Self-Similarity of Images and Non-local Image Processing

by

Devin Glew

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Mathematics
in
Applied Mathematics

Waterloo, Ontario, Canada, 2011

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis has two related goals: the first involves the concept of self-similarity of images. Image self-similarity is important because it forms the basis for many imaging techniques such as non-local means denoising and fractal image coding. Research so far has been focused largely on self-similarity in the pixel domain. That is, examining how well different regions in an image mimic each other. Also, most works so far concerning self-similarity have utilized only the mean squared error (MSE).

In this thesis, self-similarity is examined in terms of the pixel and wavelet representations of images. In each of these domains, two ways of measuring similarity are considered: the MSE and a relatively new measurement of image fidelity called the Structural Similarity (SSIM) Index. We show that the MSE and SSIM Index give very different answers to the question of how self-similar images really are.

The second goal of this thesis involves non-local image processing. First, a generalization of the well known non-local means denoising algorithm is proposed and examined. The groundwork for this generalization is set by the aforementioned results on image self-similarity with respect to the MSE. This new method is then extended to the wavelet representation of images. Experimental results are given to illustrate the applications of these new ideas.

Acknowledgments

First off, I wish to thank my supervisor, Edward Vrscay, for his support, patience and good-naturedness during the course of the research for and preparation of this thesis. His wonderful attitude towards both work and fun made my stay at the University of Waterloo an excellent one. I could not have asked for a more friendly and approachable supervisor.

Second, I want to thank my thesis review committee/readers, Jeff Orchard and Justin Wan, for their helpful comments in preparing the final version of my thesis.

My time at the University of Waterloo was supported in several ways including Ontario Graduate Scholarships, Faculty of Mathematics scholarships, and teaching assistanceships. For all of these I am very appreciative. Aside from the obvious financial flexibility these afforded me, I know the teaching experience I gained will be invaluable as I move forward.

Contents

List of Figures

Chapter 1

Mathematical Preliminaries

1.1 Introduction

This chapter contains the background material necessary to understand the methods of this thesis. This discussion is not meant to be extensive and results are given without proof. Image functions are described, some measurements of image fidelity are given, several spaces are defined, and a short primer on one and two-dimensional wavelet methods is provided.

1.2 Images as Functions

A digital image is essentially a matrix $\mathcal{I}$ whose entries are known as greyscale or colour values. Given an $m \times n$ grid $I \subset \mathbb{Z}^2$, an $m \times n$ digital image may be viewed as a function u on I. Let $i \in I$ and consider the pair $(i, u(i))$ which is called a pixel. The second component $u(i)$ is the image value at i. If $u(i) \in \mathbb{R}$ then u is called a greyscale image. In this case, $\mathcal{I}$ is simply a two-dimensional matrix. In the case of colour images, $u(i) \in \mathbb{R}^3$ (one component for each of red, green and blue) and $\mathcal{I}$ is a three-dimensional matrix.

The $u(i)$ are also restricted further to what is called a *greyscale range*. For example, the greyscale range could be $0 - 255$, in which case u would represent an 8-bit image (there are 2^8 possible values for each $u(i)$). In the case of normalized images the greyscale range is $[0, 1]$.

In the rest of this thesis only greyscale images are considered but the concepts introduced can certainly be extended to include other types of images.

1.3 Comparison of Images

To compare results and measure error, it is necessary to be able to quantify error in some way. There are several ways by which this will be accomplished in this thesis, and for reference they are all included here. The first of these is the L^2 distance.

Definition 1.1. Let $x = \{x_1, x_2, \cdots, x_N\}$ and $y = \{y_1, y_2, \cdots, y_N\}$ be two real sets of data. The L^2 distance between x and y is

$$\left(\sum_{k=1}^{N}(x_k - y_k)^2\right)^{\frac{1}{2}}.$$

A variation on the L^2 distance is the root mean squared error (RMSE) which is obtained by a rescaling of the L^2 distance.

Definition 1.2. The root mean squared error between x and y is

$$\mathrm{RMSE}(x, y) = \left(\frac{1}{N}\sum_{k=1}^{N}(x_k - y_k)^2\right)^{\frac{1}{2}}.$$

Occasionally the RMSE will be reported as the mean squared error (MSE) where $\sqrt{\mathrm{MSE}} = \mathrm{RMSE}$.

The last form of measurement that will be used differs greatly from the L^2 distance and is called the Structural Similarity (SSIM) Index. As an entire chapter is devoted to the development and usefulness of the SSIM Index as a measurement of image fidelity, only a definition will be given here.

Definition 1.3. The SSIM value of x relative to y is

$$S(x, y) = \frac{2\bar{x}\bar{y} + C_1}{\bar{x}^2 + \bar{y}^2 + C_1}\,\frac{2s_x s_y + C_2}{s_x^2 + s_y^2 + C_2}\,\frac{s_{xy} + C_3}{s_x s_y + C_3},$$

where

$$\begin{aligned}
\bar{x} &= \frac{1}{N}\sum_{k=1}^{N} x_k, \\
s_{xy} &= \frac{1}{N-1}\sum_{k=1}^{N}(x_k - \bar{x})(y_k - \bar{y}), \quad \text{and} \\
s_x^2 &= \frac{1}{N-1}\sum_{k=1}^{N}(x_k - \bar{x})^2.
\end{aligned}$$

The parameters C_1, C_2 and C_3 are small positive constants.

1.4 Metric spaces

A metric space is a set X with a metric d which gives a notion of distance between two elements of X.

Definition 1.4. A metric d on a set X is a real-valued function $d(x, y)$ defined for all $x, y \in X$ such that the following conditions are satisfied:

1. Positivity: $d(x, y) \geq 0$ and $d(x, x) = 0$ for all $x, y \in X$.

2. Strict positivity: $d(x, y) = 0$ implies $x = y$.

3. Symmetry: $d(x, y) = d(y, x)$.

4. Triangle inequality: $d(x, y) \leq d(x, z) + d(z, y)$ for all $x, y, z \in X$.

The pair (X, d) is called a metric space.

Convergence of sequences in metric spaces is very important in the formulation of the proper setting for wavelet theory. This concept is now defined rigorously.

Definition 1.5. An infinite sequence $\{x_n\}$ in a metric space (X, d) is said to be a Cauchy sequence if given $\epsilon > 0$, there is an $N > 0$ so that $d(x_n, x_m) < \epsilon$ for all $n, m > N$.

Definition 1.6. An infinite sequence $\{x_n\}$ in a metric space (X, d) converges to $x \in X$ if for any $\epsilon > 0$, there is an $N > 0$ such that $d(x_n, x) < \epsilon$ for all $n > N$.

Definition 1.7. A metric space (X, d) is said to be complete if every Cauchy sequence converges to an element in X.

1.5 Normed linear spaces

A norm $\|\cdot\|$ on a vector space X gives a notion of size for each element of X.

Definition 1.8. Let X be a real or complex vector space. A real-valued function $\|\cdot\|$ defined on X is a norm on X if

1. Positivity: $\|x\| \geq 0$ for all $x \in X$.

2. Strict positivity: $\|x\| = 0$ if and only if $x = 0$.

3. Triangle inequality: $\|x + y\| \leq \|x\| + \|y\|$ for all $x, y \in X$.

4. Homogeneity: $\|\alpha x\| = |\alpha|\|x\|$ for any scalar α and all $x \in X$.

The quantity $\|x\|$ is the length of x and the pair $(X, \|\cdot\|)$ is called a normed linear space.

A norm $\|\cdot\|$ on a vector space X induces a metric by $d(x, y) = \|x - y\|$ for all $x, y \in X$. This is easily verified by checking the four conditions in the definition of a metric.

Definition 1.9. Let $(X, \|\cdot\|)$ be a normed linear space and let d be the metric induced by the norm $\|\cdot\|$. If the metric space (X, d) is complete, then $(X, \|\cdot\|)$ is called a complete normed linear space, also known as a Banach space.

A pertinent example of a complete normed linear space in the context of this thesis is $(\mathbb{R}^N, \|\cdot\|_2)$ where

$$\|x\|_2 = \left(\sum_{k=1}^{N} |x_k|^2\right)^{\frac{1}{2}}$$

is the Euclidean or L^2 norm. The L^2 distance is generated by the L^2 norm.

1.6 Inner product spaces

Inner product spaces are important in the formulation of wavelet theory. They are also intimately related to normed linear spaces and metric spaces.

Definition 1.10. Let X be a complex vector space. An inner product $\langle\ ,\ \rangle$ on X satisfies $\langle x, y\rangle \in \mathbb{C}$ for all $x, y \in X$ and

$$\begin{aligned}
\langle x+y, z\rangle &= \langle x, z\rangle + \langle y, z\rangle \text{ for all } x, y, z \in X. \\
\langle \alpha x, y\rangle &= \alpha\langle x, y\rangle \text{ for all } x, y \in X, \alpha \in \mathbb{C}. \\
\langle x, y\rangle &= \overline{\langle y, x\rangle} \text{ for all } x, y \in X. \text{ The bar denotes complex conjugation.} \\
\langle x, x\rangle &\geq 0 \text{ and } \langle x, x\rangle = 0 \text{ if and only if } x = 0.
\end{aligned}$$

The pair $(X, \langle\ ,\ \rangle)$ is called an inner product space.

Given an inner product space $(X, \langle\ ,\ \rangle)$, the inner product induces a norm by $\|x\| = \sqrt{\langle x, x\rangle}$. As mentioned, the norm induces a metric. Therefore, an inner product $\langle\ ,\ \rangle$ on a vector space X induces a metric by $d(x, y) = \sqrt{\langle x-y, x-y\rangle}$. The next definition then follows naturally.

Definition 1.11. Let $(X, \langle\ ,\ \rangle)$ be an inner product space and let d be the metric induced by $\langle\ ,\ \rangle$. If the metric space (X, d) is complete then $(X, \langle\ ,\ \rangle)$ is called a complete inner product space, also known as a Hilbert space.

An important part of the construction of a Hilbert space $(X, \langle\ ,\ \rangle)$ is the ability to construct a basis for X. This forms the groundwork for Generalized Fourier Series of which the traditional sine and cosine basis seen in Fourier analysis is a special case. To this end, a definition and a theorem are now given.

Definition 1.12. An orthonormal sequence $\{e_n\}$ in a Hilbert space H is maximal (complete) if $\langle x, e_n\rangle = 0$ for all n implies that $x = 0$.

Theorem 1.1 (Generalized Fourier Series)**.** *Let $\{e_n\}$ be an orthonormal sequence in a Hilbert space H which has a countable dense subset. If $\{e_n\}$ is complete then for any $x \in H$, $x = \sum_{n=1}^{\infty}\langle x, e_n\rangle e_n$.*

Proof. See Section 3.5 of [13]. □

1.7 Special spaces

Two spaces are now introduced that will be particularly useful in the development of wavelet-based methods. These are the $\mathcal{L}^p$ and ℓ^p spaces.

Definition 1.13. Let $\mathbf{X} \subseteq \mathbb{R}^{\mathbb{N}}$ and $p \in [1, \infty)$. Define the space $\mathcal{L}^p(\mathbf{X})$ by

$$\mathcal{L}^p(\mathbf{X}) = \left\{ f : \mathbf{X} \to \mathbb{R} \mid \int_{\mathbf{X}} |f(\mathbf{x})|^p d\mathbf{x} < \infty \right\}.$$

If $p = 2$ then the pair $(\mathcal{L}^2(\mathbf{X}), \langle\ ,\ \rangle)$, where $\langle f, g \rangle = \int_{\mathbf{X}} f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x}$, is a Hilbert space.

Definition 1.14. Let $\mathbf{X} \subseteq \mathbb{R}^N$ and let $\{\mathbf{x}_n\}$ be a sequence on $\mathbf{X}$. For $1 \leq p \leq \infty$, the space

$$\ell^p(\mathbf{X}) = \left\{ \{\mathbf{x}_n\} \mid \sum_{n=1}^{\infty} |\mathbf{x}_n|^p < \infty \right\}$$

is the set of all p-summable sequences on $\mathbf{X}$.

Once again, if $p = 2$ then the pair $(\ell^2(\mathbf{X}), \langle\ ,\ \rangle)$, where $\langle x, y \rangle = \sum_{n=1}^{\infty} x_n y_n$, is a Hilbert space.

1.8 Wavelets

1.8.1 Wavelets in one dimension

A common setting for image processing lies in the wavelet representation of an image. To this end, a brief background on wavelets will now be provided in order to establish the notation and theory that will be used later on.

Much of the background for the current theory of wavelets was established by Haar in 1909 when he showed that one can generate a complete orthonormal basis for $\mathcal{L}^2(\mathbb{R})$ from the following function known as the Haar wavelet:

$$\psi(t) = \begin{cases} 1, & 0 \leq t < 0.5, \\ -1, & 0.5 \leq t < 1, \\ 0, & \text{otherwise.} \end{cases}$$

In particular, by considering translations and dilations of $\psi(t)$ of the form

$$\psi_{jk}(t) = 2^{j/2} \psi(2^j t - k), \quad j, k \in \mathbb{Z},$$

Haar showed that $\{\psi_{jk}(t) \mid j, k \in \mathbb{Z}\}$ is a complete orthonormal basis for $\mathcal{L}^2(\mathbb{R})$. The work done by Haar is in fact a special case of what is now known as a multiresolution analysis (MRA) of $\mathcal{L}^2(\mathbb{R})$. We give a definition [8]:

Definition 1.15. Let $\{V_j\}_{j\in\mathbb{Z}}$ be a sequence of closed subspaces of $\mathcal{L}^2(\mathbb{R})$. Then $\{V_j\}$ is called a multiresolution analysis with scaling function $\phi \in \mathcal{L}^2(\mathbb{R})$ and mother wavelet $\psi \in \mathcal{L}^2(\mathbb{R})$ if:

1. The subspaces are nested: $V_j \subset V_{j+1}, j \in \mathbb{Z}$.

2. Density: $\overline{\cup_j V_j} = \mathcal{L}^2(\mathbb{R})$ (the bar denotes closure of the set).

3. Separation: $\cap_j V_j = \{0\}$.

4. There is a sequence of orthogonal complements $W_j \perp V_j$ such that $V_{j+1} = W_j \oplus V_j$.

5. The set of functions $\{\phi_{jk}(t) = 2^{j/2}\phi(2^j t - k) \mid k \in \mathbb{Z}\}$ forms an orthonormal basis for V_j.

6. The set of zero-mean functions $\{\psi_{jk}(t) = 2^{j/2}\psi(2^j t - k) \mid k \in \mathbb{Z}\}$ forms an orthonormal basis for W_j.

When writing $V_{n+1} = W_n \oplus V_n$ it is said that V_{n+1} is the direct sum of the vector spaces W_n and V_n. That is, if $z \in V_{n+1}$ then z may be written uniquely as $z = x + y$ where x and y are the projections of z in W_n and V_n, respectively.

Of particular importance are functions $f(x) \in \mathcal{L}^2(\mathbb{R})$ that admit expansions of the form

$$f(x) = b_{00}\phi_{00}(x) + c_{00}\psi_{00}(x) + \sum_{j=1}^{\infty}\sum_{k=0}^{2^j-1} c_{jk}\psi_{jk}(x),$$

where $b_{00} = \langle f, \phi_{00}\rangle$ and $c_{jk} = \langle f, \psi_{jk}\rangle$. If the scaling function ϕ and the mother wavelet ψ have compact support on $\mathbb{R}$, it follows that $f(x)$ has compact support as well. The coefficient b_{00} is called an approximation coefficient and the c_{ij} are known as wavelet or detail coefficients. These coefficients can be arranged in the form of an infinite binary tree as in Table 1.8.1. Note that in the last row, B_{jk} denotes the binary tree of infinite length whose root is c_{jk}.

<table>
<tr><td colspan="8">b_{00}</td></tr>
<tr><td colspan="8">c_{00}</td></tr>
<tr><td colspan="4">c_{10}</td><td colspan="4">c_{11}</td></tr>
<tr><td colspan="2">c_{20}</td><td colspan="2">c_{21}</td><td colspan="2">c_{22}</td><td colspan="2">c_{23}</td></tr>
<tr><td>B_{30}</td><td>B_{31}</td><td>B_{32}</td><td>B_{33}</td><td>B_{34}</td><td>B_{35}</td><td>B_{36}</td><td>B_{37}</td></tr>
</table>

Table 1.8.1: Arrangement of wavelet coefficients in an infinite binary tree

.

1.8.2 Wavelets in two dimensions

To work with images, a two-dimensional extension of the above theory is vital. The first task is to construct an orthonormal basis for $\mathcal{L}^2(\mathbb{R}^2)$. The obvious choice is to take $\{\psi_{jk}\psi_{lm} \mid j,k,l,m \in \mathbb{Z}\}$. The problem with this choice is that the multiresolution structure from one dimension is not preserved since basis functions at different resolutions are mixed. There is in fact a very elegant way to construct an MRA for $\mathcal{L}^2(\mathbb{R}^2)$ using an MRA for $\mathcal{L}^2(\mathbb{R})$ that preserves the useful features seen in one dimension.

Let $\{V_n\}_{n\in\mathbb{Z}}$ with scaling function ϕ and mother wavelet ψ be an MRA for $\mathcal{L}^2(\mathbb{R})$. To construct a wavelet basis for $\mathcal{L}^2(\mathbb{R})$, consider the sequence of closed subspaces $\{\mathbf{V}_j\}_{j\in\mathbb{Z}}$ of $\mathcal{L}^2(\mathbb{R}^2)$ defined by

$$\mathbf{V}_n = V_n \otimes V_n = \overline{\operatorname{span}\{F(x,y) = f(x)g(y)|f,g \in V_n\}},$$

called tensor product spaces [8]. The bar denotes the closure of the set. Several properties can be deduced now. First, the subspaces are nested:

$$\mathbf{V}_j \subset \mathbf{V}_{j+1}, \quad j \in \mathbb{Z}.$$

Next,

$$\overline{\cup_j \mathbf{V}_j} = \mathcal{L}^2(\mathbb{R}^2).$$

Finally,

$$\cap_j \mathbf{V}_j = \{0\}.$$

Since the set of functions

$$\{\phi_{jk}(t) = 2^{j/2}\phi(2^j t - k) \ |k \ \in \mathbb{Z}\}$$

forms an orthonormal basis for V_j, the set of functions

$$\{\Phi_{jkl}(x,y) = 2^j\phi(2^j x - k)\phi(2^j y - l) \mid k,l \in \mathbb{Z}\}$$

forms an orthonormal basis for $\mathbf{V}_j$. For each $\mathbf{V}_n$, define $\mathbf{W}_n$ to be the orthogonal complement of $\mathbf{V}_n$ in $\mathbf{V}_{n+1}$. Then

$$\begin{aligned}
\mathbf{V}_{n+1} &= V_{n+1} \otimes V_{n+1} \\
&= (W_n \oplus V_n) \otimes (W_n \oplus V_n) \\
&= (V_n \otimes V_n) \oplus [(W_n \otimes V_n) \oplus (V_n \otimes W_n) \oplus (W_n \otimes W_n)] \\
&= \mathbf{V}_n \oplus \mathbf{W}_n \qquad (1.8.1)
\end{aligned}$$

Write $\mathbf{W}_n^v = (W_n \otimes V_n)$, $\mathbf{W}_n^h = (V_n \otimes W_n)$, and $\mathbf{W}_n^d = (W_n \otimes W_n)$. Then $\mathbf{W}_n$ consists of three pieces and each of these pieces ($\mathbf{W}_n^v$, $\mathbf{W}_n^h$, and $\mathbf{W}_n^d$) have orthonormal bases given by

$$\begin{aligned}
&\overline{\{\Psi_{nij}^v(x,y) = \psi_{ni}(x)\phi_{nj}(y) \mid i,j \in \mathbb{Z}\}}, \\
&\overline{\{\Psi_{nij}^h(x,y) = \phi_{ni}(x)\psi_{nj}(y) \mid i,j \in \mathbb{Z}\}}, \quad \text{and} \\
&\overline{\{\Psi_{nij}^d(x,y) = \psi_{ni}(x)\psi_{nj}(y) \mid i,j \in \mathbb{Z}\}},
\end{aligned}$$

respectively. Thus, $\{\mathbf{V}_n\}_{n\in\mathbb{Z}}$ with scaling function $\Phi = \Phi_{000} \in \mathcal{L}^2(\mathbb{R}^2)$ and mother wavelet $\Psi = \Psi_{000} \in \mathcal{L}^2(\mathbb{R}^2)$ forms an MRA for $\mathcal{L}^2(\mathbb{R}^2)$. Having completed construction of an MRA, consider again functions $f(x,y) \in \mathcal{L}^2(\mathbb{R}^2)$ admitting expansions of the form

$$f(x,y) = b_{000}\Phi_{000}(x,y)+\sum_{j=0}^{\infty}\sum_{k=0}^{2^j-1}\sum_{l=0}^{2^j-1}\left[c^h_{jkl}\Psi^h_{jkl}(x,y) + c^v_{jkl}\Psi^v_{jkl}(x,y) + c^d_{jkl}\Psi^d_{jkl}(x,y)\right]. \tag{1.8.2}$$

In one dimension, the coefficients were arranged in the form of an infinite binary tree. In two dimensions, the coefficients are arranged in quadtrees. These quadtrees may be represented by a pyramid of blocks as Fig. 1.8.1 illustrates. For some $J > 0$ each of the blocks, D^h_j, D^v_j, and $D^d_j, 0 \leq j \leq J$, contain 2^{2j} detail coefficients c^h_{jkl}, c^v_{jkl}, and c^d_{jkl} respectively. The collections of blocks

$$D^h = \bigcup_{j=0}^{\infty} D^h_j, \quad D^v = \bigcup_{j=0}^{\infty} D^v_j, \quad D^d = \bigcup_{j=0}^{\infty} D^d_j,$$

comprise the horizontal, vertical, and diagonal quadtrees of the coefficient tree.

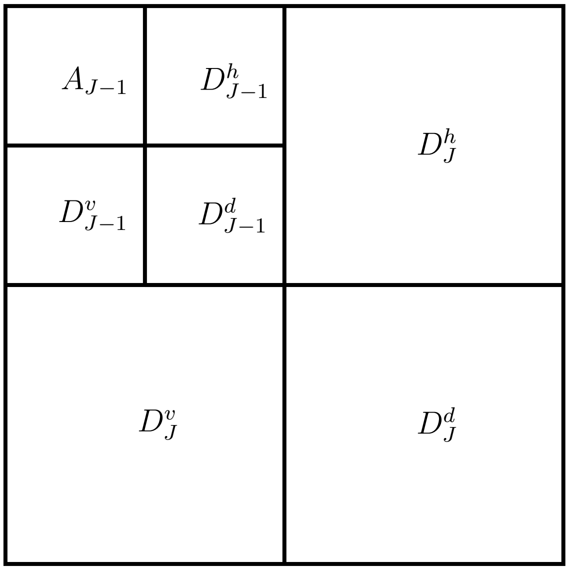


Figure 1.8.1: Pyramid of blocks of wavelet quadtree coefficients.

There exists a computationally practical way [15] to determine the expansion coefficients in (1.8.2) with what is known as an *analysis* algorithm. Briefly, begin with an "image" which is a matrix of coefficients denoted A_N. These can be obtained from a discrete sampling of a continuous signal (this is essentially what a digital image is). From A_N, compute the blocks $A_{N-1}, D^h_{N-1}, D^v_{N-1}$, and D^d_{N-1} in a process known as decomposition. These last three blocks are stored and A_{N-1} is decomposed into the blocks $A_{N-2}, D^h_{N-2}, D^v_{N-2}$, and D^d_{N-2}. This process is continued iteratively until the single entry blocks A_0, D^h_0, D^v_0, and D^d_0 (the top of the pyramid of quadtrees) are reached. The image can be constructed in reverse fashion with what is known as a *synthesis* algorithm.

Chapter 2

Self-similarity of images with respect to the L^2 norm

2.1 Introduction

In the context of imaging, the term "self-similar" brings to mind an image whose pixel values or structures in different parts of the image mimic each other in some way. However, before embarking on an investigation of image self-similarity, a natural question to ask is "Why study image self-similarity?" In a nutshell, it warrants study because it forms an integral part of many imaging techniques. It is particularly important in fractal image coding and non-local means denoising, two techniques which will be investigated later in this thesis. The success of these techniques hinges upon being able to find (possibly modified) subblocks of an image whose L^2 distance is small (or subblocks that are similar with respect to L^2). This is in fact just one of many possible definitions of image self-similarity.

At a very simplistic level, image self-similarity can be interpreted in terms of translations of image subblocks. That is, given a grid I, an image function $u = \{u(i) \mid i \in I\}$, and two $m \times n$ pixel blocks $\mathcal{R}_i$ and $\mathcal{R}_j$ in I, the two image subblocks $u(\mathcal{R}_i)$ and $u(\mathcal{R}_j)$ are similar ("close") if $\|u(\mathcal{R}_i) - u(\mathcal{R}_j)\|$ is small for some norm $\|\cdot\|$. When the L^2 norm is used, this definition forms the basis for the traditional non-local means denoising algorithm [7] which will be explored in Chapter 4.

This strictly translational definition is often too restrictive, particularly in the context of imaging. For example, when looking at a unevenly lit wall in a room, one could consider the differently lit parts as being similar up to a shift in brightness or a greyscale shift. Therefore, the above definition for image self-similarity could be modified to say that two image subblocks are similar if $\|u(\mathcal{R}_i) - u(\mathcal{R}_j) - \beta\|$ is small for some constant β depending on $u(\mathcal{R}_i)$ and $u(\mathcal{R}_j)$.

This relaxation can be taken one step further by allowing for affine greyscale transformations of image subblocks. In this case, two image subblocks are said to be similar if $\|u(\mathcal{R}_i) - \alpha u(\mathcal{R}_j) - \beta\|$ is small for some constants α and β depending on $u(\mathcal{R}_i)$ and $u(\mathcal{R}_j)$. When combined with a decimation in pixel space, this forms the basis for fractal image coding [3,4,14]. Informally, this is written as $u(\mathcal{R}_i) \approx \alpha u(\mathcal{D}_j) + \beta$ where $\mathcal{D}_j$ is larger than $\mathcal{R}_i$. This idea will be explored in detail later.

These three definitions will be important in forming models of image self-similarity in the pixel domain. Later, modified versions of these models will be applied in the wavelet domain.

2.2 Image self-similarity in the pixel domain

Let I be a grid and let u represent a greyscale image on I. In order to mathematically define the concept of self-similarity of an image, consider the following model that was introduced in [2]. There are several components which comprise the model:

1. Range blocks: A set $\mathcal{R}$ of non-overlapping $n \times n$ pixel blocks $\mathcal{R}_i, 1 \leq i \leq N_{\mathcal{R}}$, that forms a partition of I.

