

\[ \| u(x) \|_{L^2(R_{g})}^2 = \int_{R_{g}} u(x, t)^2 \, dt. \]

Integrate over all \( x \in X \) to define norm of \( u \):

\[ \| u \|_{Y}^2 = \int_X \| u(x) \|_{L^2(R_{g})}^2 \, dx. \]
**Complete metric space** \((Y, d_Y)\): 

1. At each \(x \in X\), compute \(L^2\) distance between functions \(u(x)\) and \(v(x)\): 
   
   \[
   \|u(x) - v(x)\|^2_{L^2(R_g)} = \int_{R_g} [u(x, t) - v(x, t)]^2 \, dt 
   \]

2. Integrate over all \(x \in X\): 
   
   \[
   \left[ d_Y(u, v) \right]^2 = \int_X \|u(x) - v(x)\|^2_{L^2(R_g)} \, dx. 
   \]
Hilbert space:
Since $u(x), v(x) \in L^2(R_g)$, we may compute their inner product $\langle u(x), v(x) \rangle_{L^2(R_g)}$. Integrate over all $x \in X$ to define inner product between function-valued image mappings,

$$\langle u, v \rangle_Y = \int_X \langle u(x), v(x) \rangle_{L^2(R_g)} \, dx, \quad u, v \in Y.$$
A complete metric space \((Y, d_Y)\) of function-valued images

Self-similarity of greyscale images  
Self-similarity of hyperspectral images  
A class of block fractal transforms on hyperspectral images

**Complete metric space \((Y, d_Y)\) of function-valued image mappings**

\[ Y = \{ u : X \rightarrow L^2(R_g) \mid \|u\|_Y < \infty \} \]

where

\[ \|u\|^2_Y = \int_X \|u(x)\|^2_{L^2(R_g)} \, dx \]

In our applications,

\[ R_g = [a, b] \subset R_+ = [0, \infty). \]
Introduction

A complete metric space \((Y, d_Y)\) of function-valued images

Self-similarity of greyscale images

Self-similarity of hyperspectral images

A class of block fractal transforms on hyperspectral images

Outline

1. Introduction
2. A complete metric space \((Y, d_Y)\) of function-valued images
3. Self-similarity of greyscale images
4. Self-similarity of hyperspectral images
5. A class of block fractal transforms on hyperspectral images
Self-similarity of greyscale images


It was shown that images generally possess a considerable degree of **affine self-similarity**, i.e.,

Subblocks of an image are well approximated by a number of other subblocks – with the possible help of affine greyscale transformations

Self-similarity of images has been implicitly used in a number of **nonlocal image processing schemes**, including

- Yes, vector quantization! (Fractal image coding is, in fact, “self-vector quantization”.)
A simple 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 $\bigcup_i R_i = X$. (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 $\bigcup_i D_i = X$.

3. Let $w_{ij}: D_j \rightarrow R_i$ denote affine geometric transformation (along with decimation, if necessary). There are 8 possible mappings of squares to squares - here we consider only one (no rotation/flipping).

In ICIAR08 study, 8 × 8-pixel range blocks $R_j$ and 8 × 8- or 16 × 16-pixel domain blocks were used.
How well are subimages $u(R_i)$ approximated by subimages $u(D_j)$?

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

where $\phi_i : \mathbb{R} \rightarrow \mathbb{R}$ is a greyscale transformation.

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

---

Introduction A complete metric space $(Y, d_Y)$ of function-valued images Self-similarity of greyscale images Self-similarity of hyperspectral images A class of block fractal transforms on hyperspectral images
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 \( \Delta_{ij} \), \( 1 \leq i \leq N_R \), \( 1 \leq j \leq N_D \):

\[ \Delta_{ij} = \| u(R_i) - \phi(u(w_{ij}^{-1}(R_i))) \|_2 \]

Note that all images are assumed to be **normalized**, i.e., \( 0 \leq u_{pq} \leq 1 \), so that

\[ 0 \leq \Delta_{ij} \leq 1 \]
Four particular cases of self-similarity considered:

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

   \[ u(R_i) \approx u(D_j) \]

   Employed in nonlocal means denoising

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

   \[ u(R_i) \approx u(D_j) + [u(R_i) - u(D_j)] \]

3. **Case 3 (Affine, same scale):** The $w_{ij}$ are translations but we optimize $\alpha$ and $\beta$:

   \[ u(R_i) \approx \alpha_i u(D_j) + \beta_i \]

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

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

   Employed in fractal image coding
Same-scale self-similarity – Cases 1, 2 and 3

Recall:

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

We expect that

$$0 \leq \Delta_{ij}^{(\text{Case 3})} \leq \Delta_{ij}^{(\text{Case 2})} \leq \Delta_{ij}^{(\text{Case 1})}$$
“World’s most self-similar image”

The “flat” image,

\[
u(x, y) = C \quad \text{(constant)}
\]

\(\Delta^{(\text{Case } q)}\)-error distributions have single peaks at \(\Delta = 0\), for \(q = 1, 2, 3\) and 4.