2. Domain blocks: A set $\mathcal{D}$ of $m \times m$ pixel blocks $\mathcal{D}_i, 1 \leq i \leq N_{\mathcal{D}}$ such that $\cup_i \mathcal{D}_i = I$. The $\mathcal{D}_i$ are not necessarily disjoint.

3. Affine geometric transformations $w_{ij}^{(k)} : \mathcal{D}_j \to \mathcal{R}_i, 1 \leq k \leq 8$. There are eight possible mappings (four rotations and four inversions about the centre) since both blocks are square. If $m > n$, the $w_{ij}^{(k)}$ are contractive mappings and incorporate a pixel decimation operation.

4. Affine maps $\phi : R_g \to R_g$ of the form $\phi(t) = \alpha t + \beta$, where R_g denotes an appropriate greyscale range, usually $[0,1]$ or $0 - 255$.

The goal is to examine how well a portion of u can be approximated by another portion of u. Based on the above this can be written as

$$u(\mathcal{R}_i) \approx \phi_{ij}(u(w_{ij}^{-1}(\mathcal{R}_i))) = \alpha_{ij} u(\mathcal{D}_j) + \beta_{ij}, \quad 1 \leq i \leq N_{\mathcal{R}}, \quad 1 \leq j \leq N_{\mathcal{D}}, \tag{2.2.1}$$

omitting the k superscripts for notational convenience. For each non-overlapping range block $u(\mathcal{R}_i)$, the distribution of errors Δ_{ij} between $u(\mathcal{R}_i)$ and $u(\mathcal{D}_j)$ associated with (2.2.1) is of primary interest. These errors take the form

$$\Delta_{ij} = \min_{\alpha,\beta} \| u(\mathcal{R}_i) - \alpha u(\mathcal{D}_j) - \beta \|_2, \tag{2.2.2}$$

where the (α, β) pairs are chosen from an appropriate parameter space to ensure that $\phi : R_g \to R_g$. Four cases involving optimization over α and β as well as decimations in pixel space will be considered. The four cases are:

1. Purely translational: The domain and range blocks $\mathcal{D}_j$ and $\mathcal{R}_i$ have the same size. The $w_{ij}^{(k)}$ are translations. Set $\alpha = 1$ and $\beta = 0$.

2. Translational and greyscale shift: The domain and range blocks have the same size. The $w_{ij}^{(k)}$ are translations. Set $\alpha = 1$ and optimize over β.

3. Affine, same-scale: The domain and range blocks have the same size. The $w_{ij}^{(k)}$ are translations. Optimize over α and β.

4. Affine, two-scale: The domain blocks are larger than the range blocks. The affine transformations $w_{ij}^{(k)}$ incorporate a decimation in pixel space. Optimize over α and β.

The problem of solving for the optimal coefficients α and β is a simple least squares problem which takes the following form: Given two real-valued data sets $\{u_1, \cdots, u_N\}$ and $\{v_1, \cdots, v_N\}$, minimize the sum of the squared errors,

$$R(\alpha, \beta) = \sum_{k=1}^{N} (u_k - \alpha v_k - \beta)^2. \tag{2.2.3}$$

In the context of images, the data sets are the pixel intensities of two different image subblocks. It is easy to show that the optimal parameters for cases 2 and 3 are

1. Case 2: $\beta = \bar{u} - \bar{v}$.

2. Case 3: $\alpha = \frac{s_{uv}}{s_v^2}$ and $\beta = \bar{u} - \alpha\bar{v}$.

Here,

$$\begin{aligned} \bar{u} &= \frac{1}{N}\sum_{k=1}^{N} u_k, \\ s_{uv} &= \frac{1}{N-1}\sum_{k=1}^{N}(u_k - \bar{u})(v_k - \bar{v}), \quad \text{and} \\ s_u^2 &= \frac{1}{N-1}\sum_{k=1}^{N}(u_k - \bar{u})^2. \end{aligned}$$

In what follows, images with normalized greyscale ranges are used. That is, $u(i) \in [0, 1]$ for all $i \in I$. Also, non-overlapping 8×8 range blocks are used for computational efficiency. The distance reported between two images subblocks of the same size is the root mean squared error (RMSE).

2.2.1 Cases 1, 2, and 3

For these cases, the standard 8-bit 512 × 512 *Lena* and *Mandrill* images are used. Results are illustrated with histograms that describe the error distribution between distinct domain and range blocks pairs. A histogram with a peak close to zero indicates a good degree of self-similarity, and the more closely concentrated a histogram is to zero, the higher the degree of self-similarity. Since more parameters are optimized over as the case numbers increase, the errors will satisfy

$$0 \leq \Delta_{ij}^{\text{Case 3}} \leq \Delta_{ij}^{\text{Case 2}} \leq \Delta_{ij}^{\text{Case 1}}. \tag{2.2.4}$$

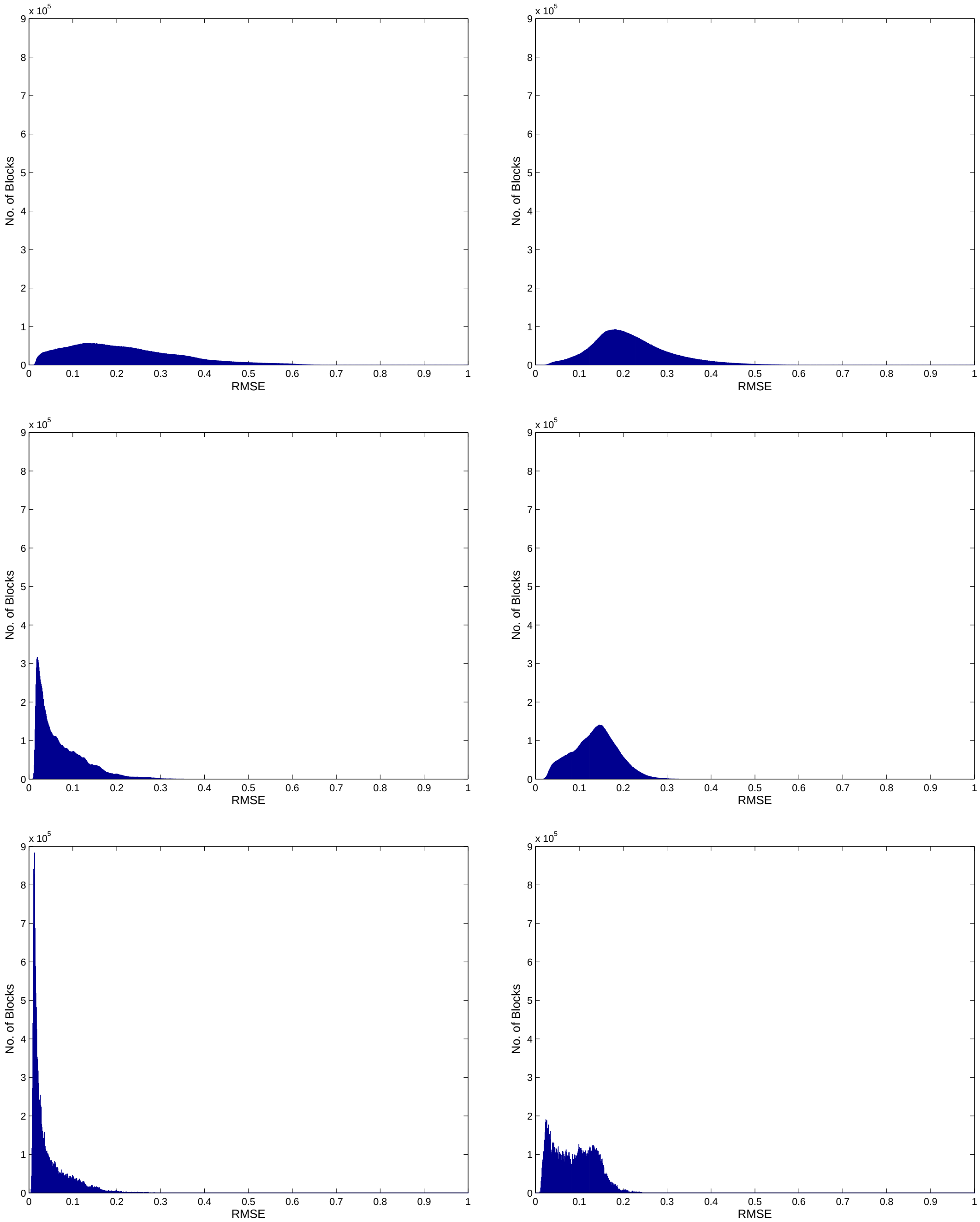


Figure 2.2.1: Case 1-3 pixel-based error distributions for normalized *Lena* (left) and *Mandrill* (right). From top to bottom are case 1, case 2 and case 3.

Note that case 4 involves a decimation in pixel space, and therefore its results are not directly comparable to cases 1, 2 and 3. Its usefulness is more in its application to fractal image coding and will be explored in the next section.

The top row of Fig. 2.2.1 corresponds to case 1 with *Lena's* results on the left and *Mandrill's* results on the right. Note that both images exhibit peaking in their error distributions at approximately 0.15. The next two rows show that the error distributions for *Lena* become extremely concentrated about zero when affine transformations are allowed. Even in the case where only β is allowed to vary great improvements are observed. On the other hand, while the *Mandrill* error distributions move closer to zero in cases 2 and 3, the improvement is not nearly as pronounced as with *Lena*. The conclusion from these error distributions is that both images are fairly self-similar translationally.

With a little ingenuity, it turns out that *Lena's* remarkable improvement can be explained in terms of block variances. To see this, modify case 2 slightly by setting $\alpha = 0$ and optimize over β. By optimizing $R(0, \beta)$ over β, it is not difficult to show that the optimal value is $\beta = \overline{u(\mathcal{R}_i)}$ (the bar denotes the mean). Thus this modified case 2 is simply approximating an image subblock with its mean. A plot of the RMSE histogram is then a plot of the block standard deviations. Fig. 2.2.2 gives the standard deviation histogram distributions for *Lena* and *Mandrill*. These show that the majority of *Lena's* subblocks have standard deviations close to zero while *Mandrill's* standard deviations are much more diffuse. In other words, *Lena's* blocks are generally "flatter" than those of *Mandrill*.

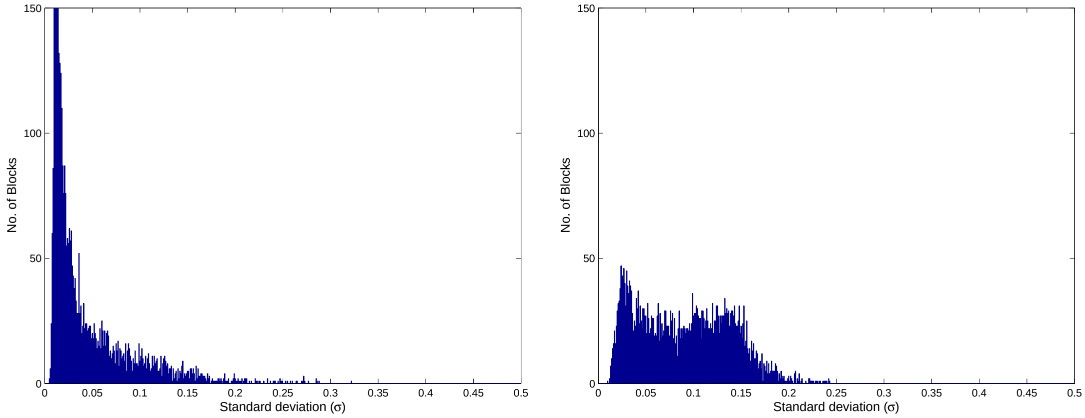


Figure 2.2.2: Histogram distributions of image subblock standard deviations for *Lena* (left) and *Mandrill* (right).

The best approximation of an image subblock by a constant (obtained by setting $\alpha = 0$ and optimizing over β in (2.2.2)) will generally produce poorer results than when α is non-zero. Symbolically,

$$0 \leq \Delta_{ij}^{\text{Case 2}} \leq \sigma(u(\mathcal{R}_i)).$$

Therefore, since *Lena's* blocks are generally quite flat, this should have the effect of pushing its case 2 error distribution closer to zero which implies that its case 3 error distribution should be closer to zero as well. On the other hand, the same conclusion cannot be drawn for *Mandrill* since its standard deviation histogram is so dispersed.

A simple test of this theory about block variances involves examining how well an image can be approximated by another distinct image. In particular, we

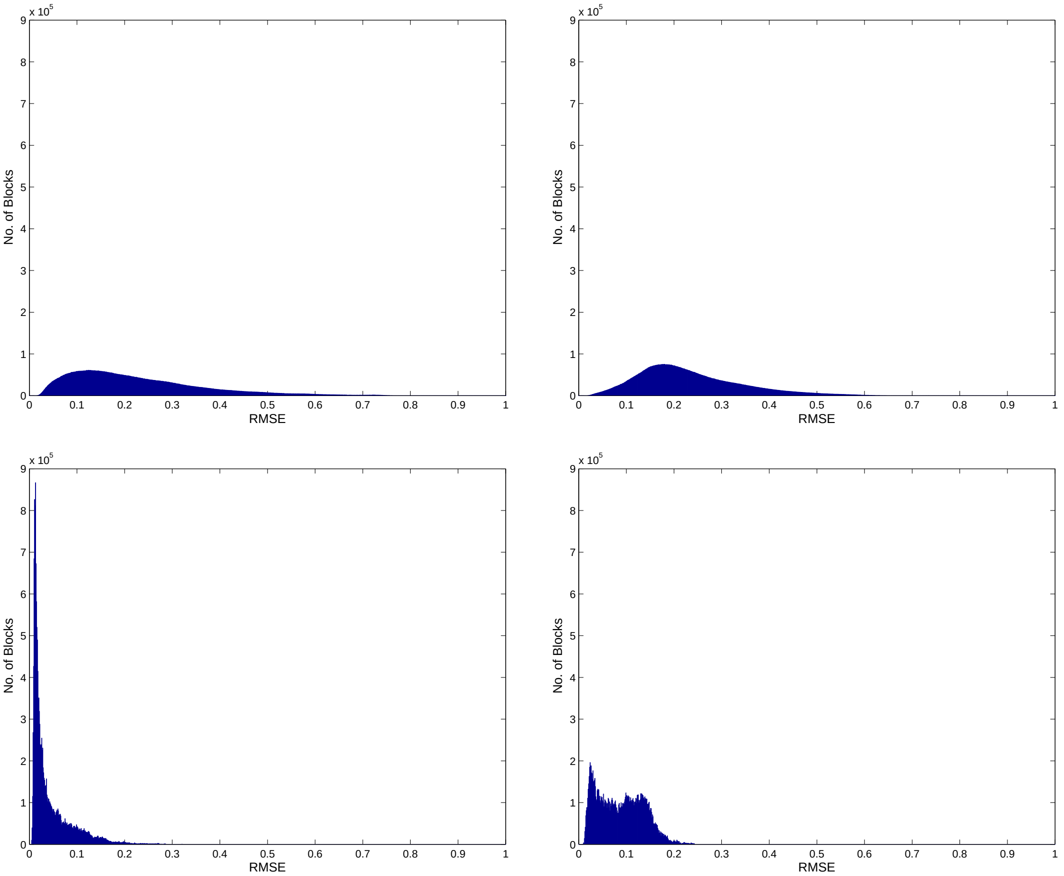


Figure 2.2.3: Pixel-based error distributions obtained by approximating *Lena* with *Goldhill* (left) and *Mandrill* with *Barbara* (right). The top row is case 1 and the bottom row is case 3.

tested how well *Lena* was approximated by *Goldhill* and how well *Mandrill* was approximated by *Barbara*. The pictures shown in Fig. 2.2.3 show the results for cases 1 and 3 (*Lena* on the left, *Mandrill* on the right). Note the striking similarity to the case 1 and 3 histograms seen before in Fig. 2.2.1. This shows that the source of image domain blocks is not as important as how well an image range block can be approximated. Since a block's variance is the main contributing factor to how well the block may be approximated, this shows that the degree of self-similarity of an image is a direct consequence of the variance of its subblocks.

2.2.2 Case 4

This section considers the case where $m > n$, i.e., where the range blocks are smaller than the domain blocks. In particular, given an image function u representing a 512×512 image, we examine how well all 4096 non-overlapping 8×8 image subblocks $u(\mathcal{R}_i)$ are approximated by all 1024 non-overlapping 16×16 image subblocks $u(\mathcal{D}_j)$. For the pixel decimation operation, 2×2 subblocks of $u(\mathcal{D}_j)$ are averaged. Fig. 2.2.4

shows the error distribution histograms for *Lena* and *Mandrill.* These indicate that an image subblock is generally well-approximated by a number of larger subblocks, especially in the *Lena* case. Also note that the case 4 histograms strongly mimic the case 3 histograms.

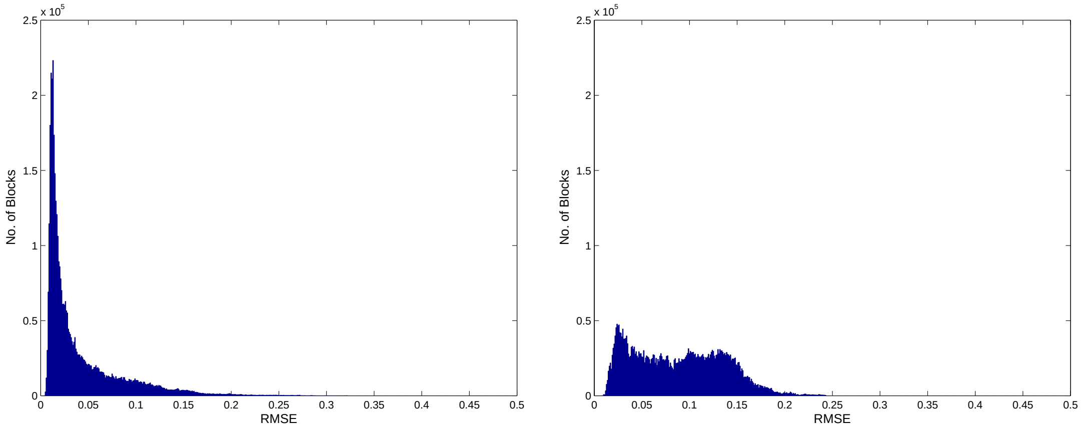


Figure 2.2.4: Case 4 pixel-based error distributions for normalized *Lena* (left) and *Mandrill* (right).

The previous discussion of self-similarity is not limited to *Lena* and *Mandrill.* Indeed, as the plots in Fig. 2.2.5 show, many other images also exhibit a good degree of self-similarity under optimization over α and β. Of course, this is still a small sample of images. Extensive experimentation [2] has shown that self-similarity is a common feature of natural images, the degree of which varies from better to worse than what is shown in the results presented here.

2.2.3 Application of case 4 to fractal coding

Let u be an image function. The idea behind fractal image coding is to approximate an image subblock $u(\mathcal{R}_i)$ with a geometrically-contracted, affine greyscale-modified copy of a larger image subblock $u(\mathcal{D}_j)$ [3]. Finding the best domain block for each range block by defining the pairs $(i, j(i))$ and the optimal greyscale parameter pairs (α_i, β_i) comprises the fractal code of u. This then defines a fractal operator T by modifying Eqn. (2.2.1). For such a pairing $(i, j(i))$, rewrite Eqn. (2.2.1) as

$$u(x) \approx (Tu)(x) = \alpha_i u(w^{-1}_{i,j(i)}(x)) + \beta_i, \quad x \in \mathcal{R}_i, \quad 1 \leq i \leq N_{\mathcal{R}_i}. \qquad (2.2.5)$$

The following theorem will be useful.

Theorem 2.1 (Banach's fixed point theorem)**.** *Let* (X, d) *be a complete metric space and let* $T : X \to X$ *be an operator and suppose there exists* $0 \leq c < 1$ *such that* $d(T(x), T(y)) \leq cd(x, y)$ *for all* $x, y \in X$*. Then* T *has a unique fixed point in* X*. That is, there exists a unique* $x^* \in X$ *such that* $T(x^*) = x^*$*.*

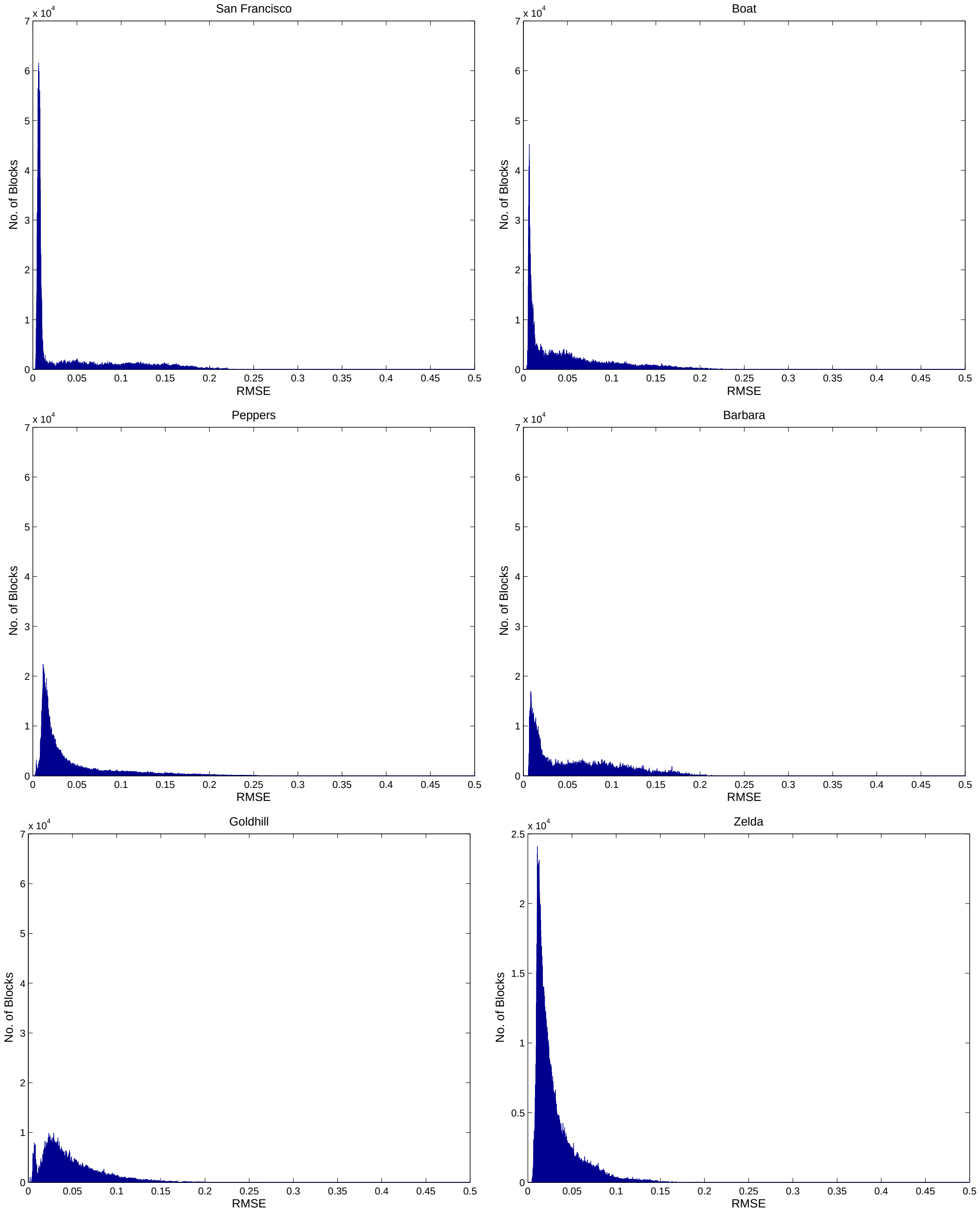


Figure 2.2.5: Case 4 pixel-based error distributions for six other normalized test images.

Such a T is called contractive (with contractive factor c). Under certain conditions on the maps $w_{i,j(i)}$, the operator T defined in (2.2.5) is a contractive operator in $\mathcal{L}^2(X)$ [11] and by Banach's theorem has a unique fixed point u^*. This fixed point u^* is an approximation to u. In application, it is desirable to make u^* as close to u as possible. How good this approximation can be depends on a variety of things including, most significantly, how self-similar an image is with regards to the mapping in Eqn. (2.2.5). One result that gives an upper bound on the distance between u and u^* is a corollary to Banach's theorem.

Corollary 2.1. *Let (X, d) be a complete metric space and let $T : X \to X$ be a contractive mapping on X with fixed point $x^* \in X$ and contractive factor c. Then for all $x \in X$*

$$\|x - x^*\| \leq \frac{1}{1-c}\|x - Tx\|.$$

To exploit this property, for each range block $u(\mathcal{R}_i)$ the domain pool is searched for the domain block $u(\mathcal{D}_{j(i)})$ that minimizes$\|u(\mathcal{R}_i) - Tu(\mathcal{D}_{j(i)})\|_2$. The image block $u(\mathcal{R}_i)$ is then replaced with a modified copy of $u(\mathcal{D}_{j(i)})$.

To give an example, the fractal operator T associated with *Lena* was computed. Fig. 2.2.6 shows the resulting images after three and six iterations of T starting from a blank black image. Also included is a histogram showing the steady-state distribution of errors for the domain and range block pairs which define the fractal transform for *Lena*. Since it is concentrated so closely around zero, this histogram shows that most of the range blocks of *Lena* are well approximated by domain blocks at a larger scale.

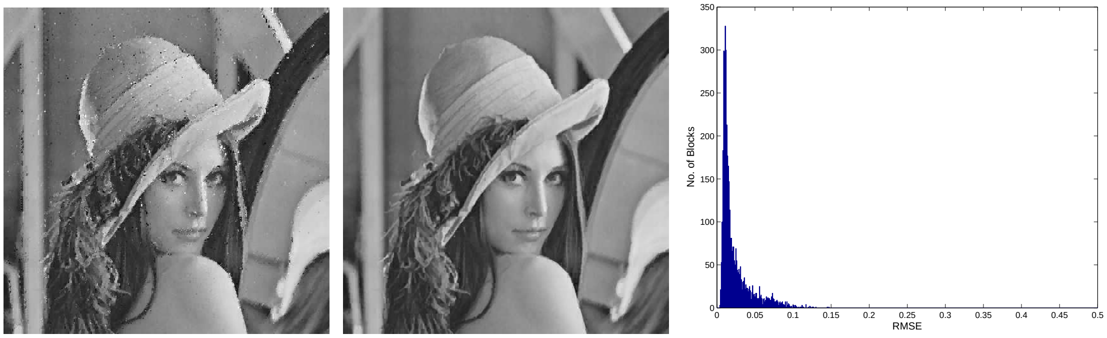


Figure 2.2.6: Left, centre: Fixed point approximation to *Lena* after 3 and 6 iterations of the fractal transform. Right: Distribution of errors for the best domain and range block pairings.

Also of interest are histogram distributions for α and β. On pp. 77-78 of [1], the α and β distributions for *Lena* and *Mandrill* were examined. For both of the images, the α distributions had a large symmetric peak about zero and the β distributions had their largest peaks around 0.5. As we showed earlier, *Lena* and *Mandrill* are quite different images in terms of their block variances and L^2-based error distributions. It is quite interesting then that their α and β statistics are so similar.

2.3 Image self-similarity in the wavelet domain

Consider an image of size $2^K \times 2^K$ $(K \geq 0)$ and the standard tensor-product (real) wavelet basis expansion of this image as discussed in Sec. 1.8.2. The wavelet coefficients can be arranged in a pyramid of blocks at each decomposition level. At level k of this decomposition, $0 \leq k \leq K$, the coefficients can further be partitioned into horizontal, vertical, and diagonal detail coefficients which will be denoted by A_k^h, A_k^v, and A_k^d respectively. Given a wavelet coefficient $a_{kij}^\lambda, \lambda \in \{h, v, d\}$, let A_{kij}^λ denote the unique quadtree rooted at a_{kij}^λ.

The concept of self-similarity is extended to the wavelet domain by examining how well quadtrees in the wavelet expansion of an image are approximated by other quadtrees, both at the same and different scales. Some preliminary work on this subject has already been done in [17]. These approximations will be of the form $A_{kij}^\lambda \approx \alpha A_{k'i'j'}^{\lambda'}, 0 \leq k' \leq k$. This is similar to what was done in the pixel domain except for the omission of the constant β. The reasoning behind this is that these wavelet quadtree expansions are theoretically of infinite length and we wish to preserve the ℓ^2-summable nature of the expansion coefficients. There are three cases:

1. Purely translational: Wavelet quadtrees are compared at the same scale. Set $k = k'$ and $\alpha = 1$.

3. Affine, one-scale: Quadtrees at the same scale are compared. Optimize over α.

4. Affine, two-scale: Let $k' < k$. Quadtrees at higher scales are approximated using quadtrees at lower scales. This is equivalent to a pixel decimation operation. Optimize over α.

The cases are numbered so that they may be considered as counterparts to the cases explored in the pixel domain. To find the optimal α, simply minimize $R(\alpha, \beta)$ in Eqn. (2.2.3) for the case $\beta = 0$. This yields

$$\alpha = \sum_{k=1}^{N} u_k v_k \Bigg/ \sum_{k=1}^{N} v_k^2 \,, \tag{2.3.1}$$

Here, u and v are N-vectors containing wavelet detail coefficients from A_{kij}^λ and $A_{k'i'j'}^{\lambda'}$ respectively.

For the numerical experiments, we set $k = 6$ (there are three decompositions total) and for case 4, $k' = 3$. For computational purposes, only quadtrees with the same orientation are compared (so $\lambda = \lambda'$). The Haar wavelet is used and the RMSE between quadtrees is reported.

2.3.1 Cases 1 and 3

Fig. 2.3.1 contains plots of the error distributions for cases 1 and 3 for *Lena* (left) and *Mandrill* (right). Note that *Lena's* error distributions both have their peaks fairly close to zero. This indicates that in the wavelet domain, *Lena* exhibits a good degree of similarity. By allowing for optimization over α, the peak of the error distribution is pushed quite dramatically towards zero and the histogram is much less diffuse.

In contrast to *Lena's* results, the error distributions for *Mandrill* are much more diffuse, similar to what was seen in the pixel domain. The peak gets pushed closer to zero when α is allowed to vary but the results are not nearly as good as what is achieved with *Lena*.

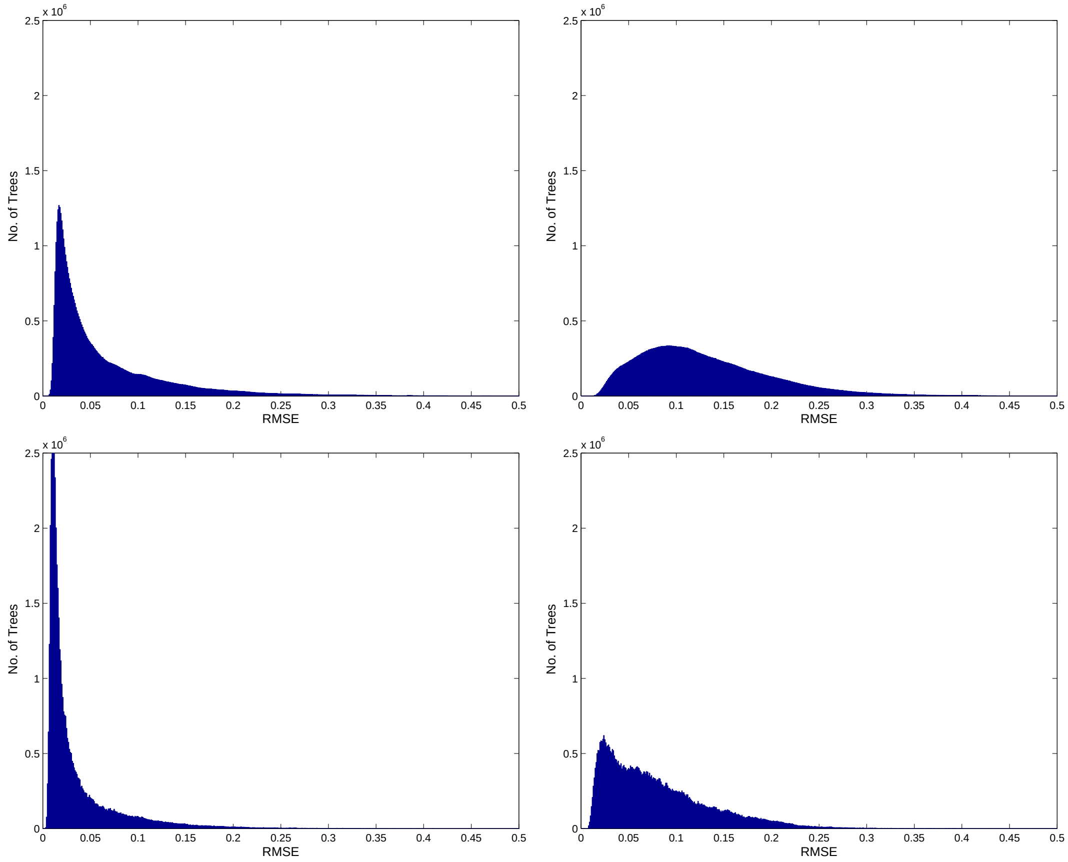


Figure 2.3.1: Cases 1 and 3 wavelet-based error distributions for *Lena* (left) and *Mandrill* (right). Top row: Case 1. Bottom row: Case 3.

To give an idea of what values the parameter α takes on, we refer the reader to the histogram distributions of α shown in Fig 2.3.2 for *Lena* and *Mandrill*. Note how similar the two histograms are and that they both have a large symmetrical peak about zero.

In the pixel domain, the degree of self-similarity of an image was found to be a direct consequence of how approximable its subblocks were. Also observed was that the lower the variance of a subblock, the easier it was to approximate. A natural question at this point is whether a similar result holds for the wavelet representation of an image. To this end, let u and v be two N-vectors and let α be

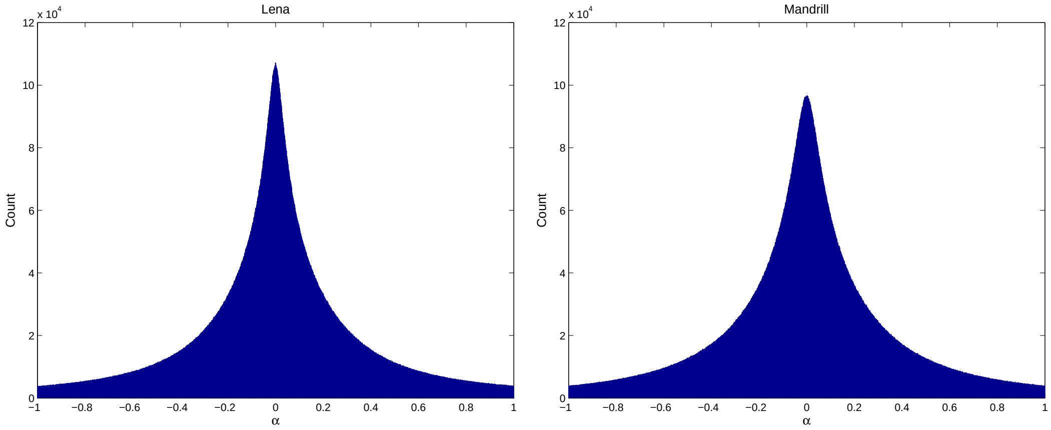


Figure 2.3.2: Histogram distributions of α in case 3 for *Lena* (left) and *Mandrill* (right).

given by (2.3.1). A simple calculation shows that

$$\|u - \alpha v\|_2 \leq \sqrt{\sum_{k=1}^{N} u_k^2}, \tag{2.3.2}$$

which is the square root of the energy of u. If u and v are two wavelet coefficient quadtrees, what this says is that the lower the energy of u, the lower the error in approximating u will be. That is, u is more approximable.

This analysis suggests examining the histogram distributions of the square roots of the energies of the quadtrees to explain Lena's great improvement in case 3. Fig. 2.3.3 contains these histograms for the quadtrees in the wavelet representations of *Lena* and *Mandrill*. These show that *Lena's* quadtrees generally have energy close to zero but *Mandrill's* energies are much more dispersed. By the above, this has the effect of pushing *Lena's* case 3 distributions closer to zero.

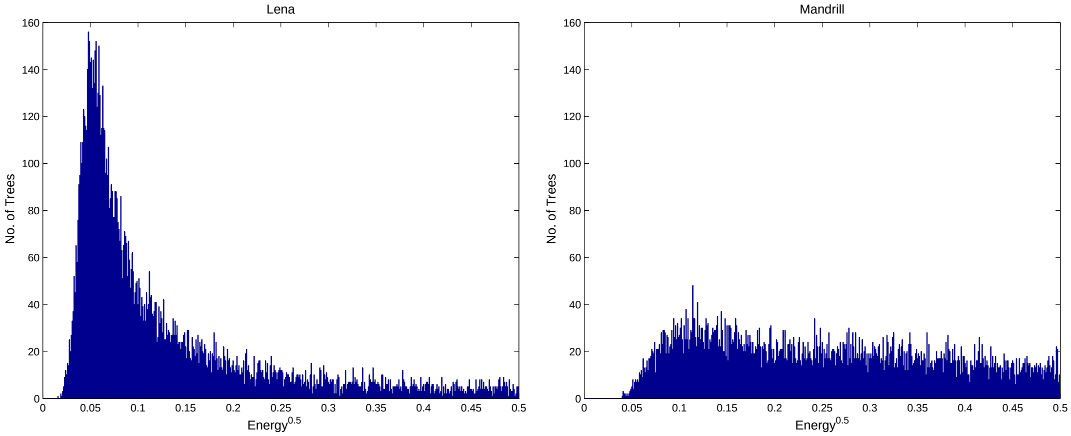


Figure 2.3.3: Histogram distributions of $\sqrt{\text{energy}}$ for wavelet quadtrees in *Lena* (left) and *Mandrill* (right).

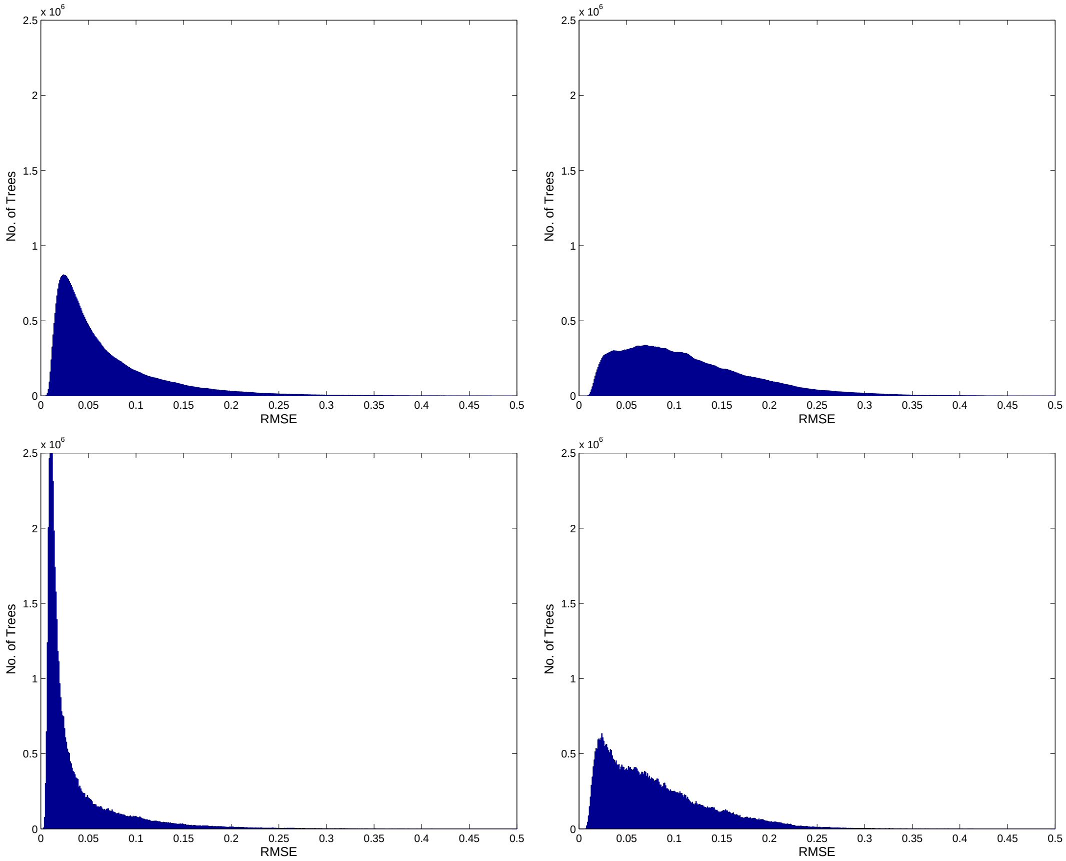


Figure 2.3.4: Cases 1 and 3 wavelet-based error distributions obtained by approximating *Lena* with *Goldhill* (left) and *Mandrill* with *Barbara* (right). The top row is case 1 and the bottom row is case 3.

Mimicking the work done in the pixel domain, we attempted to approximate quadtrees in *Lena* and *Mandrill* with quadtrees from *Goldhill* and *Mandrill* respectively. Fig. 2.3.4 shows the case 1 and 3 results of these experiments for *Lena* (left) and *Mandrill* (right). Like in the pixel domain, there is a remarkable resemblance to the case 1 and 3 results seen in Fig. 2.3.1.

This shows that the source of the quadtrees in an approximation is not as important as how well a quadtree can be approximated. Therefore, the analysis done in the pixel domain has a direct analogue in the wavelet domain: self-similarity of an image in the wavelet domain is a direct result of how well the quadtrees in the image's wavelet representation can be approximated. As we have shown, the lower a quadtree's energy, the easier it is to approximate the quadtree.

At this time it should be noted that these results do not contradict what was found in the pixel domain. In fact, they could almost be expected. This is because the lower the variance of an image, the lower the energy of its wavelet quadtrees should be.

2.3.2 Case 4

Ignoring the new scale, the case 4 results are almost identical to case 3 for both of the images. Referring to Fig. 2.3.5, *Lena* still has a huge peak right near zero while *Mandrill* is still fairly diffuse. More importantly, the plots show that higher-scale wavelet quadtrees can be well approximated by scaled lower-scale (coarser) quadtrees. This result is significant in that it forms the basis for the fractal wavelet transform, an extension of the well-known pixel-based fractal transform to the wavelet domain [9, 12, 16].

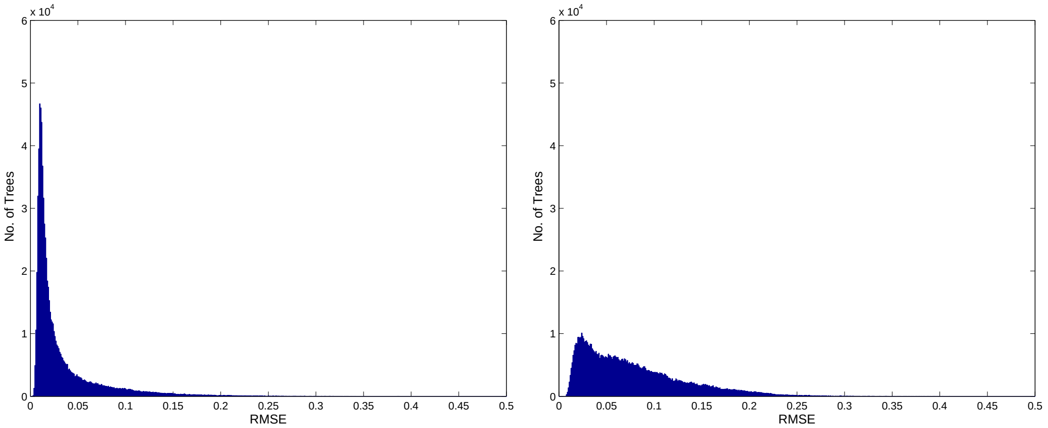


Figure 2.3.5: Case 4 wavelet-based error distributions for *Lena* (left) and *Mandrill* (right).

Finally, these results on self-similarity in the wavelet domain are not limited to *Lena* and *Mandrill*. Case 1 and 4 error distribution histograms were computed for the six other test images used in the pixel-based cases. For the case 4 histograms, see Fig. 2.3.6. The case 1 histograms were similar to the case 4 histograms although more diffuse and are omitted. Note that all of the images have their peaks quite close to zero indicating a high degree of self-similarity in the wavelet domain.

Also of significance is the degree to which (ignoring the scale) these error distributions mimic those of the pixel-based case (see Fig. 2.2.5 to compare). If any contrast can be drawn between the two sets of images, it is that in the wavelet case, the error distributions seem to be concentrated more closely about zero. The fact that these error distributions are concentrated so close to zero lends support to a wavelet-based version of the non-local means denoising algorithm. This idea will be explored in a later chapter.

2.4 Summary

In this chapter, a pixel-based model for affine self-similarity of images was introduced and extended to the wavelet representation of images. It was shown that images tend to exhibit a high degree of self-similarity with respect to the L^2 distance in both the pixel and wavelet domains. In the pixel domain, this amounted to showing

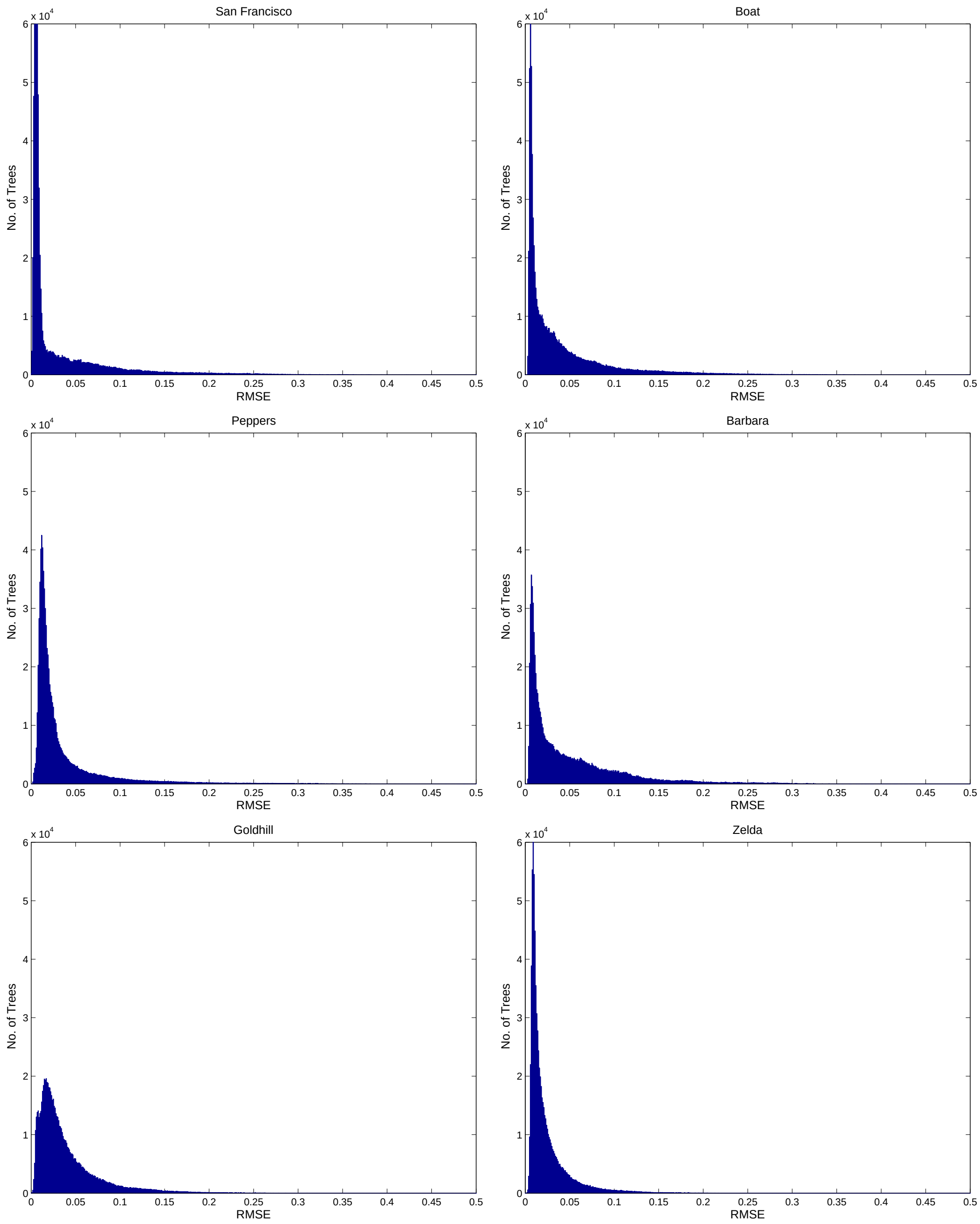


Figure 2.3.6: Case 4 wavelet-based error distributions for six other normalized test images.

that image subblocks are generally well-approximated by other (modified) image subblocks (both at the same scale and larger). In the latter context, demonstrating self-similarity involved showing that quadtrees in the wavelet representation of an image are generally well-approximated by other (modified) quadtrees, both at the same and different scales. Demonstrating that self-similarity carries over to the wavelet representation of images is an original result and it supports the extension of the non-local means denoising algorithm to the wavelet domain for the purpose of denoising wavelet quadtrees.

It was also found that self-similarity of an image is a consequence of how well its blocks (in the pixel domain) or quadtrees (in the wavelet domain) can be approximated. We showed that the lower the variance of an image subblock, the easier it was to approximate. For a quadtree, the lower its energy, the easier it was to approximate. This analysis in the wavelet domain is a new result.

Chapter 3

Image Self-Similarity and the Structural Similarity Index

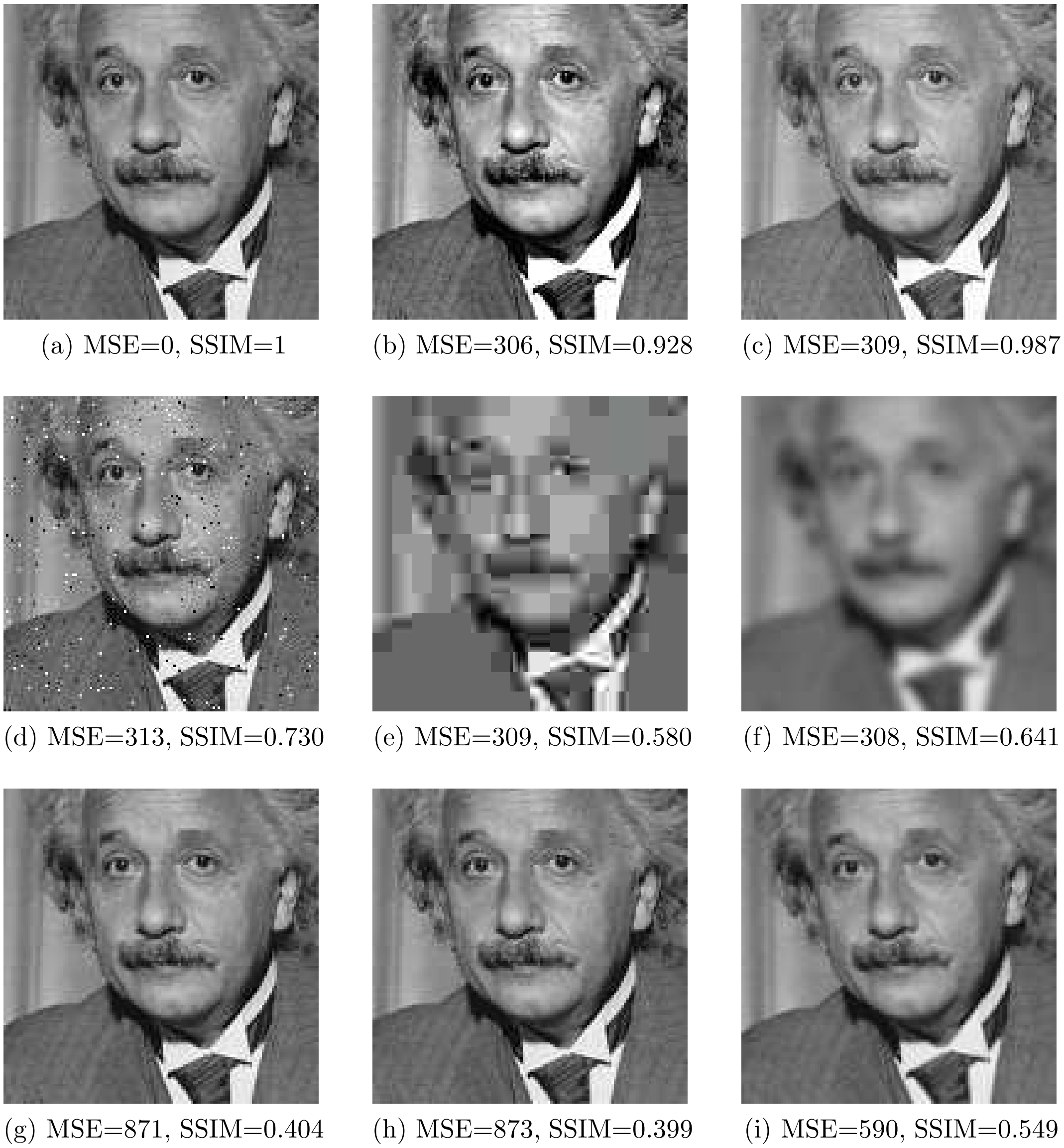
3.1 Introduction

The root mean squared error (RMSE) has been and continues to be one of the most dominantly used signal quality metrics. Indeed, the last chapter focused exclusively on the RMSE and using it to measure image similarity. However, despite its widespread usage, the RMSE fails to act as a measure of image fidelity, making it highly inappropriate to use in the context of imaging. To see this, consider the set of pictures taken from [19] in Fig. 3.1.1. They are modified versions of an original "Einstein" image (top left) created by various distortions: a contrast stretch, mean luminance shift, distortion, JPEG compression, and blurring. Along with these images are their MSE and SSIM values (relative to the original). The SSIM values will be explained shortly. Perceptually, many of the images are very different from each other and the original image but yield quite similar MSE values. On the other hand, some of the images hardly seem modified at all from the original image yet they exhibit very large MSE values. Clearly another form of measurement more suited to imaging is needed. This chapter takes a step away from the MSE to consider another measure of image similarity called the Structural Similarity Index.