Next on the list:

“Ramped” images,

\[
u(x, y) = C + Ax + By
\]

\(\Delta^{(\text{Case } q)}\)-error distributions have single peaks at \(\Delta = 0\), for \(q = 2, 3\) and 4.

And now on to more realistic images ...
Case 1 (Purely translational)

Case 1 (same-scale) self-similarity error distributions

\[ \Delta_{ij}^{(\text{Case 1})} = \| u(R_j) - u(R_i) \|_2, \quad i \neq j, \]

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

(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.
Introduction

A complete metric space \((Y, d_Y)\) of function-valued images

Self-similarity of greyscale images

Self-similarity of hyperspectral images

A class of block fractal transforms on hyperspectral images
Assume that digital hyperspectral image is supported on an $N_1 \times N_2$ pixel array, as before, but now $M$ channels per pixel.

At a pixel location $(i_1, i_2) \in X$, the hyperspectral image function is a non-negative $M$-vector with components

$$u_k(i_1, i_2), \quad 1 \leq k \leq M.$$
Also as before:

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

![Diagram of 3D block fractal transform](image-url)
Let $u(R_i)$ denote portion of hyperspectral image function supported on subblock $R_i \in X$. Here, $u(R_i)$ will be an $n \times n \times M$ cube of nonnegative real numbers.

The $L^2$ (RMS) distance, $\Delta_{ij}$, between two hyperspectral image subblocks $u(R_i)$ and $u(R_j)$ will be given by

$$\Delta_{ij} = \frac{1}{n\sqrt{M}} \left[ \sum_{i_1=I_1}^{I_1+n-1} \sum_{i_2=I_2}^{I_2+n-1} \sum_{k=1}^{M} [u_k(i_1, i_2, \ldots) - u_k(i_1 + J_1, i_2 + J_2)]^2 \right]^{1/2}$$

This may also be viewed as the error associated with the (Case 1) approximation,

$$u(R_i) \approx u(R_j) \quad \text{(Case 1)}$$
Case 2 approximations with spectral shifts

- Simplest case - the same shift, $\beta \in \mathbb{R}$, for all channels
  \[ u(R_i) \approx u(R_j) + \beta, \quad \text{(Case 2(a))} \]
  This does not improve the Case 1 approximation significantly.
- Separate shift, $\beta_k$, for each channel
  \[ u(R_i) \approx u(R_j) + \beta, \quad \text{(Case 2(b))} \]

Componentwise,
\[ u_k(i_1, i_2) \approx u_k(j_1, j_2) + \beta_k, \quad 1 \leq k \leq M \]
Case 3 approximation with affine scaling + spectral shift

\[ u(R_i) \approx \alpha u(R_j) + \beta \quad \text{(Case 3)} \]

Note that we are using only **one** scaling coefficient \( \alpha \) for all channels.

**Note:** If we used separate scaling coefficients for each channel \( k \), i.e.,

\[ u_k(R_i) \approx \alpha_k u(R_j) + \beta_k, \quad 1 \leq k \leq M, \]

then we are essentially treating a hyperspectral image as \( M \) separate greyscale images (which defeats the purpose of hyperspectral image analysis).

**Approximation errors:**

\[ 0 \leq \Delta_{ij}^{(\text{Case 3})} \leq \Delta_{ij}^{(\text{Case 2(b)})} \leq \Delta_{ij}^{(\text{Case 2(a)})} \leq \Delta_{ij}^{(\text{Case 1})} \]
Results of some computations

33-channel hyperspectral image, “Scene 2,” downloaded from webpage of D.H. Foster, University of Manchester

Per-pixel error distributions $\Delta_{ij}^{(Case \ q)}$ for 33-channel HS fern image. In all cases, 8 x 8-pixel blocks $R_i$ and $D_j$ were used.
224-channel AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) hyperspectral image, “Yellowstone calibrated scene 0,” a 224-channel image, available from JPL.

Per-pixel error distributions $\Delta_{ij}^{(Case\ m)}$ for the 224-channel AVIRIS image. In all cases, $8 \times 8$-pixel blocks $R_i$ and $D_j$ were used.
Because of the additional degree of freedom along the spectral axis, we may consider $n \times n$-pixel blocks as $n \to 1$, in particular, $n = 1$.

Case 1 error distributions $\Delta_{ij}^{(\text{Case 1})}$ for spectral functions supported on single-pixel blocks $R_i$.

However, $L^2$ distance (RMSE) is not necessarily a good indicator of signal/image fidelity or correlation.
A number of alternative quality indices exist, e.g., “structural similarity.” Here, however, we examine simple correlation $C(x, y)$ between spectral functions $x, y \in \mathbb{R}^M$.