3.2 The Structural Similarity Index

3.2.1 The standard SSIM Index

Introduced in [18], the Structural Similarity Index (hereafter known as the SSIM Index) is an effort to avoid potential drawbacks exhibited by more traditional forms of image similarity measurement such as the MSE. As its name implies, the SSIM Index focuses on structural features of images such as blurriness, noisiness, and blockiness. The motivation for formulating the SSIM Index in this way is that the human visual system is highly adapted to picking out structural information from images. From the standpoint of wanting an image similarity measure to mimic the



(a) MSE=0, SSIM=1 (b) MSE=306, SSIM=0.928 (c) MSE=309, SSIM=0.987

(d) MSE=313, SSIM=0.730 (e) MSE=309, SSIM=0.580 (f) MSE=308, SSIM=0.641

(g) MSE=871, SSIM=0.404 (h) MSE=873, SSIM=0.399 (i) MSE=590, SSIM=0.549

Figure 3.1.1: Comparison of MSE and SSIM values for an "Einstein" image and modified versions of it. (a) Original image. (b) Mean contrast stretch. (c) Luminance shift. (d) Impulsive noise contamination. (e) JPEG compression. (f) Blurring. (g) Spatial shift to the right. (h) Spatial shift to the left. (i) Counterclockwise rotation.

human visual system, incorporating this idea into it is a very natural one.

Another important aspect of an image quality measure is that it should incorporate Weber's law of perception. This states that visual sensitivity Σ to a change ΔI in a greyscale intensity value I is well-approximated by $\Sigma \approx \frac{\Delta I}{I}$, for $0 < I_0 < I < I_1 < \infty$. The implication of this result is that it suggests working with ratios of intensities, means, etc., when creating an image quality measure.

Given two positive N-dimensional signals x and y, the SSIM Index looks at their differences between luminance (brightness), contrasts, and structures. For luminance, average values of a signal are used. For contrast, signal variances are examined. Similarity of local patch structures is used to compare structures. An overall similarity measurement is created by combining the measurements of these three quantities in some way, for example, by multiplying them together. This is, of course, just one way of arriving at a similarity measurement. It remains to be seen whether more accurate results could be obtained by combining the measurements in a different way. With the above in mind, the SSIM function is defined as

$$S(x,y) = \frac{2\bar{x}\bar{y} + C_1}{\bar{x}^2 + \bar{y}^2 + C_1} \, \frac{2s_x s_y + C_2}{s_x^2 + s_y^2 + C_2} \, \frac{s_{xy} + C_3}{s_x s_y + C_3} \tag{3.2.1}$$

where

$$\begin{aligned} \bar{x} &= \frac{1}{N}\sum_{k=1}^{N} x_k, \\ s_{xy} &= \frac{1}{N-1}\sum_{k=1}^{N}(x_k - \bar{x})(y_k - \bar{y}), \quad \text{and} \\ s_x^2 &= \frac{1}{N-1}\sum_{k=1}^{N}(x_k - \bar{x})^2. \end{aligned}$$

The first, second, and third components of $S(x,y)$ model (respectively) luminance, contrast, and structural features. The parameters C_1, C_2, and C_3 are small constants (relative to the maximum size of the intensities) used to provide numerical stability and model the deviation from Weber's law as the intensities approach zero.

The SSIM Index has two very useful properties. First, it is symmetric: $S(x,y) = S(y,x)$. Second, it is bounded: $-1 \leq S(x,y) \leq 1$, with $S(x,y) = 1$ if and only if $x = y$. In practice, the SSIM value of an image (relative to another image) is often computed by moving a sliding window pixel-by-pixel across the images to get local SSIM estimates, and then these estimates are combined by averaging.

To see how the SSIM Index mimics the human visual system, the reader is once again referred to the Einstein pictures. The SSIM values are more in tune with our perceptual definition of similarity than the MSE values.

3.2.2 The SSIM Index in terms of real-valued wavelets

All of the work in this thesis done so far has been done in both the pixel and wavelet domains and structural similarity will be no exception. Therefore, it is necessary to extend the SSIM Index to work with wavelet detail coefficient quadtrees.

Let $x, y \in \mathbb{R}^M$, where $M = 2^K$ for some $K \geq 0$. Now consider the projections of x and y onto a set of N orthonormal zero-mean (real) wavelet basis functions $\{\psi_k \mid 1 \leq k \leq N\}$ which correspond to a quadtree with N nodes in the wavelet decompositions of x and y. For example, if the quadtree is rooted at the very top of the coefficient pyramid, $N = 2^K - 1$. This set of basis functions *does not* itself constitute a basis but rather is a subset of a basis for $\mathbb{R}^M$. The best L^2-based approximations of x and y in terms of this basis are written as

$$\begin{aligned} Px &= \sum_{k=1}^{N} a_k \psi_k, \quad \text{and} \\ Py &= \sum_{k=1}^{N} c_k \psi_k, \end{aligned}$$

where $a_k = \langle x, \psi_k \rangle$, $c_k = \langle y, \psi_k \rangle$ and $\langle\ ,\ \rangle$ denotes the dot product. It now remains to define $\overline{Px}$, s_{PxPy}, and s_{Px}^2 in terms of these expansion coefficients. First,

$$\overline{Px} = \sum_{k=1}^{N} a_k \overline{\psi}_k = 0$$

since the wavelet basis functions are zero-mean. Next,

$$\begin{aligned} s_{PxPy} &= \frac{1}{N-1} \sum_{k=1}^{N} (Px_k - \overline{Px})(Py_k - \overline{Py}) \\ &= \frac{1}{N-1} \langle Px, Py \rangle \\ &= \frac{1}{N-1} \sum_{k=1}^{N} a_k c_k. \end{aligned}$$

The last step follows because the basis functions are orthonormal. To get s_{Px}^2, let $y = x$ in the formula for s_{PxPy} to arrive at

$$s_{Px}^2 = \frac{1}{N-1} \sum_{k=1}^{N} a_k^2.$$

Now consider the SSIM function defined with zero stability constants. That is,

$$S(x, y) = \frac{4\bar{x}\bar{y}}{\bar{x}^2 + \bar{y}^2} \frac{s_{xy}}{s_x^2 + s_y^2}. \tag{3.2.2}$$

Let us now make the above substitutions in the SSIM function, remembering that since $\overline{Px} = \overline{Py} = 0$ the first component of (3.2.2) is one. Adding in the stability constants, the SSIM function can be expressed in terms of the projection coefficients of x and y as

$$S_{W,\mathbb{R}}(x,y) = \frac{\frac{2}{N-1}\sum_{k=1}^{N} a_k c_k + C}{\frac{1}{N-1}\sum_{k=1}^{N}[a_k^2 + c_k^2] + C}. \tag{3.2.3}$$

Formula (3.2.3) will be used to explore self-similarity of wavelet coefficient quadtrees.

3.3 Self-similarity of images with respect to the SSIM Index

3.3.1 Pixel-based structural self-similarity

This section explores a modified version of the pixel-based model of image self-similarity from Sec. 2.2 obtained by replacing the L^2 distance with the SSIM function $S(x,y)$. The reader is therefore referred to Sec. 2.2 for the technical details. The main point of interest here is the distribution of errors Δ_{ij} between distinct subblocks of an image $u(\mathcal{R}_i)$ and $u(\mathcal{D}_j)$ in terms of the SSIM function. Affine transformations of the $u(\mathcal{D}_j)$ are considered and these take the form $\phi(t) = \alpha t + \beta$. If x and y denote two N-dimensional signals, the parameters α and β are chosen to maximize $S(x, \alpha y + \beta)$. A detailed derivation for these coefficients may be found in Appendix B.4 and it turns out that the optimal values are

$$\alpha = \operatorname{sgn}(s_{xy})\frac{s_x}{s_y} \quad \text{and} \quad \beta = \bar{x} - \alpha\bar{y}.$$

There are four cases involving optimization over α and β as well as decimations:

1. Purely translational: The domain and range blocks $\mathcal{D}_j$ and $\mathcal{R}_i$ have the same size. The $w_{ij}^{(k)}$ are translations. Set $\alpha = 1$ and $\beta = 0$.

2. Translational and greyscale shift: The domain and range blocks have the same size. The $w_{ij}^{(k)}$ are translations. Set $\alpha = 1$ and optimize over β.

3. Affine, same-scale: The domain and range blocks have the same size. The $w_{ij}^{(k)}$ are translations. Optimize over α and β.

4. Affine, two-scale: The domain blocks are larger than the range blocks. The affine transformations $w_{ij}^{(k)}$ incorporate a decimation in pixel space. Optimize over α and β.

The following experiments were performed with stability constants $C_1 = C_2 = C_3 = 30$ and used unmodified 8-bit images. For each domain and range block pairing, the error reported is

$$\Delta_{ij} = 1 - S(u(\mathcal{R}_i), \alpha u(\mathcal{D}_j) + \beta).$$

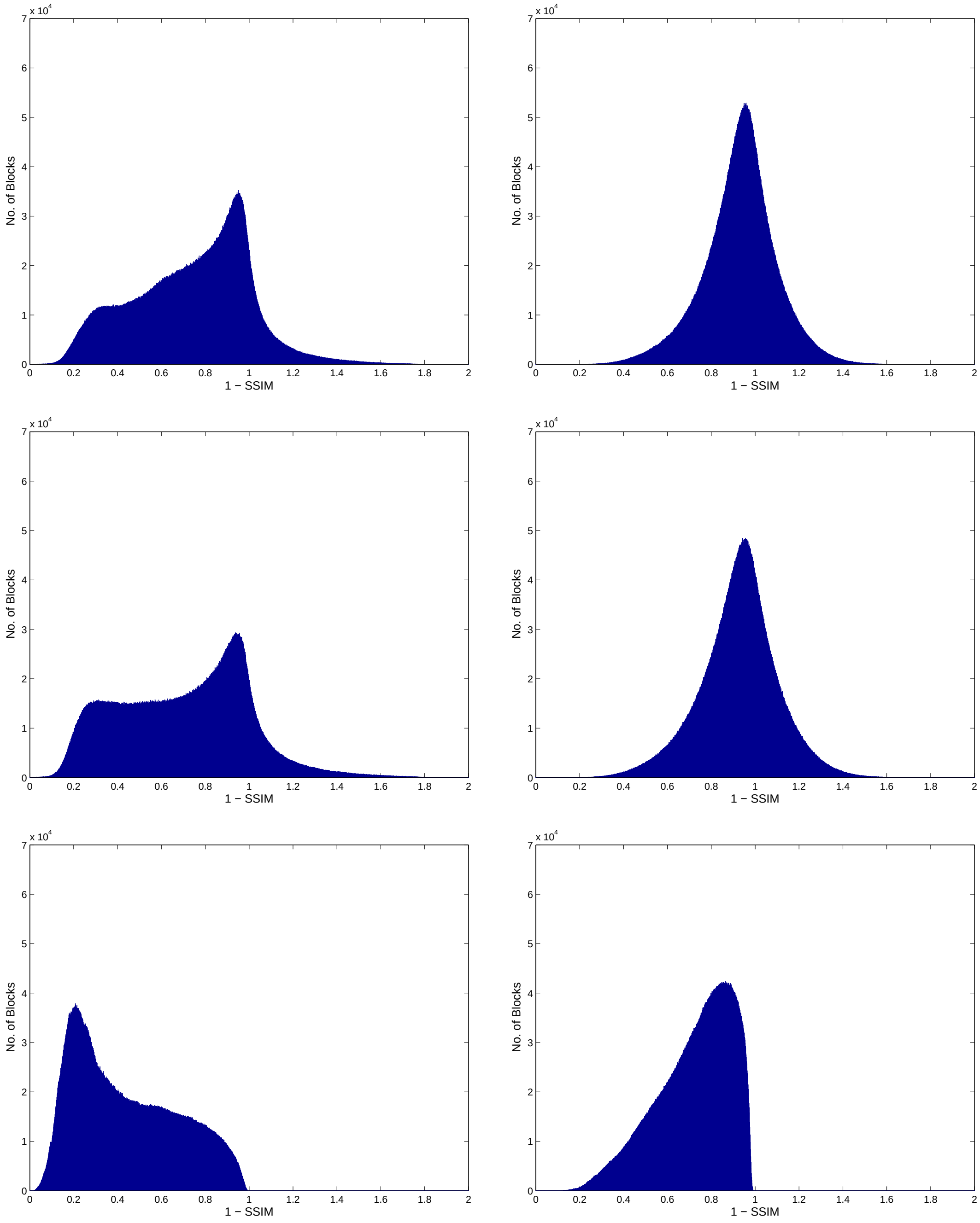


Figure 3.3.1: Case 1-3 pixel-based error distributions of 1-S for *Lena* (left) and *Mandrill* (right). From top to bottom are case 1, case 2 and case 3.

Before discussing the experimental results, it should be emphasized that the SSIM Index is a measure of *structural* similarity. Therefore, for an image to be highly self-similar with respect to the SSIM Index, it should be periodic or there should be a repetition of structures in the image. For a general image, this may be an unreasonable expectation. Finally note that since $0 \leq 1 - S \leq 2$, the more closely concentrated the error distribution histogram for an image is about zero, the more (structurally) self-similar the image is.

Fig. 3.3.1 shows that *Lena* and *Mandrill* do not exhibit a high degree of structural self-similarity. Allowing for optimization over α and β pushes the distributions of $1 - S$ closer to zero, more so for *Lena*. On the other hand, the *Mandrill* histograms remain largely concentrated around one in all three cases. In the case of *Lena*, as more parameters are allowed to vary, peaks develop close to zero (and so S is close to one) but a large portion of the errors are still clustered about $S = 0$. That being said, the C_i were not toyed with extensively so perhaps better results could be obtained while still employing reasonable values for the stability constants.

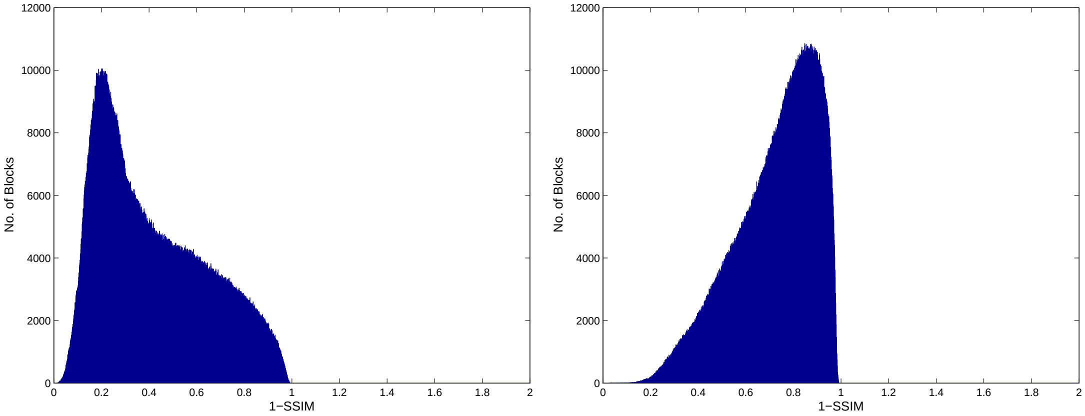


Figure 3.3.2: Case 4 pixel-based distributions of 1-S for *Lena* (left) and *Mandrill* (right).

For completeness, the case 4 histograms for *Lena* and *Mandrill* are included in Fig. 3.3.2. Note that these histograms are nearly identical to the case 3 histograms. This was also the case for the other six test images so their histograms are omitted.

Fig. 3.3.3 shows the case 1 error histograms for six other test images. We refrain from commenting on them at this point. These histograms will be compared to wavelet-based structural similarity histograms in the next section.

The case 3 histograms (Fig 3.3.4) were computed for six other test images to see if the trends seen so far are consistent and indeed they are. These error histograms exhibit peaking in a variety of areas, both close to and away from zero. With the exceptions of *Barbara* and *Goldhill*, the largest peaks seem to be situated near zero but many of the values fall too far away from zero for these images to be considered self-similar with respect to the SSIM Index.

3.3.2 Structural vs. L^2-based self-similarity of image subblocks

The last section showed that images don't exhibit a high degree of structural self-similarity. As it turns out, this can be explained in terms of the function $1 - S$ that

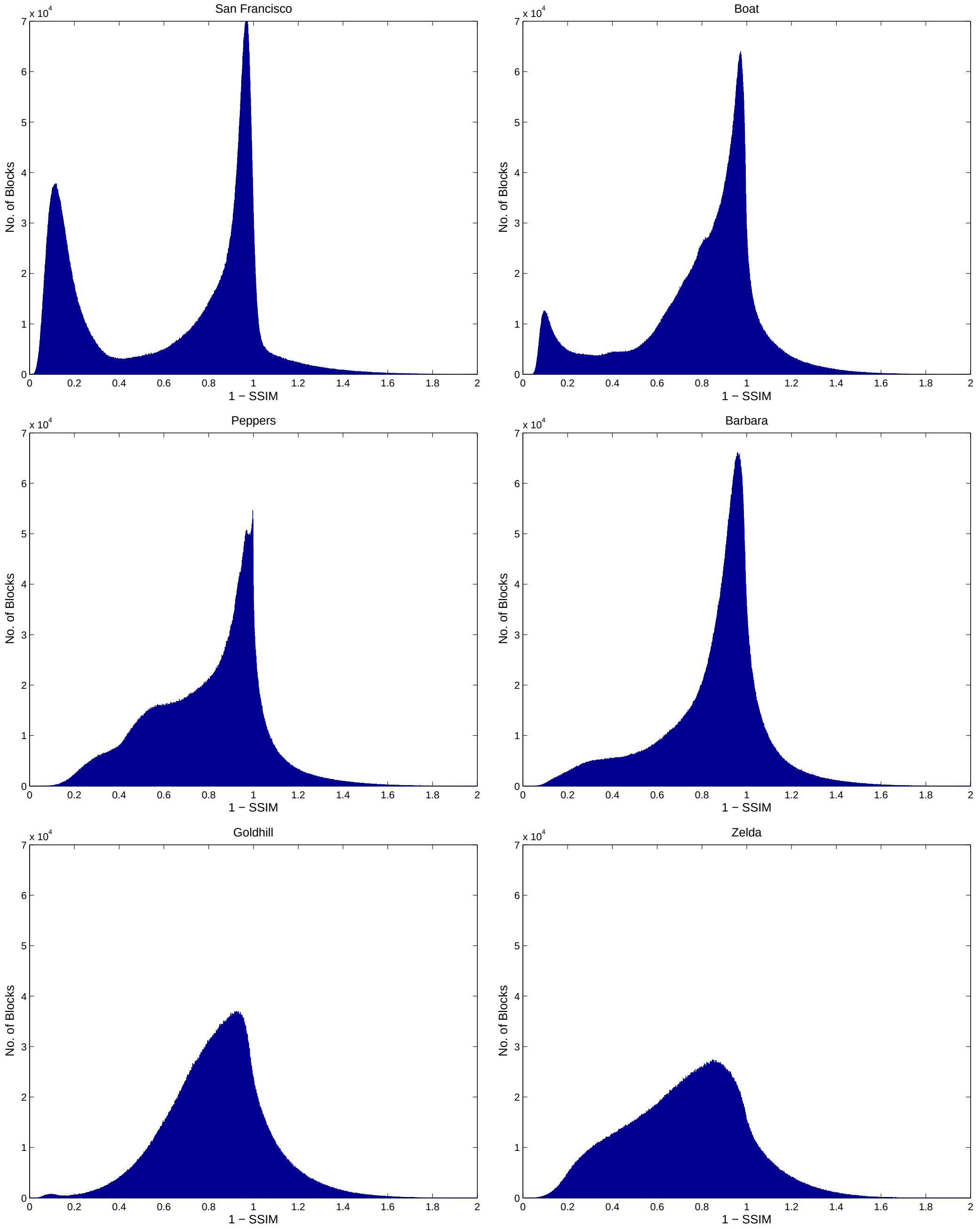


Figure 3.3.3: Case 1 pixel-based distributions of $1 - S$ for six other test images.

was plotted in the preceding histograms. Let $x, y \in \mathbb{R}^N$ be two signals and consider $1 - S(x, \alpha y + \beta)$ where α and β are the coefficients that maximize $S(x, \alpha y + \beta)$. For cases 2 and 3, these coefficients were given by

1. Case 2: $\alpha = 1$ and $\beta = \bar{x} - \bar{y}$.

2. Case 3: $\alpha = \text{sgn}(s_{xy})\frac{s_x}{s_y}$ and $\beta = \bar{x} - \alpha\bar{y}$.

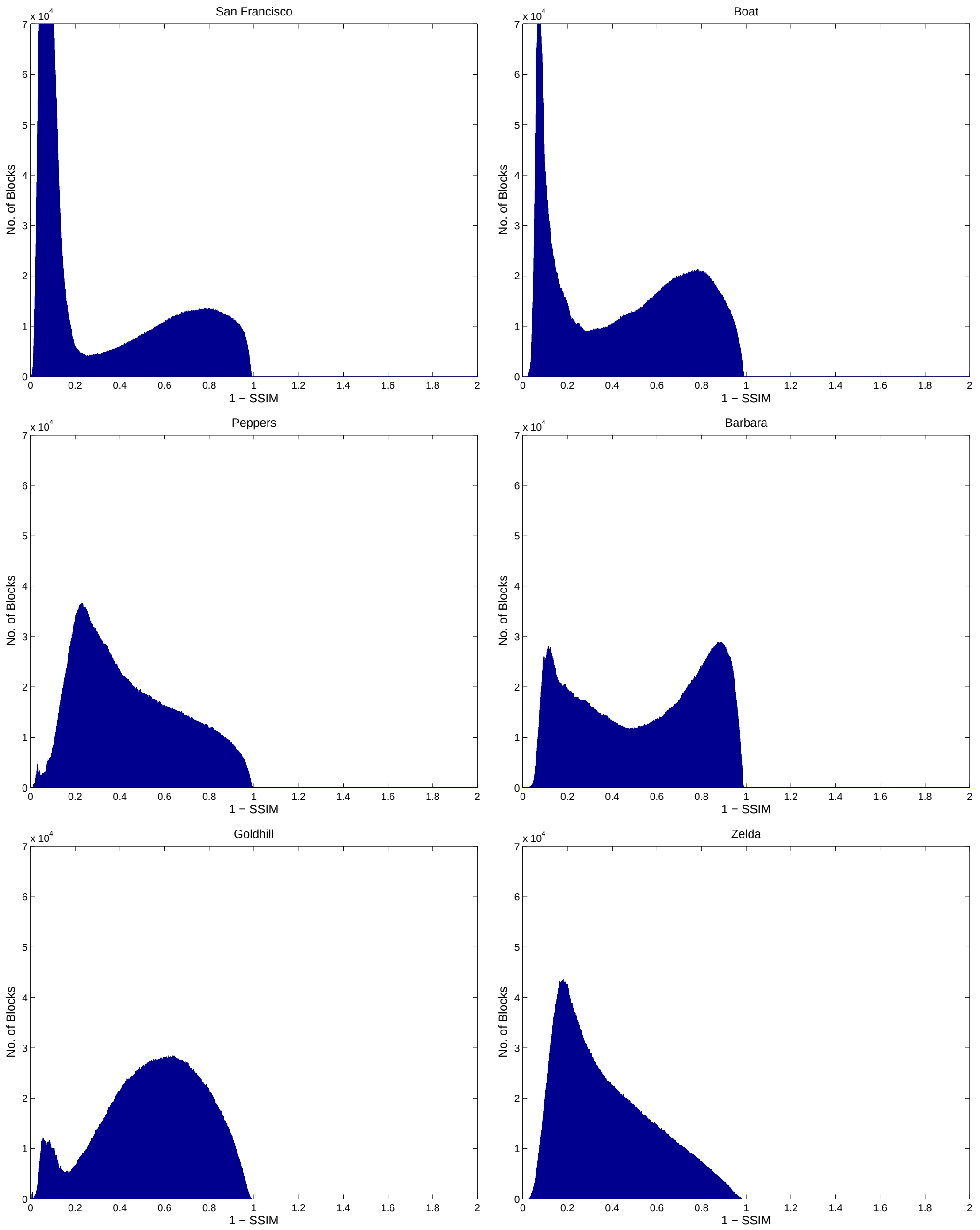


Figure 3.3.4: Case 3 pixel-based distributions of $1 - S$ for six other test images.

Let $u = x$ and $v = \alpha y + \beta$ and for simplicity, consider the SSIM function with zero stability constants. By the choices for α and β, $\bar{u} = \bar{v}$. Since the means are equal,

$$1 - S(u, v) = 1 - \frac{2s_{uv}}{s_u^2 + s_v^2} = \frac{s_u^2 + s_v^2 - 2s_{uv}}{s_u^2 + s_v^2} = \frac{1}{N-1}\frac{\|u - v\|_2^2}{s_u^2 + s_v^2}. \qquad (3.3.1)$$

Originally derived in [6], this result shows that $1 - S(u, v)$ is an inverse variance-weighted squared L^2 distance. In fact, even more is true. From [5], $\sqrt{1 - S(x, y)}$ is a metric when the means of the two signals are matched like above.

Recall that when examining self-similarity of images in terms of the L^2 distance there was a strong bias towards image subblocks of low variance and that it was easier to approximate low-variance subblocks. From (3.3.1), we see that it is exactly this variance property that keeps the distributions of $1 - S$ away from zero in cases 2 and 3. This then gives another interpretation for the self-similarity of images. That is, perhaps images are not as self-similar as we once thought.

3.3.3 Wavelet-based structural self-similarity

Now self-similarity in the wavelet domain with respect to the SSIM Index will be investigated using the wavelet-based version of the SSIM function,

$$S_{W,\mathbb{R}}(x,y) = \frac{\frac{2}{N-1}\displaystyle\sum_{k=1}^{N} a_k c_k + C}{\frac{1}{N-1}\displaystyle\sum_{k=1}^{N} [a_k^2 + c_k^2] + C}.$$

Consider then the standard tensor-product (real) wavelet basis expansion of a $2^K \times 2^K$ image for which the detail coefficients can be arranged into a pyramid of blocks at each decomposition level. At level k of this decomposition, $0 \leq k \leq K$, the coefficients can further be partitioned into horizontal, vertical, and diagonal detail coefficients which will be denoted by A_k^h, A_k^v, and A_k^d respectively. Given a wavelet coefficient $a_{kij}^{\lambda}, \lambda \in \{h, v, d\}$, let A_{kij}^{λ} denote the unique quadtree rooted at a_{kij}^{λ}.

The concept of self-similarity is extended to the wavelet domain by examining how well quadtrees in a wavelet decomposition of an image are approximated by other quadtrees, both at the same and different scales. These approximations will be of the form $A_{kij}^{\lambda} \approx \alpha A_{k'i'j'}^{\lambda'}, 0 \leq k' \leq k$. Since wavelet quadtrees are being compared, the a_k and c_k will be the wavelet coefficients of these quadtrees. Three cases will be explored:

1. Wavelet quadtrees at the same scale are compared. Set $\alpha = 1$.

3. Affine, one-scale: Quadtrees at the same scale are compared. Optimize over α.

4. Affine, two-scale: Let $k' < k$. Quadtrees at higher scales are approximated using quadtrees at lower scales. This is equivalent to a pixel decimation operation. Optimize over α.

The cases above are numbered so that they may roughly be considered as counterparts to the cases considered in the pixel domain. To find the optimal α, maximize $S_{W,\mathbb{R}}(x, \alpha y)$ with respect to α. As is shown in Appendix B.5, this gives

$$\alpha = \operatorname{sgn}\left(\sum_{k=1}^{N} a_k c_k\right) \sum_{k=1}^{N} a_k^2 \bigg/ \sum_{k=1}^{N} c_k^2 \,.$$

In the calculations that follow, wavelet quadtrees of unmodified 8-bit images with same orientation are compared (that is, $\lambda = \lambda'$). The errors reported are

$$^{\lambda}\Delta^{kij}_{k'i'j'} = 1 - S(A^{\lambda}_{kij}, \alpha A^{\lambda}_{k'i'j'}).$$

For the numerical experiments, we set $k = 6$ (there are three decompositions total) and for case 4, $k' = 3$. The Haar wavelet is used. Like before, the stability constant C is fixed at 30.

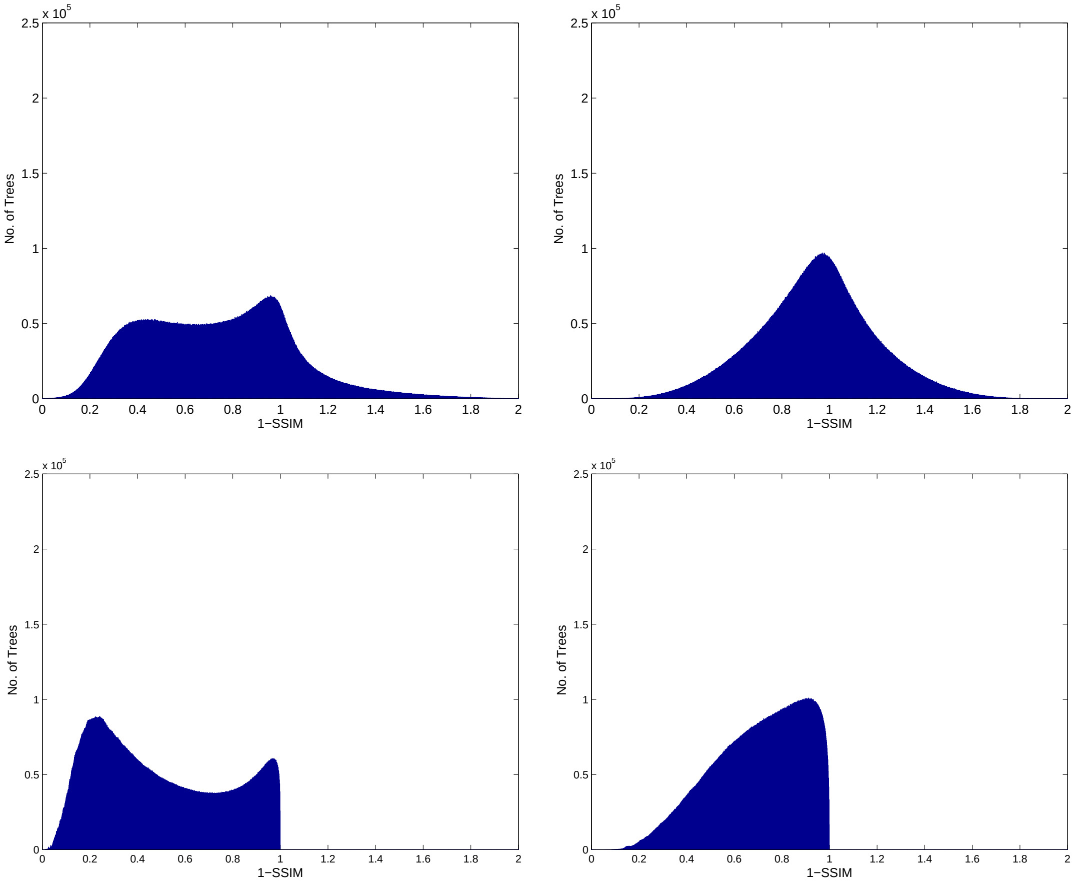


Figure 3.3.5: Cases 1 and 3 wavelet-based distributions of $1 - S$ for *Lena* (left) and *Mandrill* (right). The top row is case 1 and the bottom row is case 3.

First, Fig. 3.3.5 shows the histogram distributions of $1 - S$ (cases 1 and 3) for *Lena* and *Mandrill*. These histograms show significant peaking away from zero which indicates that *Lena* and *Mandrill* do not exhibit a high degree of self-similarity in the wavelet domain. Even though allowing α does yield an improvement, it is not enough to change this conclusion. Next, compare with Fig. 3.3.1 which contains some of the pixel-based distributions of $1-S$ for *Lena* and *Mandrill*. The two sets of histograms have generally the same shape, but the pixel-based ones seem to be more concentrated about zero. This indicates that quadtrees in the wavelet representation of an image generally exhibit a lower degree of structural self-similarity than the original image subblocks.

Once again, plots of the α histogram distributions for case 3 are interesting to look at. These histograms for *Lena* and *Mandrill* are included in Fig. 3.3.6. The *Lena* histogram exhibits a huge peak around zero and the *Mandrill* histogram appears to be a more diffuse version of *Lena's*.

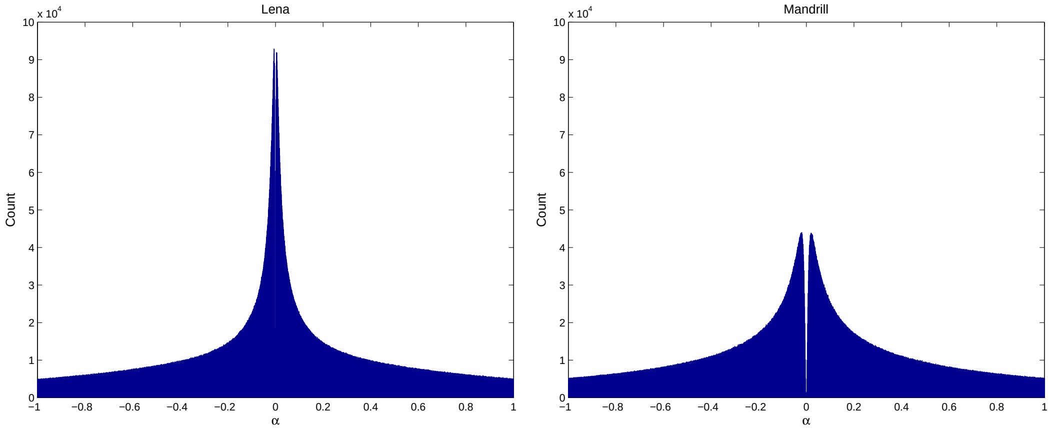


Figure 3.3.6: Histogram distributions of the parameter α in case 3 for *Lena* (left) and *Mandrill* (right).

For case 4, as Fig. 3.3.7 shows, approximating quadtrees with scaled quadtrees higher up in the wavelet coefficient tree does not yield significant improvements for *Lena* and *Mandrill*. The histograms for the other six test images are omitted here but they indicated the same thing. In general, the case 4 distributions tend to mimic the case 3 results, and there is still a large amount of peaking away from zero.

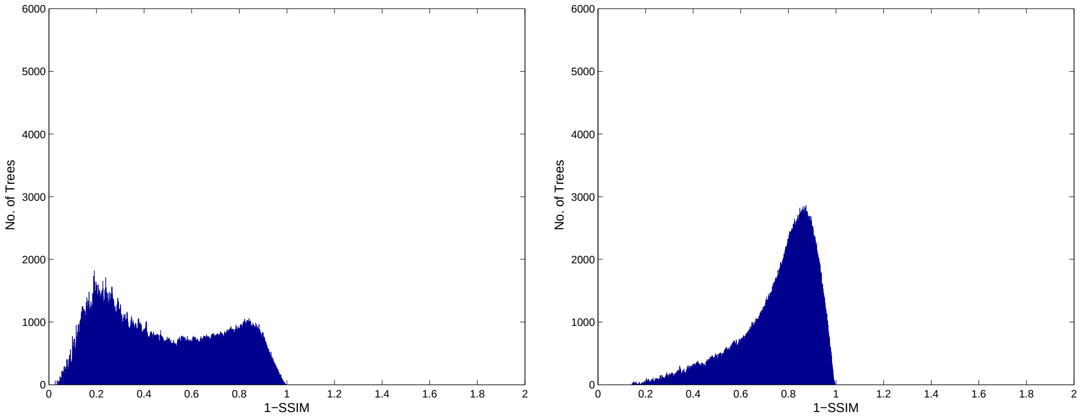


Figure 3.3.7: Case 4 wavelet-based distributions of $1-S$ for *Lena* (left) and *Mandrill* (right).

Fig. 3.3.8 shows the case 1 histograms of $1 - S$ for the six other test images. Note the similarity to the histograms found in Fig. 3.3.3. As with *Lena* and *Mandrill*, the wavelet-based histograms tend to be more concentrated around zero than their pixel-based counterparts. That being said, there is still a very significant concentration of errors about $S = 0$ in the wavelet-based case. This further evidences the fact that structural self-similarity of images is not a feature generally present in the wavelet domain either.

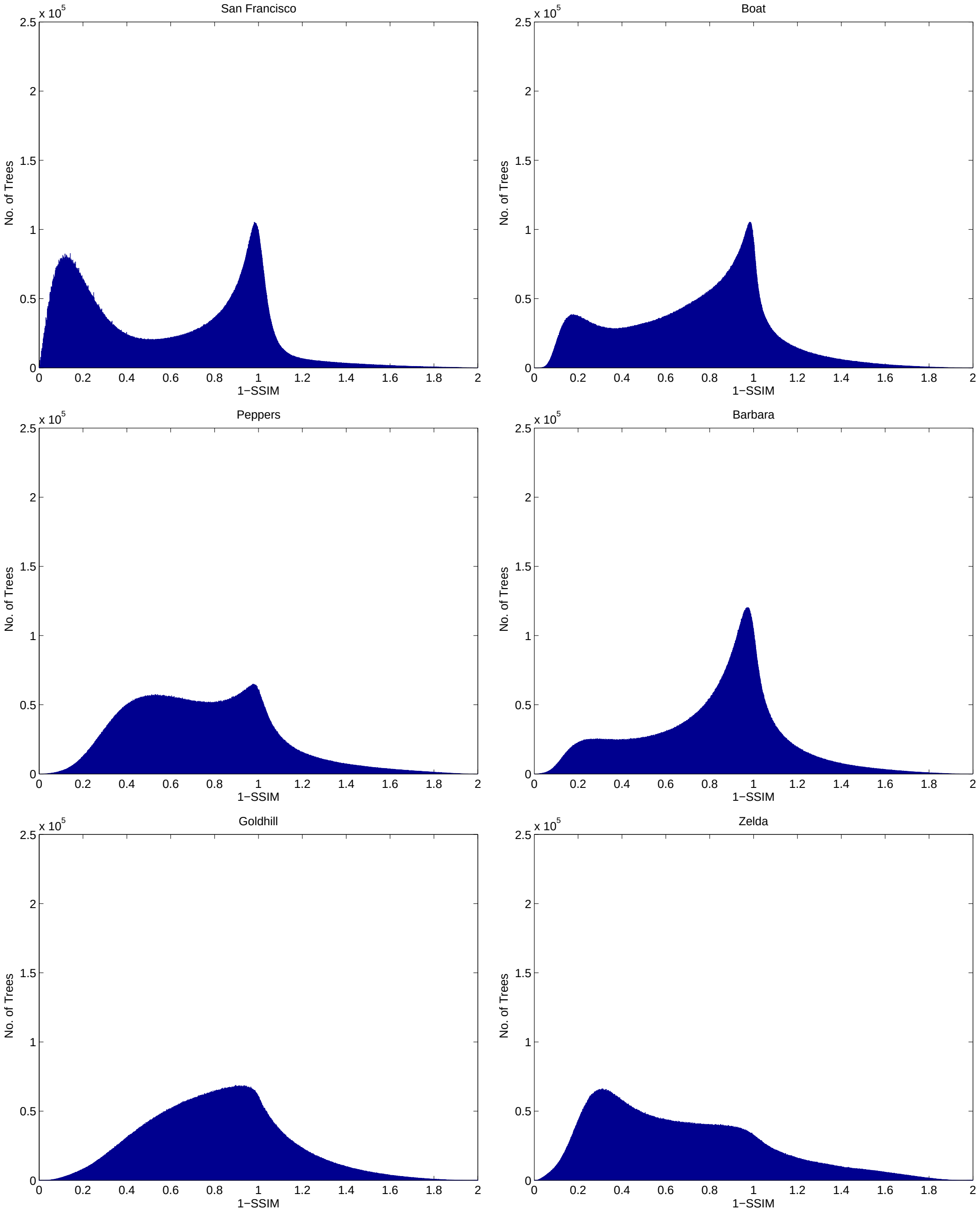


Figure 3.3.8: Case 1 wavelet-based distributions of $1 - S$ for six other test images.

Finally, the case 3 error distributions for the other test images are shown in Fig. 3.3.9. They show that all of the distributions of $1 - S$ get pushed towards zero when α is allowed to vary. Still though, for several of the images (most notably

San Francisco, *Boat* and *Barbara*), there are large peaks around one, indicating a generally low degree of structural similarity between distinct quadtrees.

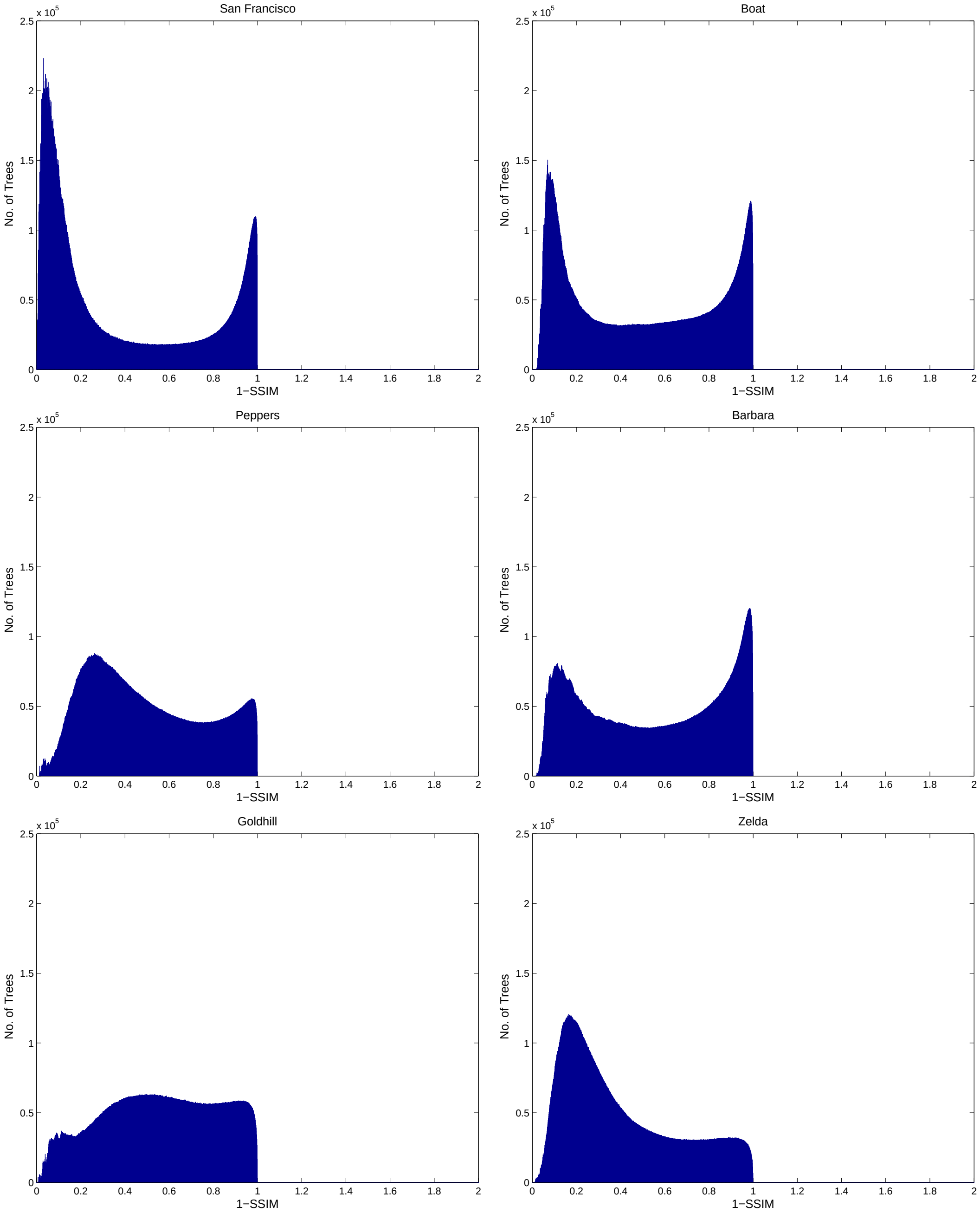


Figure 3.3.9: Case 3 wavelet-based distributions of $1 - S$ for six other test images.

3.3.4 Structural vs. L^2-based self-similarity of wavelet quadtrees

In the pixel domain, the poor degree of structural similarity exhibited by all of the test images was explained by showing that if two signals $x, y \in \mathbb{R}^N$ have equal means then $1 - S(x,y)$ is an inverse variance-weighted squared L^2 distance. Now consider

$$1 - S_{W,\mathbb{R}}(x,y) = 1 - \frac{\frac{2}{N-1}\sum_{k=1}^{N} a_k c_k + C}{\frac{1}{N-1}\sum_{k=1}^{N}[a_k^2 + c_k^2] + C} = \frac{\frac{1}{N-1}\sum_{k=1}^{N}(a_k - c_k)^2}{\frac{1}{N-1}\sum_{k=1}^{N}[a_k^2 + c_k^2] + C}. \qquad (3.3.2)$$

This shows that $1 - S_{W,\mathbb{R}}(x,y)$ is an inverse energy-weighted L^2 distance. In the last chapter, when examining self-similarity of wavelet quadtrees with respect to the L^2 distance, it was shown that a strong bias existed towards quadtrees having low energy. From (3.3.2) then, if a quadtree has low energy, $1 - S$ should be pushed away from zero. To summarize, the lower the energy of the quadtrees from an image, the less structurally self-similar the image will be in the wavelet domain.

3.4 Summary

In this chapter the Structural Similarity Index was introduced and examined in detail. Previously an unexplored topic, it was shown that images are not generally self-similar with respect to the structural similarity index in both the pixel and wavelet domains. In the pixel domain, this was explained by showing that $1-S(x,y)$ is an inverse variance-weighted squared L^2 distance when $\bar{x} = \bar{y}$. The implication of this is that the lower the variance of subblocks in an image (and therefore the more self-similar it is with respect to the L^2 distance), the less self-similar the image will be with respect to the SSIM Index. Given that the SSIM Index seems to be much more accurate than the L^2 error when it comes to measuring image quality, this offers new insight into the concept of image self-similarity. That is, perhaps images are not as self-similar as the results obtained using L^2 lead us to believe.

To explore structural similarity of wavelet detail coefficient quadtrees, we derived a new form of the SSIM function in terms of real-valued wavelet basis functions and denoted it by $S_{W,\mathbb{R}}(x,y)$. To explain the lack of structural similarity in the wavelet domain, it was shown that $1 - S_{W,\mathbb{R}}(x,y)$ is an inverse energy-weighted squared L^2 distance. As a result, the lower the energy of the wavelet quadtrees in an image (and the more self-similar it is with respect to the L^2 distance), the less self-similar it will be with respect to the SSIM Index. The work done in the wavelet domain is completely original.

Chapter 4

Non-local image processing

4.1 Introduction

Let u be an image function. Non-local image processing modifies part of the image, $u(x)$, with other parts of the image, $u(y_i)$, where the points y_i are not necessarily close to x. This is in contrast to many local methods of image processing which are based on the values of neighbouring pixels of x such as local averaging. This chapter focuses on a non-local procedure called non-local means denoising.

4.2 Non-local means denoising in the pixel domain

4.2.1 The non-local means denoising alorithm

Non-local (NL) means denoising [7] is an attempt to restore a noisy digital image by using information from different parts of the image. This is done by computing an estimate for a denoised pixel as a weighted sum of the other pixels in the noisy image. Consider an image defined on a grid I which will be denoted by $u = \{u(i) \mid i \in I\}$. Next, let $n = \{n(i) \mid i \in I\}$ be a noisy image created by adding independent and identically distributed (i.i.d.) samples of zero-mean noise to u. The value of a pixel in the noiseless image is computed as

$$d(i) = \sum_{j \in I} w(i,j)n(j).$$

The weights satisfy $0 \leq w(i,j) \leq 1$ and $\sum_{j \in I} w(i,j) = 1$. They are determined by the similarity between windows (image subblocks) about pixels i and j which will be denoted by N_i and N_j respectively. The reason for approaching the denoising process with averaging is that because the noise is zero-mean, when large numbers of samples are averaged over, the noise will average out to zero.

To compare windows, the L^2 distance is used. Roughly, the reason the L^2 distance works so well is that it preserves the order of similarity between pixels [10].

Therefore, the most similar pixels to a pixel p in n are expected to be the most similar pixels to p in u. The weights are given by the formula

$$w(i,j) = \frac{1}{Z_i} \exp\left(-\frac{\|N_i - N_j\|_2^2}{h^2} \right),$$

where

$$Z_i = \sum_{j \in I} \exp\left(-\frac{\|N_i - N_j\|_2^2}{h^2} \right).$$

The parameter h controls the decay of the weights. From the formula for $w(i,j)$ it is clear that the smaller $\|N_i - N_j\|_2^2$ is the larger the weight $w(i,j)$, and that the smaller h is, less pixels contribute significantly to the weighted sum.

NL means denoising exploits the fact that images tend to exhibit a high degree of self-similarity with respect to the L^2 distance, a topic explored extensively in Chapter 2. The goal of the first part of this chapter is to generalize the usual NL means algorithm by allowing for affine transformations of image subblocks. Once again, these transformations take the form $\phi(t) = \alpha t + \beta$. The value of a pixel in a noiseless image will be estimated as

$$d(i) = \sum_{j \in I} w(i,j)(\alpha n(j) + \beta)$$

The weights satisfy $0 \leq w(i,j) \leq 1$ and $\sum_{j \in I} w(i,j) = 1$. However, they are now calculated as

$$w(i,j) = \frac{1}{Z_i} \exp\left(-\frac{\|N_i - \alpha N_j - \beta\|_2^2}{h^2} \right),$$

where

$$Z_i = \sum_{j \in I} \exp\left(-\frac{\|N_i - \alpha N_j - \beta\|_2^2}{h^2} \right).$$

The parameters α and β are chosen to minimize

$$\Delta_{ij} = \|N_i - \alpha N_j - \beta\|_2.$$

Three cases of the transformation $\phi(t) = \alpha t + \beta$ and their effects on NL means denoising will be explored:

1. Purely translational: Set $\alpha = 1$ and $\beta = 0$. This is the traditional form of the NL means denoising algorithm.

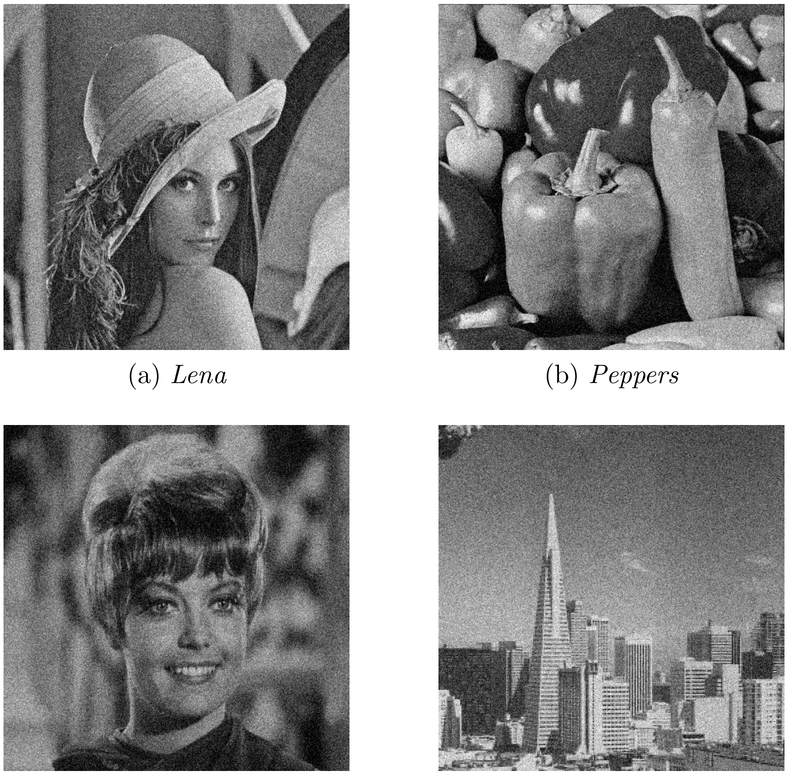
2. Translational and greyscale shift: Set $\alpha = 1$ and optimize over β.