The dramatic correlation demonstrated in these plots strongly suggests that single-pixel spectral functions are quite suitable for nonlocal methods of image processing.
Introduction

A complete metric space \((Y, d_Y)\) of function-valued images

Self-similarity of greyscale images

Self-similarity of hyperspectral images

A class of block fractal transforms on hyperspectral images
In fractal image coding of greyscale images:

1. Affine greyscale transformations are employed, i.e.: \( \phi(t) = \alpha t + \beta \).
2. Domain blocks \( D_j \) are \textbf{larger} than range blocks \( R_j \).

As before, 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 \( \bigcup_i R_i = X \). (For convenience, assume that they are nonoverlapping.)
2. Let \( \mathcal{D} \) denote a set of \( 2n \times 2n \)-pixel domain subblocks \( D_j \), \( 1 \leq j \leq N_D \), where \( m \geq n \) and \( \bigcup_i D_i = X \).
3. Let \( w_{ij} : D_j \rightarrow R_i \) denote affine geometric \textbf{contraction mapping} - in pixel domain this is accomplished by some kind of decimation/downsampling.
Fractal transform of greyscale image

For $1 \leq i \leq N_R$, approximate $u(R_i)$ with greyscale modified and spatially contracted (decimated) copy of $u(D_{j(i)})$:

$$u(R_i) \approx \alpha_i u(D_{j(i)})' + \beta_i \tag{Case 4}$$

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

$$=: (Tu)(R_i), \quad 1 \leq i \leq N_R.$$

$T$ is fractal transform operator. (Prime denotes spatial contraction/pixel decimation.)
Fractal transform of hyperspectral image

For $1 \leq i \leq N_R$, approximate the “data cube” $u(R_i)$ with greyscale modified and spatially contracted (decimated) copy of “data cube” $u(D_j(i))$:

$$u(R_i) \approx \alpha_i u(D_j(i))' + \beta_i$$  \hspace{1cm} (Case 4)

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

$$=: (Tu)(R_i), \quad 1 \leq i \leq N_R.$$  

$T$ is fractal transform operator. (Prime denotes spatial contraction/pixel decimation.)

**Note:** As in Case 3 approximations of hyperspectral images, we employ one scaling coefficient $\alpha$ and a vector of shift coefficients $\beta_i$. 
Under appropriate conditions on $\alpha_i$, the hyperspectral fractal transform operator $T$ is \textbf{contractive} on the metric space $(Y, d_Y)$ of hyperspectral images.

From Banach’s Fixed Point Theorem, there exists a unique $\bar{u} \in Y$ such that

$$\bar{u} = T\bar{u}.$$

Furthermore,

For any “seed” image $u_0 \in Y$, if we define the iteration procedure,

$$u_{n+1} = Tu_n,$$

then

$$d_Y(u_n, \bar{u}) \to 0 \quad \text{as} \quad n \to \infty.$$
Inverse problem for hyperspectral fractal transforms on \((Y, d_Y)\)

Given a target element (hyperspectral image) \(u \in Y\), find a contractive fractal transform \(T : Y \to Y\) such that its fixed point \(\tilde{u}\) approximates \(u\) to a desired accuracy, i.e.,

\[d_Y(\tilde{u}, u) < \epsilon.\]

Such a fractal transform \(T\) will be defined by

1. The range block-domain block assignments \((i, j(i)), 1 \leq i \leq N_R,\)
2. The scaling coefficients \(a_i\) and \(\beta_i, 1 \leq i \leq N_R.\)

- The hyperspectral image \(u\) has been approximated by the fixed point \(\tilde{u}\) of the contractive fractal transform operator \(T\).
- The fixed point \(\tilde{u}\) may be generated by iteration of \(T\).

**Result:** The hyperspectral image \(u\) has been **fractally coded**.
Most, if not all, fractal image coding methods rely on a simple consequence of Banach’s Fixed Point Theorem, known as the Collage Theorem.

Given a contraction mapping \( T : Y \rightarrow Y \) with contraction factor \( c_T \in [0, 1) \) and fixed point \( \bar{u} \), then for any \( u \in Y \),

\[
\| u - \bar{u} \| \leq \frac{1}{1 - c_T} \| u - Tu \|
\]

In order to approximate the target \( u \) with a fixed point \( \bar{u} \), we look for a transform \( T \) that maps the target \( u \) as close as possible to itself, i.e., we minimize the **collage distance** \( \| u - Tu \| \).

This is accomplished by finding, for each range block \( u(R_i) \), the domain block \( u(D_{j(i)}) \) that **best approximates** \( u(R_i) \), i.e., minimizes the approximation error \( \Delta_{ij} \).
Example: Fractal coding of 224-channel AVIRIS “Yellowstone” image

Channel 120. Left: Original. Right: Fractal-based approximation.
8 × 8-pixel range blocks and 16 × 16-domain blocks.
Example: Fractal coding of 224-channel AVIRIS “Yellowstone” image

8 × 8-pixel range blocks and 16 × 16-domain blocks.