3. Affine, same-scale: Optimize over both α and β.

In the experiments that follow, noisy versions of the test images were created by adding $\mathcal{N}(0, 0.01)$ white noise to the original. Here, $\mathcal{N}(\mu, \sigma^2)$ denotes the normal probability distribution with mean μ and variance σ^2. Fig. 4.2.1 shows four of these noisy test images and Table 4.2.1 shows the RMSE and SSIM values (relative to their respective originals) for these noisy images and four others.

	Lena	*Mandrill*	*Goldhill*	*Boat*
RMSE	0.0990	0.0996	0.0986	0.0985
SSIM	0.2587	0.5381	0.3209	0.3117
	Barbara	*Peppers*	*San Fran.*	*Zelda*
RMSE	0.0990	0.0968	0.0999	0.0978
SSIM	0.3773	0.2618	0.2563	0.1968

Table 4.2.1: RMSE and SSIM values relative to their originals for eight noisy test images.



(a) *Lena*

(b) *Peppers*

(c) *Zelda*

(d) *San Francisco*

Figure 4.2.1: Four noisy test images created by adding $\mathcal{N}(0, 0.01)$ white noise to the originals.

The generalized NL means denoising algorithm proposed above will be considered in two different ways: the first is the traditional pixel-based way already described. The second is a block-based method useful primarily for its lower computational cost. The second method is inspired by the idea that since a denoised pixel value can be estimated with a weighted sum of the other pixels in the noisy image, the same method should work for image subblocks. In other words, letting

$n(B_i)$ denote a block of pixels centred at pixel i in the noisy image n, the denoised image subblock will be estimated by

$$d(B_i) = \sum_{j \in I} w(B_i, B_j)(\alpha n(B_j) + \beta).$$

The weights are calculated in an analogous manner.

4.2.2 Experimental results

In the experimentation process, several noisy test images were processed using the proposed generalized NL means denoising algorithm under the three cases discussed above. For these experiments, normalized 8-bit greyscale images were used and the parameter $h = 0.5$.

For the pixel-based algorithm, 7 × 7 similarity neighbourhoods centred at each of the pixels were used. Fig. 4.2.2 shows the pictoral results for *Lena*, *Peppers*, and *Zelda* and Table 4.2.2 gives the RMSE and SSIM values for eight denoised test images including the three just mentioned. First notice that for both *Peppers* and *Zelda* the RMSE and SSIM values decrease and increase respectively as the case number is increased. For *Lena*, however, only when moving from case 1 to case 2 is a decrease in RMSE and increase in SSIM observed. This seems to be a trend as the calculations for the other five test images show. The RMSE and SSIM values in case 3 seem to be worse than for case 2.

Pixel-based results					
Case		*Lena*	*Mandrill*	*Goldhill*	*Boat*
1	RMSE SSIM	0.0404 0.4558	0.0745 0.6275	0.0464 0.5020	0.0476 0.4427
2	RMSE SSIM	0.0370 0.4985	0.0716 0.6548	0.0424 0.5463	0.0444 0.4675
3	RMSE SSIM	0.0352 0.4925	0.0717 0.5205	0.0422 0.4876	0.0433 0.4307
		Barbara	*Peppers*	*San Fran.*	*Zelda*
1	RMSE SSIM	0.0495 0.5703	0.0447 0.4617	0.0466 0.3489	0.0342 0.5040
2	RMSE SSIM	0.0458 0.6050	0.0416 0.5054	0.0449 0.3527	0.0300 0.5688
3	RMSE SSIM	0.0477 0.5593	0.0397 0.5162	0.0443 0.3245	0.0280 0.6077

Table 4.2.2: RMSE and SSIM values relative to their respective originals for eight images produced by pixel-based NL means denoising.

Interestingly, allowing for optimization over only β (case 2) seems to preserve more details in the image compared to the strictly translational model. For example, look at the band on the hat in *Lena* when moving from case 1 to case 2. In case

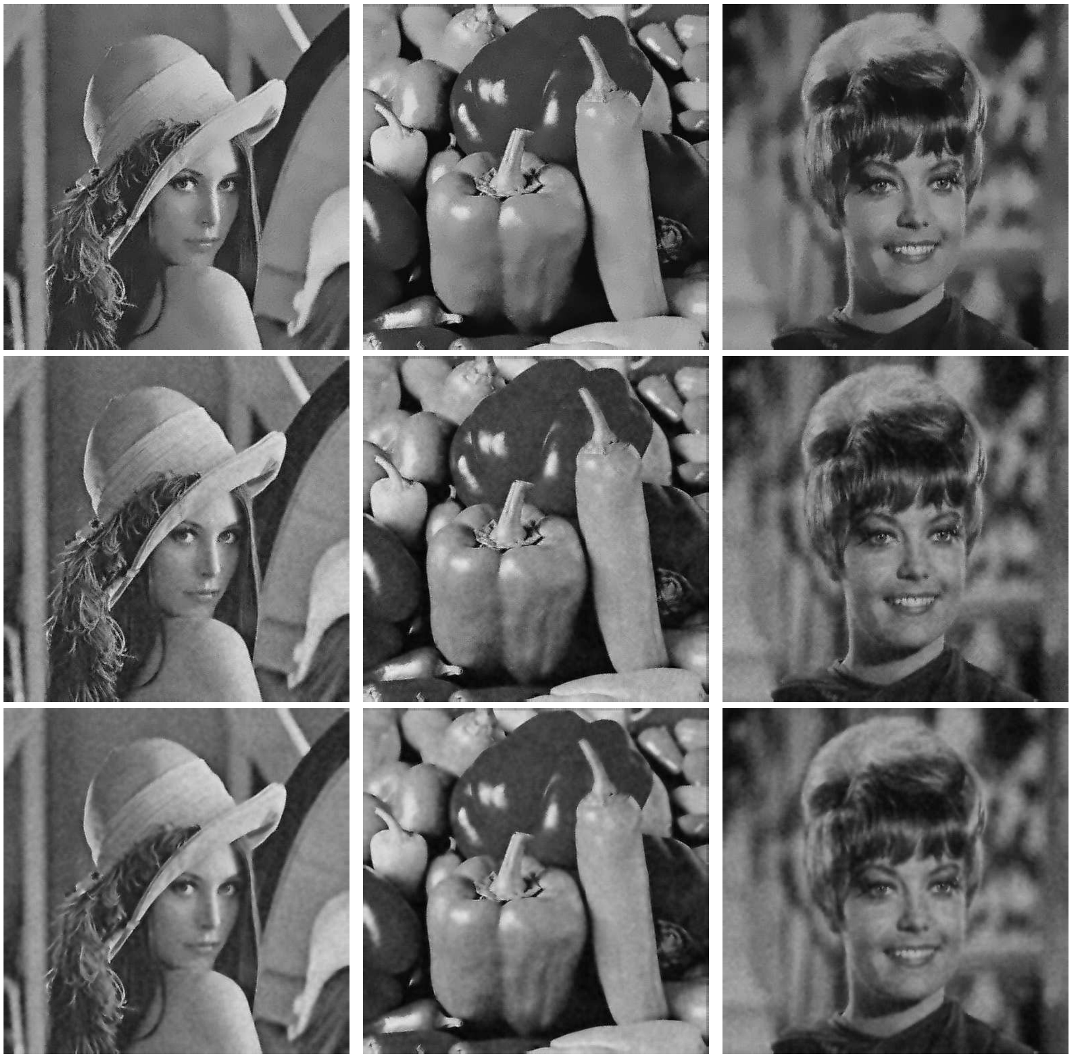


Figure 4.2.2: Denoised *Lena*, *Peppers*, and *Zelda* images created using a pixel-based NL means denoising algorithm. The first row is case 1, the second is case 2, and the third is case 3.

1, the band is more washed out and of uniform texture while in cases 2 and 3, more lines and folds are visible. However, along with this heightened detail comes blurring. The blurring is even worse in case 3. This blurring causes a decrease in the SSIM values from case 2 to case 3 for most of the test images. Also notable is the presence of some sort of spurious texture across the images in cases 2 and 3 which is not present in the original image.

One may question whether the choice of h is optimal for the denoising process. Additional tests were performed to see if any better results could be achieved by allowing h to vary between 0.35 and 1, and the results were similar or inferior to what was found above. Therefore, judging by the image quality (both perceptual and mathematical), when implementing the generalized NL means denoising algorithm, there is a benefit to considering greyscale shifts, but it is generally detrimental to consider affine transformations.

Let us now make the transition to a block-based version of the generalized NL means denoising algorithm. Here an attempt is made to estimate an 8 × 8 denoised image subblock by writing it as a weighted sum of all non-overlapping 8 × 8 subblocks in the noisy image. The non-overlapping condition is chosen for numerical efficiency.

Block-based results					
Case		*Lena*	*Mandrill*	*Goldhill*	*Boat*
1	RMSE	0.0509	0.0785	0.0554	0.0581
	SSIM	0.3945	0.6017	0.4553	0.3917
2	RMSE	0.0441	0.0692	0.0481	0.0508
	SSIM	0.363	0.6135	0.4545	0.4024
3	RMSE	0.0460	0.0704	0.0515	0.0518
	SSIM	0.4048	0.5582	0.3959	0.3946
		Barbara	*Peppers*	*San Fran.*	*Zelda*
1	RMSE	0.0634	0.0537	0.0548	0.0416
	SSIM	0.4784	0.3894	0.3165	0.393
2	RMSE	0.0553	0.0465	0.0479	0.0367
	SSIM	0.5053	0.5317	0.3350	0.4777
3	RMSE	0.0573	0.0477	0.0490	0.0383
	SSIM	0.4808	0.4166	0.3427	0.4509

Table 4.2.3: RMSE and SSIM values relative to their respective originals for eight images produced by block-based NL means denoising.

Fig. 4.2.3 and Table 4.2.3 show the results of the block-based experiments for *Lena*, *Peppers*, and *San Francisco*. The first column is case 1, the second column is case 2, and the third column is case 3. *San Francisco* is included here because for the block-based algorithm, it was the only image that achieved its best SSIM value in case 3. However, as in the pixel-based case, we see that case 2 is generally the best according to the RMSE and SSIM values relative to the original image. That being said, case 1 suffers from less blockiness than cases 2 and 3. Also, judging by the presence of more lines on the band of the hat in *Lena* and the quality of the windows on the buildings in *San Francisco*, case 1 seems to preserve details the best.

The blockiness seen here could be improved by considering smaller blocks at a larger computational cost. In an effort to improve the case 3 results, h was allowed to vary between 0.35 and 1. Unfortunately, no significant improvements were observed.

To summarize, for all but one test image (*Goldhill*), an increase in the SSIM values was observed when moving from case 1 to case 2. Only once was an increase in SSIM observed when moving from case 2 to case 3. Therefore, the same conclusion as in the pixel-based case is reached: allowing for affine transformations is generally detrimental to the denoising process.

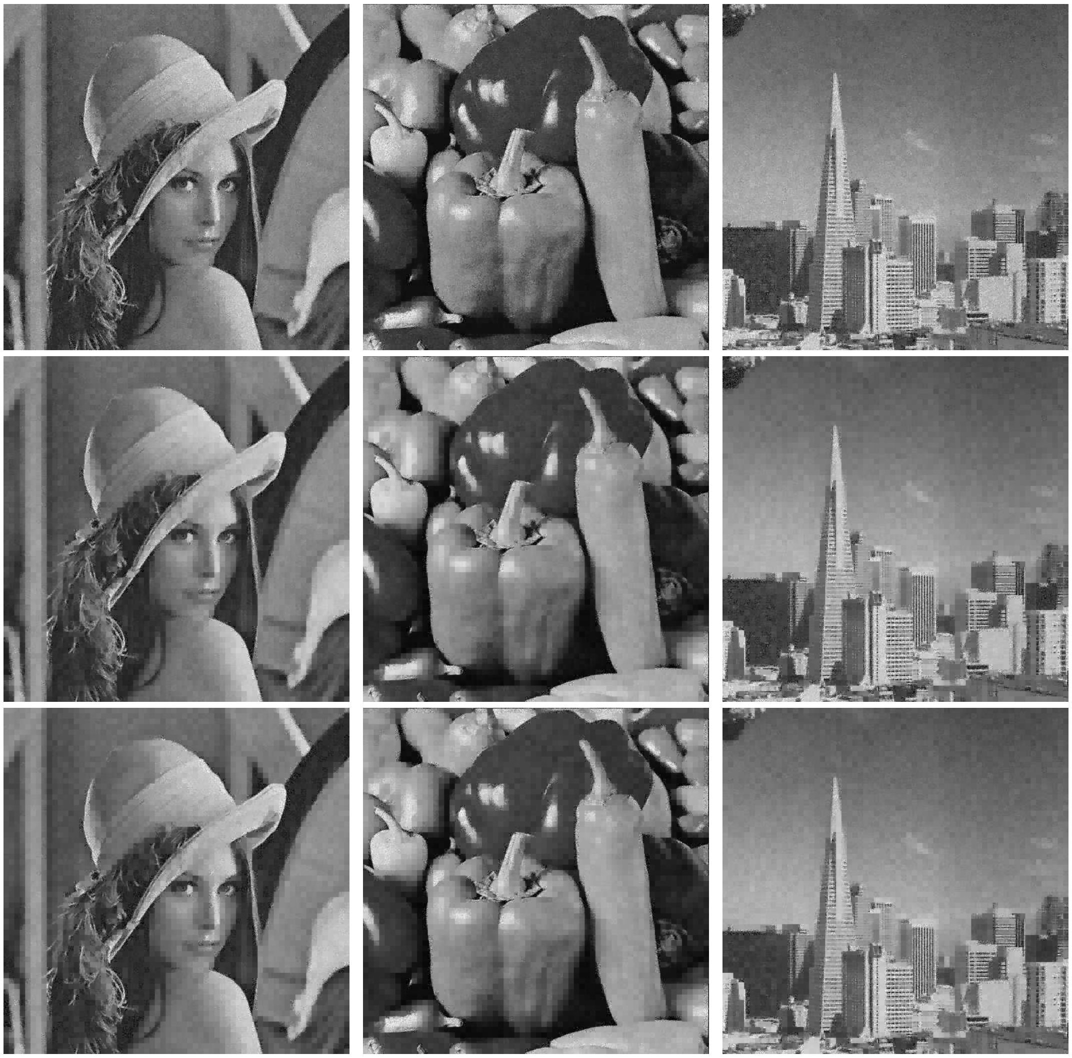


Figure 4.2.3: Denoised *Lena*, *Peppers*, and *San Francisco* images created using a block-based NL means denoising method. The first row is case 1, the second is case 2, and the third is case 3.

To briefly compare the block-based and pixel-based methods, look at Tables 4.2.2 and 4.2.3. From these we see that the results of the block-based method are inferior to the pixel-based results except for *Mandrill*, cases 2 and 3, and *Peppers*, case 2. Given the results for the rest of the images, *Mandrill* and *Peppers* seem to be a deviation from the norm. Indeed, this conclusion is not entirely surprising. Replacing entire blocks does not utilize pixel similarity windows as well as the pixel-based method and so second-rate results are generally expected.

4.3 Non-local means denoising in the wavelet domain

Consider a wavelet expansion of an image of resolution $2^K \times 2^K, K \geq 0$. It has already been shown that wavelet quadtrees can generally be well-approximated by (scaled) copies of other wavelet quadtrees at the same or higher levels. With this in mind, it is natural to consider an extension of the non-local means denoising algorithm to the wavelet domain.

Consider the standard tensor-product (real) wavelet basis expansion of a noisy image for which the wavelet coefficients can be arranged into a pyramid of blocks at each decomposition level. At level k of this decomposition, $0 \leq k \leq K$, the coefficients can be further partitioned into horizontal, vertical, and diagonal detail coefficients (denoted by A_k^h, A_k^v, and A_k^d respectively). Given a wavelet coefficient $n_{kij}^\lambda, \lambda \in \{h, v, d\}$, from a noisy image, let N_{kij}^λ denote the unique quadtree rooted at n_{kij}^λ. An estimate for a denoised quadtree is found by computing a weighted sum of all of the other quadtrees in the wavelet expansion of the noisy image. Futhermore, transformations of quadtrees of the form $\phi(t) = \alpha t$ are allowed. Letting D_{kij}^λ denote the estimate for the denoised quadtree, this gives

$$D_{kij}^\lambda = \sum_{\substack{0 \leq k' \leq k \\ 0 \leq i', j' \leq 2^{k'}-1}} w(N_{kij}^\lambda, N_{k'i'j'}^{\lambda'}) \alpha N_{k'i'j'}^{\lambda'}, \tag{4.3.1}$$

The weights are calculated as

$$w(N_{kij}^\lambda, N_{k'i'j'}^{\lambda'}) = \frac{1}{Z_i} \exp\left(-\frac{\|N_{kij}^\lambda - \alpha N_{k'i'j'}^{\lambda'}\|_2^2}{h^2}\right),$$

where

$$Z_i = \sum_{\substack{0 \leq k' \leq k \\ 0 \leq i', j' \leq 2^{k'}-1}} \exp\left(-\frac{\|N_{kij}^\lambda - \alpha N_{k'i'j'}^{\lambda'}\|_2^2}{h^2}\right).$$

The parameter α is chosen to minimize

$${}_{\lambda'}^{\lambda}\Delta_{k'i'j'}^{kij} = \|N_{kij}^\lambda - \alpha N_{k'i'j'}^{\lambda'}\|_2.$$

Of course, to compare quadtrees at different levels, the tree rooted at the higher level in the quadtree structure has to be truncated to ensure that the quadtrees being compared are of the same size.

For the following experiments, two special cases of the transformation $\phi(t) = \alpha t$ are considered. Numbering them so that the cases correspond somewhat to the work done previously, they are:

1. Purely translational: Wavelet quadtrees at the same scale are compared. Set $\alpha = 1$. Since entire quadtrees are being replaced, this is analogous to block-based NL means denoising.

3. Affine, same-scale: Wavelet quadtrees at the same scale are compared. Optimize over α.

Optimization over β is not allowed since at least theoretically, these quadtrees are of infinite length and should be ℓ^2-summable.

To perform the wavelet-based denoising, estimates were calculated for detail coefficient quadtrees rooted at level $k = 6$ (there are three decompositions total). For computational ease, only quadtrees rooted at the same level and with the same orientation were compared. That is, $k = k'$ and $\lambda = \lambda'$ in Eqn. (4.3.1). The Daubechies-6 wavelet is used. While a variety of values for h were tested, the following pictures display the results for $h = 0.3$. Different values for h led to many different RMSE and SSIM values but 0.3 seemed to be the value that generally led to high SSIM values for both cases.

Fig. 4.3.1 shows the pictorial results of these calculations for *Lena*, *Peppers*, *Zelda*, and *San Francisco*. Table 4.3.1 shows the numerical results for all eight test images. Visually there is very little variation between the two cases except that case 1 is sharper and more detailed. The case 1 images look most similar to the case 2 images from the block and pixel-based denoising. Examining the table of RMSE and SSIM values, there is no pattern to which case is better since the values change so much from image to image. When h was allowed to vary, substantial change in the RMSE and SSIM values was observed. For some values of h, case 1 was better while for others case 3 was best. Since the results seem to be largely image-dependent, we are not able to say at this time which approach is truly better when performing wavelet-based NL means denoising. With that being said, both cases do offer an improvement over the original noisy image in both SSIM and RMSE.

Case		*Lena*	*Mandrill*	*Goldhill*	*Boat*
1	RMSE SSIM	0.0402 0.4651	0.0621 0.6569	0.0432 0.5320	0.0451 0.4463
3	RMSE SSIM	0.0405 0.4713	0.0694 0.5502	0.0455 0.4695	0.0470 0.4262
		Barbara	*Peppers*	*San Fran.*	*Zelda*
1	RMSE SSIM	0.0477 0.5612	0.0405 0.4774	0.0454 0.3388	0.0332 0.4309
3	RMSE SSIM	0.0502 0.5467	0.0414 0.4831	0.0459 0.3492	0.0317 0.5882

Table 4.3.1: RMSE and SSIM values relative to their respective originals for eight images produced by wavelet-based NL means denoising.

We also include the α histogram distribution associated with case 3 for *Lena* and *Mandrill* in Fig. 4.3.2. These histograms are virtually identical and highly symmetric about zero.

One question that should be asked is "How (or should) the approximation coefficients be denoised?" One possibility is to save the approximation coefficients

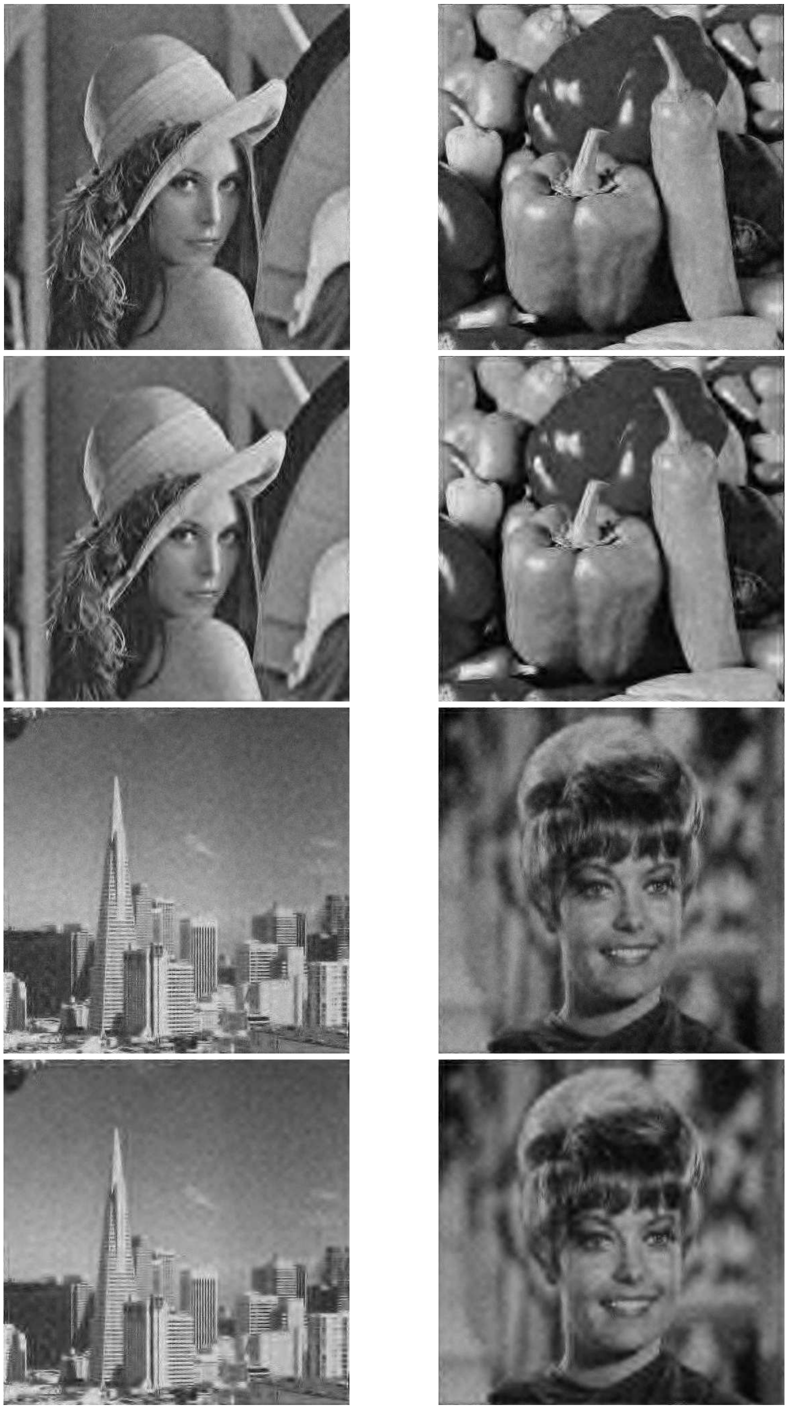


Figure 4.3.1: Denoised images created using the wavelet-based denoising method. Rows 1 and 2: Cases 1 and 3 for *Lena* and *Peppers*. Rows 3 and 4: Cases 1 and 3 for *San Francisco* and *Zelda*.

at the highest decomposition level. Since these coefficients can be displayed in block format, NL means denoising algorithm could be performed on this two-dimensional coefficient matrix. This idea was explored during the experimentation process and

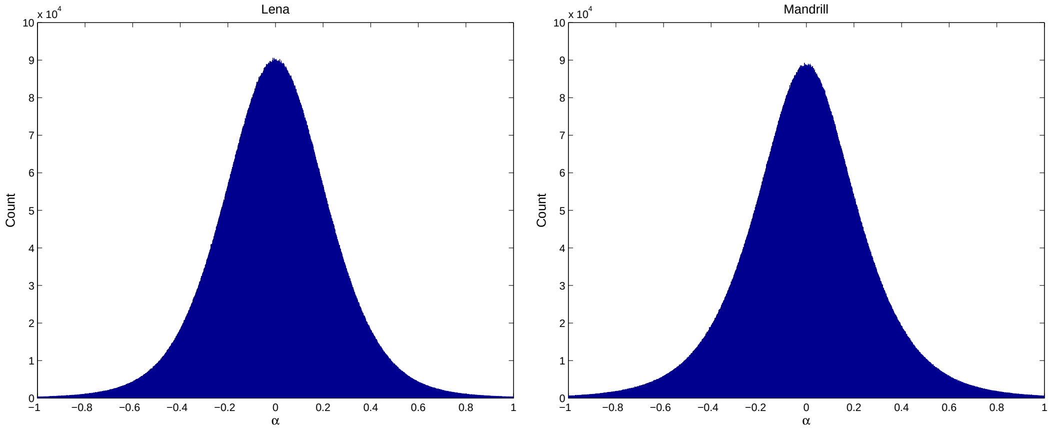


Figure 4.3.2: Histogram distributions of the parameter α in case 3 of the wavelet-based NL means denoising algorithm for noisy *Lena* (left) and *Mandrill* (right).

the results were virtually identical whether the approximation coefficients were modified or not. This indicates that most of the noise in the wavelet expansion is contained within the detail coefficients.

4.4 Summary

In this chapter a generalization of the well known non-local means denoising algorithm involving affine transformation of image subblocks was proposed and implemented in two ways: a pixel-based and a block-based method. The model was then adapted to the wavelet representation of images for the purpose of denoising wavelet quadtrees.

To compare the pixel and block-based methods, it suffices to say that the while the block-based method offers a great computational advantage (depending on what size of blocks are used of course), it can't keep up with the pixel-based method in terms of image quality due to the blockiness the denoised image takes on. Comparing the results from the wavelet-based method to the block- and pixel-based results, the quality of the images for the wavelet-based method can't compete. In the wavelet-based images there is also a lot of blurring and ringing around edges which makes it an unattractive choice.

Of large significance is whether any advantage is found by allowing for affine transformations of the form $\phi(t) = \alpha t + \beta$ in NL means denoising. For the pixel and block-based methods, there is no advantage to considering optimization over α because of the obvious degradation in image quality (most noticeably, blurring and spurious texturing). On the other hand, allowing for greyscale shifts improves the image quality according to the SSIM Index and seems to preserve more details. The computational cost is not much more here than in the strictly translational case so this could be an attractive choice for NL means denoising.

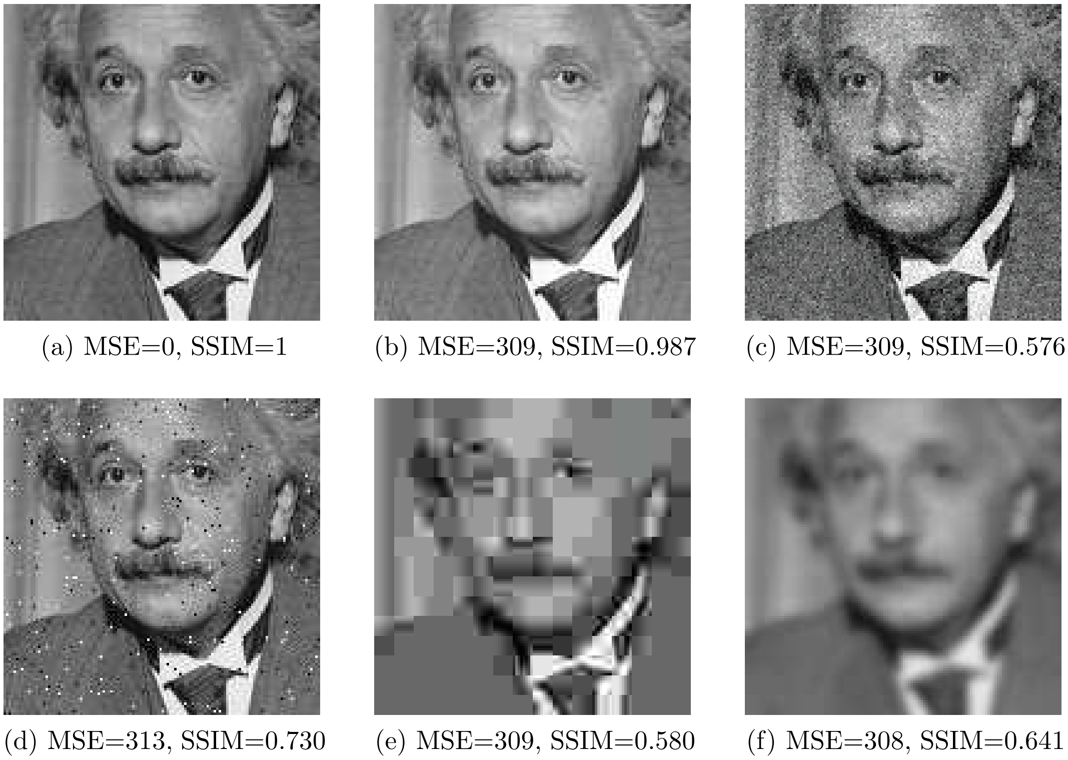
In the wavelet-based cases, such a wide range of results are observed that we are unable to say definitively at this time whether letting α vary is good or bad for denoising process. That being said, because of the extra blurring and loss of detail that results when α varies, allowing it to change is probably a poor choice.

Chapter 5

Optimizing the SSIM function on an L^2 ball

5.1 Introduction and the problem

Consider the following set of Einstein pictures, some of which were presented in Chapter 3.



(a) MSE=0, SSIM=1

(b) MSE=309, SSIM=0.987

(c) MSE=309, SSIM=0.576

(d) MSE=313, SSIM=0.730

(e) MSE=309, SSIM=0.580

(f) MSE=308, SSIM=0.641

Figure 5.1.1: Comparison of modified versions of an "Einstein" image which are approximately constrained to the L^2 ball of radius 310 centred on the original. (a) Original image. (b) Luminance shift. (c) Additive Gaussian noise. (d) Impulsive noise contamination. (e) JPEG compression. (f) Blurring.

The modified images all have MSE values quite close to 310 but they are very different perceptually. These perceptual differences are further illustrated by the wide range of SSIM values. Given a reference image I_0, it would be interesting to be able to mathematically describe images that are a prescribed L^2 distance away from I_0 as being the "best" or "worst" approximation to I_0 according to SSIM. In fact, this is a specific case of the following more general problem [21]: find the best and worst approximations to a reference image (according to some image similarity measure) while being constrained away from the reference image on a level set of some other distance function.

In [21], they propose using a gradient ascent/descent to approximate these critical points. The problem with this is that there is no guarantee of reaching an absolute maximum or minimum. In this chapter, we take a more mathematical approach and seek an analytic solution to the following problem:

Given a point $a = (a_1, \cdots, a_N) \in \mathbb{R}^N$, let $S_D(a)$ denote the L^2 ball of radius D centered at a. That is,

$$S_D(a) = \{x \in \mathbb{R}^N \mid \|x - a\|_2 = D\}.$$

Find and classify the critical points of the SSIM function,

$$S(x, a) = \frac{4\bar{x}\bar{a}s_{xa}}{(\bar{x}^2 + \bar{a}^2)(s_x^2 + s_a^2)}, \tag{5.1.1}$$

on $S_D(a)$.

5.2 The solution

The method of Lagrange multipliers will be used to tackle this problem but first a few definitions are necessary:

$$\begin{aligned} \bar{x} &= \frac{1}{N}\sum_{k=1}^{N} x_k, \\ s_x^2 &= \frac{1}{N-1}\sum_{k=1}^{N}(x_k - \bar{x})^2, \\ s_{xa} &= \frac{1}{N-1}\sum_{k=1}^{N}(x_k - \bar{x})(a_k - \bar{a}). \end{aligned} \tag{5.2.1}$$

Now consider the Lagrangian function

$$L(x) = S(x, a) + \lambda g(x),$$

where

$$g(x) = \sum_{k=1}^{N}(x_k - a_k)^2 - D^2,$$

and λ denotes a Lagrange multiplier. Impose the conditions that

$$\frac{\partial L}{\partial x_p} = 0, \quad 1 \le p \le N,$$

and

$$\frac{\partial L}{\partial \lambda} = 0. \tag{5.2.2}$$

The final condition specifies that $g(x) = 0$ or $x \in S_D(a)$. Computing partial derivatives, it is found that

$$\begin{aligned} \frac{\partial S}{\partial x_p} &= \frac{4\bar{a}}{N(\bar{x}^2+\bar{a}^2)^2(s_x^2+s_a^2)^2}\left[s_{xa}(s_x^2+s_a^2)(\bar{a}^2-\bar{x}^2)\right. \\ &+ \left.\frac{N}{N-1}\bar{x}(\bar{x}^2+\bar{a}^2)(s_x^2+s_a^2)(a_p-\bar{a}) - \frac{2N}{N-1}\bar{x}s_{xa}(\bar{x}^2+\bar{a}^2)(x_p-\bar{x})\right] \end{aligned} \tag{5.2.3}$$

Then the constraints $\frac{\partial L}{\partial x_p} = 0$ become

$$\begin{aligned} &\frac{4\bar{a}}{N(\bar{x}^2+\bar{a}^2)^2(s_x^2+s_a^2)^2}\left[s_{xa}(s_x^2+s_a^2)(\bar{a}^2-\bar{x}^2) + \frac{N}{N-1}\bar{x}(\bar{x}^2+\bar{a}^2)(s_x^2+s_a^2)(a_p-\bar{a})\right. \\ &\left. -\frac{2N}{N-1}\bar{x}s_{xa}(\bar{x}^2+\bar{a}^2)(x_p-\bar{x})\right] + 2\lambda(x_p-a_p) = 0, \quad 1 \le p \le N. \end{aligned} \tag{5.2.4}$$

Summing up both sides of (5.2.4) for $1 \le p \le N$, the following equality is obtained:

$$\frac{4\bar{a}}{(\bar{x}^2+\bar{a}^2)^2(s_x^2+s_a^2)^2}s_{xa}(s_x^2+s_a^2)(\bar{a}^2-\bar{x}^2) + 2N\lambda(\bar{x}-\bar{a}) = 0. \tag{5.2.5}$$

This is clearly satisfied if $\bar{x} = \bar{a}$. To fully solve the problem, it will also be necessary to consider the case $\bar{x} \ne \bar{a}$.

Case 1: $\bar{x} = \bar{a}$

In this case, the equations in (5.2.4) become, after some simplification and manipulation,

$$\begin{aligned} &\frac{1}{(N-1)(s_x^2+s_a^2)^2}\left[(s_x^2+s_a^2)(a_p-x_p) + (s_x^2+s_a^2-2s_{xa})(x_p-\bar{a})\right] \\ &+ \lambda(x_p-a_p) = 0, \qquad 1 \le p \le N. \end{aligned} \tag{5.2.6}$$

It is easy to show that

$$s_x^2 + s_a^2 - 2s_{xa} = \frac{D^2}{N-1}. \tag{5.2.7}$$

Substituting this into (5.2.6) yields the set of equations

$$\frac{1}{(N-1)(s_x^2+s_a^2)^2}\left[(s_x^2+s_a^2)(a_p-x_p) + \frac{D^2}{N-1}(x_p-\bar{a})\right] + \lambda(x_p-a_p) = 0, \quad 1 \le p \le N. \tag{5.2.8}$$

Rewrite these equations as follows:

$$\left[\frac{1}{(N-1)(s_x^2+s_a^2)}-\lambda\right](a_p-x_p)=-\frac{D^2}{(N-1)(s_x^2+s_a^2)^2}(x_p-\bar{a}), \qquad 1\leq p\leq N. \tag{5.2.9}$$

If $x_p=\bar{a}$ for any $p\in\{1,2,\cdots,N\}$, then there are two possibilities:

1. $x_p=a_p$, implying that $a_p=\bar{a}$. Note that the equations $x_p=a_p$ cannot be true for all $p\in\{1,2,\cdots,N\}$ since this would imply that $x=a$, violating the condition that $x\in S_D(a)$.

2. The Lagrange multiplier λ satisfies the equation,

$$\lambda=\frac{1}{(N-1)(s_x^2+s_a^2)}, \tag{5.2.10}$$

   at the extremum.

Case 2: If Eqn. (5.2.10) holds, then, from Eqn. (5.2.9), $x_p=\bar{a}$ for all $1\leq p\leq N$. But this implies that

$$\sum_{k=1}^{N}(x_k-a_k)^2=\sum_{k=1}^{N}(a_k-\bar{a})^2=(N-1)s_a^2, \tag{5.2.11}$$

which is not necessarily equal to D^2. In fact, s_a^2 and D can be chosen independently. Hence x does not necessarily lie on $S_D(a)$, violating the constraint.

Case 1: Rearrange the equations in (5.2.8) for those values of p such that $a_p\neq x_p$ and call this set of p-values $\mathcal{P}_1$:

$$\frac{(N-1)(s_x^2+s_a^2)^2}{D^2}\left[\frac{1}{(N-1)(s_x^2+s_a^2)}-\lambda\right]=\frac{x_p-\bar{a}}{x_p-a_p}, \qquad p\in\mathcal{P}_1. \tag{5.2.12}$$

For each pair (a,D), the LHS of (5.2.12) is a constant at each extremum, independent of p. Denote this constant as

$$\beta=\beta(a,D)=\frac{x_p-\bar{a}}{x_p-a_p}, \qquad p\in\mathcal{P}_1. \tag{5.2.13}$$

It is now useful to examine the consequences of (5.2.13), noting that

$$x_p-\bar{a}=\beta(x_p-a_p), \qquad 1\leq p\leq N, \tag{5.2.14}$$

since for $p\notin\mathcal{P}_1$, both sides of the equation are zero. First,

$$\sum_{k=1}^{N}(x_k-\bar{a})^2 \;=\; \beta^2D^2. \tag{5.2.15}$$

Clearly then, since $\bar{x} = \bar{a}$,

$$s_x^2 = \frac{1}{N-1}\beta^2 D^2. \tag{5.2.16}$$

Now assume $\beta \neq 0$. This is reasonable, since not all of the x_p are equal to $\bar{a}$. Then we can write

$$2s_{xa} = \frac{1}{N-1}\beta^2 D^2 - \frac{1}{N-1}D^2 + s_a^2. \tag{5.2.17}$$

Also assume that $\beta \neq 1$. If $\beta = 1$ then from (5.2.14), $a_p = \bar{a}$ and so $s_a^2 = 0$. By substituting Eqns. (5.2.17) and (5.2.16) into (5.1.1), it is immediate that $S(x,a) = 0$ and the problem is trivial. So, under the assumption $\beta \neq 1$, from the definition of s_{xa} and Eqn. (5.2.7), it can be shown that

$$s_{xa} = \frac{\beta}{\beta - 1}s_a^2. \tag{5.2.18}$$

Returning to the structural similarity function and employing the results obtained so far, including $\bar{x} = \bar{a}$, it is found that:

$$S(x,a) = \frac{\beta s_a^2}{\beta s_a^2 + \frac{\beta-1}{2}\frac{D^2}{N-1}}.$$

Now combine Eqs. (5.2.17) and (5.2.18) to get

$$\frac{2\beta}{\beta-1}s_a^2 = \frac{D^2}{N-1}\beta^2 - \frac{D^2}{N-1} + s_a^2. \tag{5.2.19}$$

Multiplying both sides by $\beta - 1$ and rearrangement produces the following cubic equation in the unknown β:

$$\frac{D^2}{N-1}\beta^3 - \frac{D^2}{N-1}\beta^2 - \left(\frac{D^2}{N-1} + s_a^2\right)\beta + \left(\frac{D^2}{N-1} - s_a^2\right) = 0. \tag{5.2.20}$$

One root of this cubic equation is $\beta = -1$. Dividing by $\beta + 1$, the other two roots are found to be

$$\beta = 1 \pm \sqrt{\frac{(N-1)s_a^2}{D^2}}.$$

It remains to find the values of $S(x,a)$ that correspond to each value of β. Substituting $\beta = -1$ into Eqn. (5.2.14) and rearranging yields the point

$$\mathbf{x} = \frac{1}{2}(\mathbf{a} - \bar{a}\mathbf{1}),$$

which does not generally lie on the ball (here, $\mathbf{1}$ denotes the N-vector $(1,1,\cdots,1)$). Thus, $\beta = -1$ is rejected.

1. $\beta = 1 + \sqrt{\frac{(N-1)s_a^2}{D^2}}$. In this case, the structural similarity is given by

$$S_1(x,a) = \frac{s_a^2 + s_a\frac{D}{\sqrt{N-1}}}{s_a^2 + s_a\frac{D}{\sqrt{N-1}} + \frac{D^2}{2(N-1)}}.$$

2. $\beta = 1 - \sqrt{\frac{(N-1)s_a^2}{D^2}}$. In this case, the structural similarity is given by

$$S_2(x, a) = \frac{s_a^2 - s_a \frac{D}{\sqrt{N-1}}}{s_a^2 - s_a \frac{D}{\sqrt{N-1}} + \frac{D^2}{2(N-1)}}.$$

To compare the values of S_1 and S_2, it is convenient to express them as follows:

$$S_{1,2} = \frac{a \pm c}{a \pm c + b},$$

where

$$a = s_a^2, \qquad b = \frac{D^2}{2(N-1)}, \qquad c = s_a \frac{D}{\sqrt{N-1}}.$$

Since $a, b, c > 0$, it follows from simple algebra that $S_1 > S_2$.

Since the values of β corresponding to the extrema have been identified, the points $x \in \mathbb{R}^N$ at which each of the extrema occur can be computed from Eqn. (5.2.13). For the β value corresponding to each case, solve for x_p:

$$x_p = \frac{1}{\beta - 1}(\beta a_p - \bar{a}), \qquad 1 \leq p \leq N. \tag{5.2.21}$$

Summing over all p and dividing by N yields

$$\bar{x} = \frac{\beta - 1}{\beta - 1}\bar{a} = \bar{a},$$

confirming that the condition $\bar{x} = \bar{a}$ is satisfied. Then, substituting for β and rewriting Eqn. (5.2.21) gives

$$x_p = a_p \pm \frac{D}{s_a\sqrt{N-1}}(a_p - \bar{a}). \tag{5.2.22}$$

In vector format, Eqn. (5.2.22) may be written as

$$\mathbf{x} = \mathbf{a} \pm \frac{D}{s_a\sqrt{N-1}}\,\mathbf{a}',$$

where $\mathbf{a}' = \mathbf{a} - \bar{a}\mathbf{1}$ denotes the zero-mean component of $\mathbf{a}$. In fact, a closer inspection shows that

$$\mathbf{x} = \mathbf{a} \pm D\hat{\mathbf{a}}',$$

where $\hat{\mathbf{a}}'$ is the unit vector in the direction of the zero-mean component $\mathbf{a}'$. Thus, $x \in S_D(a)$.

Case 2: $\bar{x} \neq \bar{a}$

Returning to Eqn. (5.2.5), if $\bar{x} \neq \bar{a}$, then the factor $\bar{x} - \bar{a}$ may be divided out to obtain the result

$$\lambda = \frac{2\bar{a}s_{xa}(\bar{x}+\bar{a})}{N(\bar{x}^2+\bar{a}^2)^2(s_x^2+s_a^2)}.$$

Substituting this result into Eqn. (5.2.4) and rearranging yields the equation

$$(x_p - \bar{x})\sigma_{xa}\left[(\sigma_x^2+\sigma_a^2)(\bar{a}+\bar{x}) - \frac{2N}{N-1}\bar{x}(\bar{x}^2+\bar{a}^2)\right]$$
$$+(a_p-\bar{a})(\sigma_x^2+\sigma_a^2)\left[\frac{N}{N-1}\bar{x}(\bar{x}^2+\bar{a}^2) - \sigma_{xa}(\bar{x}+\bar{a})\right] = 0. \qquad (5.2.23)$$

As in Case 1, rearrange the equations in (5.2.23) for those values of p such that $a_p \neq \bar{a}$ and call this set of p-values $\mathcal{P}_2$. After this rearrangement, an analysis similar to that of Case 1 shows that the following ratio is constant at each extremum:

$$\alpha = \alpha(a, D) = \frac{x_p - \bar{x}}{a_p - \bar{a}}, \qquad p \in \mathcal{P}_2.$$

The consequences of this relation will again be examined, noting that

$$x_p - \bar{x} = \alpha(a_p - \bar{a}), \qquad 1 \leq p \leq N, \qquad (5.2.24)$$

since for $p \notin \mathcal{P}_2$, both sides of the equation are zero. First, squaring both sides and summing over $1 \leq p \leq N$ yields

$$s_x^2 = \alpha^2 s_a^2. \qquad (5.2.25)$$

Further, directly from the definition of s_{xa} it is found that,

$$s_{xa} = \alpha s_a^2. \qquad (5.2.26)$$

It now remains to determine acceptable values for α. To this end, return to Eqn. (5.2.3) which gives the components of the gradient vector $\vec{\nabla}S$. At a stationary point $\mathbf{x}$, it is necessary that $\vec{\nabla}S(\mathbf{x})$ be a constant multiple of the outward normal vector $\hat{\mathbf{n}}$ to the sphere S_D at $\mathbf{x}$, given by

$$\hat{\mathbf{n}} = \frac{1}{D}(x_1 - a_1, x_2 - a_2, \cdots, x_N - a_N) = \frac{1}{D}(\mathbf{x} - \mathbf{a}). \qquad (5.2.27)$$

This is the essence of Eqn. (5.2.4), which is a result of the Lagrangian optimization method.

To solve for the acceptable α values, substitute Eqns. (5.2.24), (5.2.25) and (5.2.26) into Eqn. (5.2.3):

$$\frac{\partial S}{\partial x_p} = \frac{4\bar{a}}{N(\bar{x}^2+\bar{a}^2)^2(s_x^2+s_a^2)^2}\left[\alpha s_a^4(1+\alpha^2)(\bar{a}^2-\bar{x}^2)\right.$$
$$\left. + \frac{N}{N-1}\bar{x}(\bar{x}^2+\bar{a}^2)(1-\alpha^2)(a_p-\bar{a})\right]. \qquad (5.2.28)$$

In general, the only way that the gradient vector $\vec{\nabla} S$ can be a multiple of the normal vector $\hat{\mathbf{n}}$ is when the final term in the above equation vanishes, i.e., when $\alpha = \pm 1$. Then

$$\frac{\partial S}{\partial x_p} = \frac{4\bar{a}}{N(\bar{x}^2 + \bar{a}^2)^2 (s_x^2 + s_a^2)^2} \left[\alpha s_a^4 (1 + \alpha^2)(\bar{a}^2 - \bar{x}^2) \right], \qquad 1 \leq p \leq N. \tag{5.2.29}$$

Now both of the possible values for α are considered separately.

1. **Case 1:** $\alpha = 1$. In this case, Eqn. (5.2.24) can be written as

$$x_p - a_p = \bar{x} - \bar{a}. \tag{5.2.30}$$

Furthermore, since $\mathbf{x}$ must lie on the sphere $S_D(a)$ squaring both sides of (5.2.30) and summing over $1 \leq p \leq N$ yields

$$\sum_{p=1}^{N} (x_p - a_p)^2 = D^2 = N(\bar{x} - \bar{a})^2.$$

This implies that

$$\bar{x} = \bar{a} \pm \frac{D}{\sqrt{N}}. \tag{5.2.31}$$

Substituting this result into Eqn. (5.2.30) gives

$$x_p = a_p \pm \frac{D}{\sqrt{N}}.$$

This implies that

$$\mathbf{x} = \mathbf{a} \pm \frac{D}{\sqrt{N}} (1, 1, \cdots, 1) = \mathbf{a} \pm D\hat{\mathbf{1}},$$

where $\hat{\mathbf{1}}$ is the unit vector in the direction $\mathbf{1} = (1, 1, \cdots, 1)$. In other words, the vector $\mathbf{x} - \mathbf{a}$ is perpendicular to the plane $\bar{x} = \bar{a}$.

From Eqn. (5.2.31), there are two more subcases:

(a) **Case 1(a):** The mean of $\mathbf{x}$ is given by

$$\bar{x} = \bar{a} + \frac{D}{\sqrt{N}}.$$

Evaluating $S(x, a)$ at this point gives

$$S_{1a}(x, a) = \frac{2\bar{a}\left(\bar{a} + \frac{D}{\sqrt{N}}\right)}{2\bar{a}\left(\bar{a} + \frac{D}{\sqrt{N}}\right) + \frac{D^2}{N}}.$$

(b) **Case 1(b):** The mean of $\mathbf{x}$ is given by

$$\bar{x} = \bar{a} - \frac{D}{\sqrt{N}}.$$

Once again, evaluating $S(x, a)$ at this point gives:

$$S_{1b}(x, a) = \frac{2\bar{a}\left(\bar{a} - \frac{D}{\sqrt{N}}\right)}{2\bar{a}\left(\bar{a} - \frac{D}{\sqrt{N}}\right) + \frac{D^2}{N}}.$$

2. **Case 2:** $\alpha = -1$. In this case, Eqn. (5.2.24) becomes

$$x_p - \bar{x} = \bar{a} - a_p. \tag{5.2.32}$$

Rewrite the above equation as

$$x_p - a_p = (\bar{x} - \bar{a}) + 2(\bar{a} - a_p).$$

Now square both sides and sum over the index $1 \leq p \leq N$:

$$\begin{aligned} D^2 &= \sum_{p=1}^{N} (x_p - a_p)^2 \\ &= N(\bar{x} - \bar{a})^2 + 4\sum_{p=1}^{N}(a_p - \bar{a})^2 + 2(\bar{x} - \bar{a})\sum_{p=1}^{N}(\bar{a} - a_p) \\ &= N(\bar{x} - \bar{a})^2 + 4(N-1)s_a^2. \end{aligned}$$

A slight rearrangement yields

$$\bar{x} = \bar{a} \pm \frac{1}{\sqrt{N}}\sqrt{D^2 - 4(N-1)s_a^2} \tag{5.2.33}$$

This result is feasible provided D and N are chosen so that

$$\Delta = D^2 - 4(N-1)s_a^2 \geq 0.$$

If this is true then by substituting (5.2.33) into (5.2.32) it is found that the two critical points are given by

$$\begin{aligned} \mathbf{x}_1 &= 2\bar{a}\mathbf{1} - \mathbf{a} + \sqrt{\Delta}\hat{\mathbf{1}}, \quad \text{and} \\ \mathbf{x}_2 &= 2\bar{a}\mathbf{1} - \mathbf{a} - \sqrt{\Delta}\hat{\mathbf{1}}, \end{aligned}$$

with corresponding values of $S(x, a)$ given by

$$\begin{aligned} S_{-1a} &= \frac{2\bar{a}\left(\bar{a} + \sqrt{\frac{\Delta}{N}}\right)}{2\bar{a}\left(\bar{a} + \sqrt{\frac{\Delta}{N}}\right) + \frac{\Delta}{N}}, \quad \text{and} \\ S_{-1b} &= \frac{2\bar{a}\left(\bar{a} - \sqrt{\frac{\Delta}{N}}\right)}{2\bar{a}\left(\bar{a} - \sqrt{\frac{\Delta}{N}}\right) + \frac{\Delta}{N}}. \end{aligned}$$

The quantities S_{1a}, S_{1b}, S_{-1a}, and S_{-1b} are of the same form that was encountered in the equal means case. The same analysis shows that

$$\begin{aligned} S_{1a} &> S_{1b} \text{ and } S_{-1a} > S_{-1b} \quad \text{if} \quad \bar{a} > 0, \\ S_{1a} &< S_{1b} \text{ and } S_{-1a} < S_{-1b} \quad \text{if} \quad \bar{a} < 0. \end{aligned}$$

5.3 Conclusion and some results

Let us summarize our findings below, introducing appropriate notation to differentiate between the two cases. The following critical points and corresponding values of $S(x, a)$ were identified:

1. **Case 1:** $\bar{x} = \bar{a}$

$$\begin{aligned} S_\beta^{(1)} &= \frac{s_a^2 + s_a \frac{D}{\sqrt{N-1}}}{s_a^2 + s_a \frac{D}{\sqrt{N-1}} + \frac{D^2}{2(N-1)}} \quad \text{at} \quad \mathbf{x} = \mathbf{a} + D\hat{\mathbf{a}}' \\ S_\beta^{(2)} &= \frac{s_a^2 - s_a \frac{D}{\sqrt{N-1}}}{s_a^2 - s_a \frac{D}{\sqrt{N-1}} + \frac{D^2}{2(N-1)}} \quad \text{at} \quad \mathbf{x} = \mathbf{a} - D\hat{\mathbf{a}}' \end{aligned}$$

2. **Case 2:** $\bar{x} \neq \bar{a}$

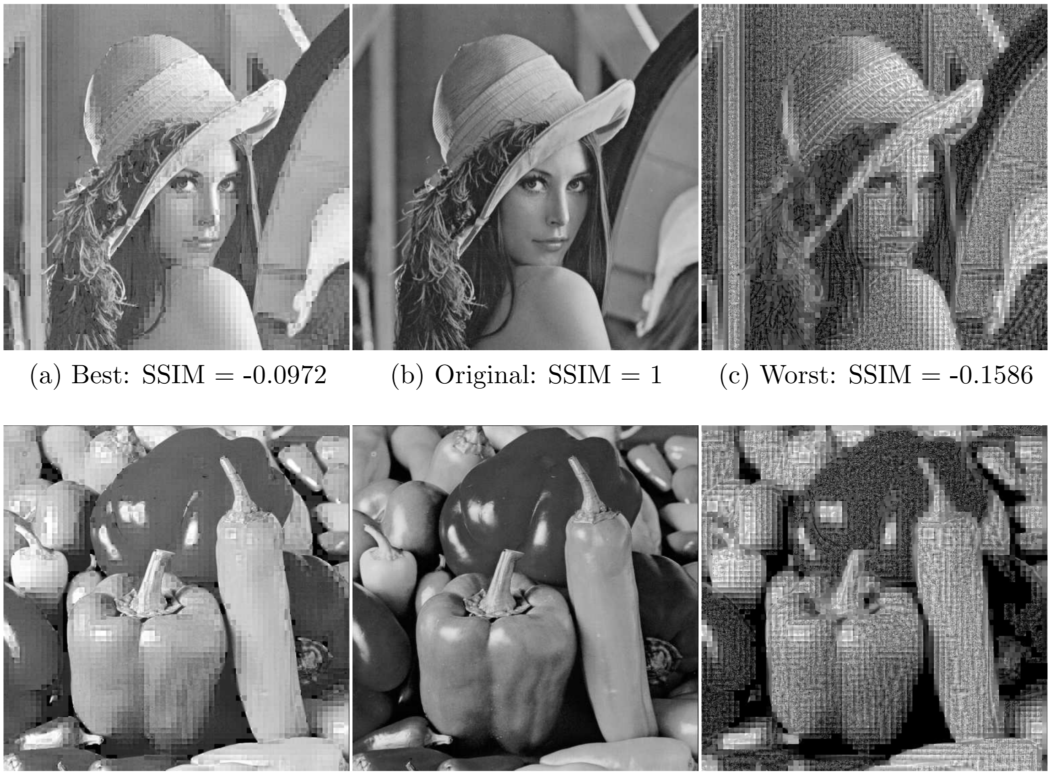
$$\begin{aligned} S_\alpha^{(1)} &= \frac{\bar{a}\left(\bar{a} + \frac{D}{\sqrt{N}}\right)}{\bar{a}\left(\bar{a} + \frac{D}{\sqrt{N}}\right) + \frac{D^2}{2N}} \quad \text{at} \quad \mathbf{x} = \mathbf{a} + D\hat{\mathbf{1}} \\ S_\alpha^{(2)} &= \frac{\bar{a}\left(\bar{a} - \frac{D}{\sqrt{N}}\right)}{\bar{a}\left(\bar{a} - \frac{D}{\sqrt{N}}\right) + \frac{D^2}{2N}} \quad \text{at} \quad \mathbf{x} = \mathbf{a} - D\hat{\mathbf{1}}. \\ S_\alpha^{(3)} &= \frac{\bar{a}\left(\bar{a} + \sqrt{\frac{\Delta}{N}}\right)}{\bar{a}\left(\bar{a} + \sqrt{\frac{\Delta}{N}}\right) + \frac{\Delta}{2N}} \quad \text{at} \quad \mathbf{x} = 2\bar{a}\mathbf{1} - \mathbf{a} + \sqrt{\Delta}\hat{\mathbf{1}}, \\ S_\alpha^{(4)} &= \frac{\bar{a}\left(\bar{a} - \sqrt{\frac{\Delta}{N}}\right)}{\bar{a}\left(\bar{a} - \sqrt{\frac{\Delta}{N}}\right) + \frac{\Delta}{2N}} \quad \text{at} \quad \mathbf{x} = 2\bar{a}\mathbf{1} - \mathbf{a} - \sqrt{\Delta}\hat{\mathbf{1}}. \end{aligned}$$

The last two critical points exist provided $\Delta = D^2 - 4(N-1)s_a^2 \geq 0$.

These formulas have been verified numerically. It would be desirable to be able to derive a condition that guarantees whether the global extrema for $S(x, a)$ occur on or off the plane $\bar{x} = \bar{a}$. Indeed, numerical results show that global maxima and minima may be obtained both on and off the plane. Unfortunately, actually classifying and comparing these critical points seems quite complicated if it is indeed possible. However, it has been shown that $S_\beta^{(1)} > S_\beta^{(2)}$. It is also true that

$$\begin{aligned} S_\alpha^{(1)} &> S_\alpha^{(2)} \text{ and } S_\alpha^{(3)} > S_\alpha^{(4)} \quad \text{if} \quad \bar{a} > 0, \quad \text{and} \\ S_\alpha^{(1)} &< S_\alpha^{(2)} \text{ and } S_\alpha^{(3)} < S_\alpha^{(4)} \quad \text{if} \quad \bar{a} < 0. \end{aligned}$$

The above results will now be illustrated with a couple of examples using *Lena* and *Peppers*. Instead of using the entire image, a local approach is taken. Let R_i denote an image subblock. For each R_i, we seek the best approximation to R_i according to SSIM while being constrained on an L^2 ball about R_i. In the following calculations, 8×8 non-overlapping image subblocks are used and the radius of the L^2 ball is $D = 300$. Fig. 5.3.1 shows the results of these experiments.



(a) Best: SSIM = -0.0972 (b) Original: SSIM = 1 (c) Worst: SSIM = -0.1586

(d) Best: SSIM = -0.0435 (e) Original: SSIM = 1 (f) Worst: SSIM = -0.0701

Figure 5.3.1: Best and worst approximations to *Lena* and *Peppers* while constrained on an L^2 ball of radius 300.

The calculations that produced these images showed that the best approximations on the L^2 ball almost always occurs off the plane $\bar{x} = \bar{a}$, while the worst approximations are usually obtained on it. In fact, of all the eight test images used, *Peppers* was the only image where a best approximation occurred on the plane $\bar{x} = \bar{a}$. However, even for *Peppers*, this was the case only the case a handful of times. Visually, the best approximations on the L^2 ball often appear to be a greyscale shift of the original image (these correspond to the SSIM value $S_\alpha^{(1)}$). The blocky appearance of the images is a result of maximum and minimum values being obtained for values of $\mathbf{x}$ not corresponding to $S_\alpha^{(1)}$ or $S_\alpha^{(2)}$. The reason for this blocky appearance then is that $S_\alpha^{(1)}$ and $S_\alpha^{(2)}$ result from a uniform greyscale shift of the image subblocks, and all the other SSIM values do not.

Bibliography

[1] S.K. Alexander. *Multiscale Methods in Image Modelling and Image Processing.* PhD thesis, University of Waterloo, Waterloo, ON, Canada, 2005.

[2] S.K. Alexander, E.R. Vrscay, and S. Tsurumi. A simple, general model for the affine self-similarity of images. In *5th International Conference on Image Analysis and Recognition ICIAR '08*, pages 192–203, Póvoa de Varzim, Portugal, 2008. Springer-Verlag.

[3] Z. Barahav, D. Malah, and E. Karnin. *Fractal Image Compression: Theory and Application.* Springer-Verlag, 1995.

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

[5] D. Brunet, E.R. Vrscay, and Z. Wang. A class of image metrics based on the structural similarity quality index. In *8th International Conference on Image Analysis and Recognition ICIAR '11*, Burnaby, BC, Canada, June 2011.

[6] D. Brunet, E.R. Vrscay, and Z. Wang. Structural similarity-based affine approximation and self-similarity revisted. In *8th International Conference on Image Analysis and Recognition ICIAR '11*, Burnaby, BC, Canada, June 2011.

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

[8] I. Daubechies. *Ten Lectures on Wavelets.* CBMS-NSF Reg. Conf. Series in Applied Math. SIAM, 1992.

[9] G.M. Davis. A wavelet-based analysis of fractal image compression. *IEEE Transactions on Image Processing*, 7(2):141–154, 1998.

[10] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision*, pages 1033–1038, 1999.

[11] B. Forte and E.R. Vrscay. Theory of generalized fractal transforms. In Y. Fisher, editor, *Conf. Proc. NATO ASI Fractal Image Encoding and Analysis*, Trondheim, Norway, July 1995. Springer-Verlag.

[12] B. Forte and E.R. Vrscay. Inverse problem methods for generalized fractal transforms. In Y. Fisher, editor, *Fractal Image Encoding and Analysis.* Springer Verlag, 1998.

[13] E. Kreyszig. *Introductory Functional Analysis with Application.* Wiley, 1978.

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

[15] S. Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way.* Academic Press, 3rd edition, 2008.

[16] E.R. Vrscay. A generalized class of fractal-wavelet transforms for image representation and compression. *Canadian Journal of Electrical and Computer Engineering*, 23(2):69–83, 1998.

[17] E.R. Vrscay. A simple model for affine self-similarity of images and its applications. In *2008 International Workshop on Local and Nonlocal Approximation in Image Processing LNLA '08*, pages 23–24, Lausanne, Switzerland, August 2008.

[18] Z. Wang and A.C. Bovik. A universal image quality index. *Signal Processing*, 9(3):81–84, 2002.

[19] Z. Wang and A.C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, January 2009.

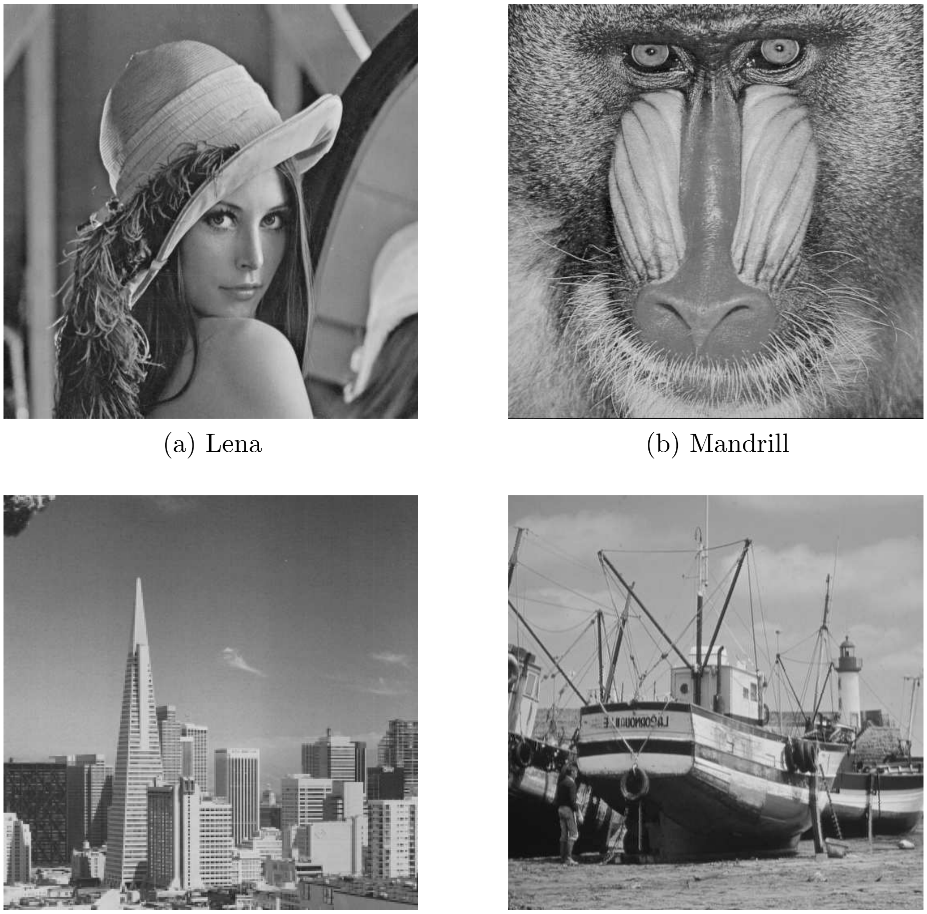
[20] Z. Wang and E. Simoncelli. Translation insensitive image similarity in complex wavelet domain. In *IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP '05*, pages 573–576, 2005.

[21] Z. Wang and E. Simoncelli. Maximum differentiation (mad) competition: A methodology for comparing computational models of perceptual quantities. *J. Vis.*, 8(12):1–13, 9 2008.

Appendix A

Test images

Several 8-bit greyscale images are used repeatedly in experiments throughout this thesis and they are included here for reference.



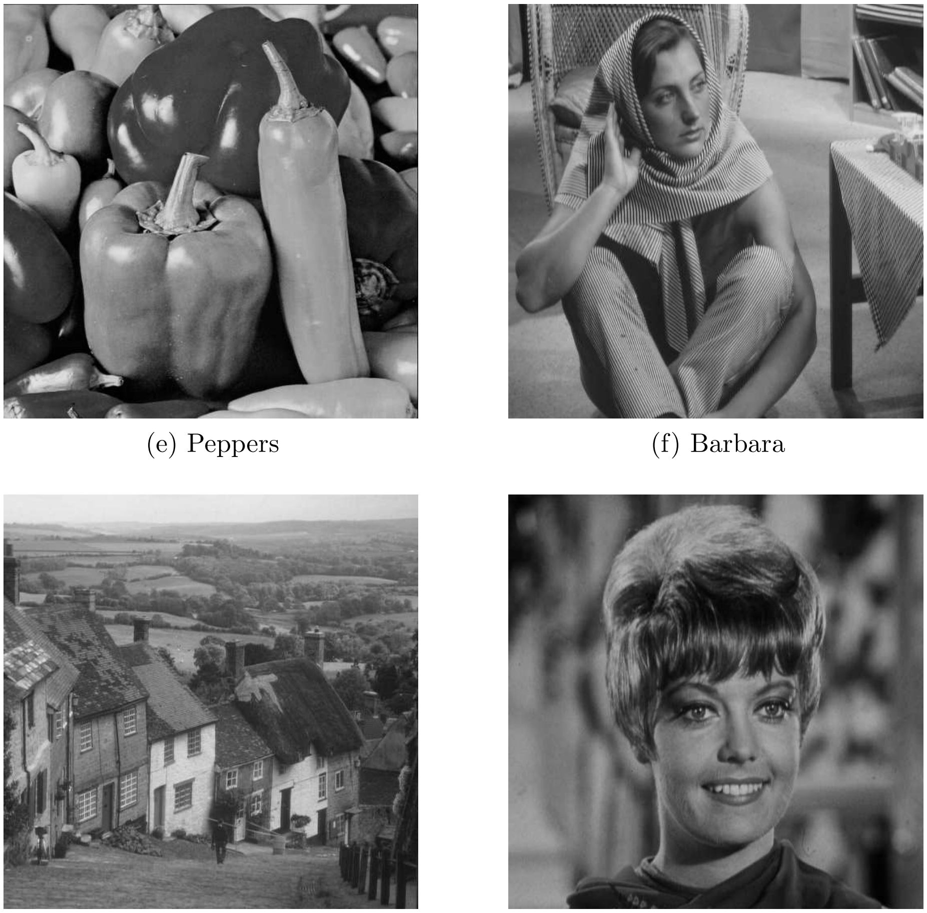
(a) Lena

(b) Mandrill

(c) San Francisco

(d) Boat

Figure A.0.1: Test images.



(e) Peppers

(f) Barbara

(g) Goldhill

(h) Zelda

Figure A.0.1: Test images.

Appendix B

More on the SSIM function

B.1 Notation

Let $x, y \in \mathbb{R}^N$. Throughout this appendix the SSIM function with zero stability constants will be used:

$$S(x,y) = S_1(x,y)S_2(x,y) = \frac{2\bar{x}\bar{y}}{\bar{x}^2+\bar{y}^2} \cdot \frac{2s_{xy}}{s_x^2+s_y^2}, \tag{B.1.1}$$

where

$$\begin{aligned} \bar{x} &= \frac{1}{N}\sum_{k=1}^{N} x_k, \\ s_{xy} &= \frac{1}{N-1}\sum_{k=1}^{N}(x_k-\bar{x})(y_k-\bar{y}), \quad \text{and} \\ s_x^2 &= \frac{1}{N-1}\sum_{k=1}^{N}(x_k-\bar{x})^2. \end{aligned}$$

B.2 The SSIM Index in terms of complex wavelets

An extension of the SSIM Index to complex-valued wavelets is given in [20]. There is, however, very little mathematical justification given for the formula. This section shows that much like in the case of real-valued wavelets considered earlier in the chapter on the SSIM Index, their formula can be justified by considering projections of vectors onto complex-valued wavelet basis functions.

Before this can be done, a few modified definitions are required. Let $x, y \in \mathbb{R}^M$ where $M = 2^K$ for some $K \geq 0$. The mean of the signal, $\bar{x}$, is still defined the same way. However, the definitions for s_{xy} and s_x^2 are modified:

$$\begin{aligned} s_{xy} &= \frac{1}{N-1}\sum_{k=1}^{M}(x_k - \bar{x})(y_k - \bar{y})^*, \quad \text{and} \\ s_x^2 &= \frac{1}{N-1}\sum_{k=1}^{M}|x_k - \bar{x}|^2. \end{aligned}$$

Here, z^* denotes complex conjugation and $|z|$ denotes the modulus. Note that s_{xy} is no longer symmetric and as result, the SSIM Index is no longer symmetric with respect to its arguments.

Now consider the projections of x and y onto a set of orthonormal zero-mean (complex) wavelet basis functions $\{\Psi_k \mid 1 \leq k \leq N\}$ which corresponds to a quadtree with N nodes in the wavelet decompositions of x and y. For example, if the quadtree is rooted at the very top of the pyramid of coefficients, $N = 2^K - 1$. This set of functions *does not* itself constitute a basis but is a subset of a basis for $\mathbb{C}^M$. The best L^2-based approximations of x and y in terms of this basis are written as

$$\begin{aligned} Px &= \sum_{k=1}^{N} a_k \Psi_k \quad \text{and} \\ Py &= \sum_{k=1}^{N} c_k \Psi_k, \end{aligned}$$

where $a_k = \langle x, \Psi_k \rangle$ and $c_k = \langle y, \Psi_k \rangle$. It now remains to define $\overline{Px}$, s_{PxPy}, and s_{Px}^2 in terms of these expansion coefficients. First,

$$\overline{Px} = \sum_{k=1}^{N} a_k \overline{\Psi}_k = 0$$

since the wavelet basis functions are zero-mean. Next,

$$\begin{aligned} s_{PxPy} &= \frac{1}{N-1}\sum_{k=1}^{N}(Px_k - \overline{Px})(Py_k - \overline{Py})^* \\ &= \frac{1}{N-1}\langle Px, Py \rangle \\ &= \frac{1}{N-1}\sum_{k=1}^{N} a_k c_k^* \end{aligned}$$

The last step follows because the basis functions are orthonormal. To get s_{Px}^2, let $y = x$ in the formula for s_{PxPy} to get

$$s_{Px}^2 = \frac{1}{N-1}\sum_{k=1}^{N} |a_k|^2.$$

Let us now make these substitutions in the SSIM function remembering that since $\overline{Px} = \overline{Py} = 0$, $S_1(x, y) = 1$ and so $S(x, y) = S_2(x, y)$. Adding in the stability constants, the SSIM function can be expressed in terms of the projection coefficients of x and y as

$$S_{W,\mathbb{C}}(x, y) = \frac{\frac{2}{N-1}\sum_{k=1}^{N} a_k c_k^* + C}{\frac{1}{N-1}\sum_{k=1}^{N}[|a_k|^2 + |c_k|^2] + C}. \tag{B.2.1}$$

There are two glaring issues with $S_{W,\mathbb{C}}(x, y)$. First, because there is no partial ordering on $\mathbb{C}^N$, SSIM values cannot be directly compared. Furthermore, $S_{W,\mathbb{C}}(x, y)$ is not symmetric with respect to its arguments.

Before giving the complex wavelet SSIM (CW-SSIM) formula from [20], we emphasize that it is subject to a *greatly* different interpretation than $S_{W,\mathbb{C}}(x, y)$. For this formula, let $\mathbf{a} = \{a_k \mid 1 \leq k \leq N\}$ and $\mathbf{c} = \{c_k \mid 1 \leq k \leq N\}$ denote two sets of coefficients extracted from the same spatial location in the same level of the wavelet coefficient quadtree structure of two vectors x and y, respectively. The second CW-SSIM function is

$$\tilde{S}_{W,\mathbb{C}}(x, y) = \frac{2|\sum_{k=1}^{N} a_k c_k^*| + C}{\sum_{k=1}^{N}[|a_k|^2 + |c_k|^2] + C}. \tag{B.2.2}$$

The formula $\tilde{S}_{W,\mathbb{C}}(x, y)$ can be seen to be obtained from $S_{W,\mathbb{C}}(x, y)$ by taking the modulus of the sum in the numerator. Also, it can be thought of as a kind of special case of $S_{W,\mathbb{C}}(x, y)$ where the basis functions used only correspond to one level of the wavelet coefficient pyramid. Computationally, it is implemented by moving a sliding window across the matrices of detail coefficients at each level in the wavelet decomposition pyramid structure to get local estimates. These local estimates are pooled by averaging to get an SSIM value for the entire image.

To briefly examine $\tilde{S}_{W,\mathbb{C}}(x, y)$, it is symmetric with respect to x and y and satisfies $-1 \leq \tilde{S}_{W,\mathbb{C}}(x, y) \leq 1$. An advantage to $\tilde{S}_{W,\mathbb{C}}(x, y)$ is that it is immune to local phase shifts in the image. It is, however, missing the factors of $\frac{1}{N-1}$ present in $S_{W,\mathbb{C}}(x, y)$. It remains to be seen whether taking the modulus of the numerator is the best way to compare SSIM values.

B.3 An extension to vector-valued signals

Applying the same type of argument as above, one could consider many different bases on $\mathbb{C}^N$ or $\mathbb{R}^N$ and come up with many different expressions for the SSIM function. However, a truly interesting and unique way of looking at the SSIM function arises when vector-valued signals on $\mathbb{R}^N$ are considered. This includes, for example, the RGB colour system where each sample is a triplet. In this section some preliminary ideas on this subject are provided along with issues that were encountered.

Let x and y be two N-vectors where each of the N samplings are n-dimensional real vectors. Before the extension to SSIM is considered, it is once again necessary to redefine the notions of the mean, variance and covariance of the signals in question. Let

$$x = (x_1, x_2, \cdots, x_N)$$

where

$$x_k = (x_{k_1}, x_{k_2}, \cdots, x_{k_n}), \quad 1 \leq k \leq N.$$

The mean of x is defined as

$$\begin{aligned} \bar{x} &= \frac{1}{N} \sum_{k=1}^{N} \overline{x_k} \\ &= \frac{1}{nN} \sum_{k=1}^{N} \sum_{l=1}^{n} x_{k_l}. \end{aligned}$$

Instead of a single value for the sample covariance between two signals x and y, there will now be an n-by-n matrix $\mathcal{C} = (c_{ij})$ called the covariance matrix between x and y whose entries are given by

$$c_{ij} = \frac{1}{N-1} \sum_{k=1}^{N} (x_{i_k} - \overline{x_i})(y_{j_k} - \overline{y_j}), \quad 1 \leq i, j \leq n. \tag{B.3.1}$$

Letting $y = x$ in (B.3.1) gives the definition for the covariance matrix of the signal x. Again, this will be an $n \times n$ matrix $\mathcal{S}_x^2 = (s_{x,ij}^2)$ whose entries are given by

$$s_{x,ij}^2 = \frac{1}{N-1} \sum_{k=1}^{N} (x_{i_k} - \overline{x_i})(x_{j_k} - \overline{x_j}), \quad 1 \leq i, j \leq n \tag{B.3.2}$$

Note that $\mathcal{S}^2$ is symmetric with respect to the main diagonal.

Before an attempt to make sense of the SSIM function defined in terms of these new quantities $\mathcal{C}$ and $\mathcal{S}_x^2$ can be made, several issues need to be resolved:

- What is the interpretation of $\frac{1}{\mathcal{S}_x^2 + \mathcal{S}_y^2}$?
- How is matrix division interpreted in the SSIM function?
- The SSIM function needs to be made consistent with the matrix/vector dimensions.

At the time of this writing, this is an open problem which will need to be addressed at a later date.

B.4 Maximizing structural similarity between two real-valued signals

In this section the parameters α and β that maximize $S(u, \alpha v + \beta)$ are derived. To start, it is easily seen that when one makes the substitutions $x = u$ and $y = \alpha v + \beta$ in (B.1.1) that

$$\begin{aligned} S_1(u, \alpha v + \beta) &= \frac{2\bar{u}(\alpha\bar{v} + \beta)}{\bar{u}^2 + (\alpha v + \beta)^2}, \quad \text{and} \\ S_2(u, \alpha v + \beta) &= \frac{2\alpha s_{uv}}{s_u^2 + \alpha^2 s_v^2}. \end{aligned}$$

As is well known, to find the optimal parameters α and β, set

$$\frac{\partial S(u, \alpha v + \beta)}{\partial \alpha} = \frac{\partial S(u, \alpha v + \beta)}{\partial \beta} = 0. \tag{B.4.1}$$

Note that the above expression for $S_2(u, \alpha v + \beta)$ does not depend on β so the second condition in (B.4.1) will only involve S_1. First compute

$$\frac{\partial S_1(u, \alpha v + \beta)}{\partial \beta} = \frac{2\bar{u}[\bar{u}^2 - (\alpha\bar{v} + \beta)^2]}{[\bar{u}^2 + (\alpha\bar{v} + \beta)^2]^2}.$$

Clearly $\frac{\partial S_1}{\partial b} = 0$ only when $\bar{u} = \pm(\alpha\bar{v} + \beta)$ or

$$\beta = \bar{u} \mp \alpha\bar{v}.$$

If $\beta = \bar{u} + \alpha\bar{v}$ then $S_1 = -1$. This case then corresponds to an absolute minimum. The absolute maximum is achieved when $\beta = \bar{u} - \alpha\bar{v}$ and in this case, $S_1 = 1$.

Now compute

$$\frac{\partial S_2(u, \alpha v + \beta)}{\partial \alpha} = \frac{2 s_{uv}}{[s_u^2 + \alpha^2 s_v^2]^2}[s_u^2 - \alpha^2 s_v^2] = 0. \tag{B.4.2}$$

If $\frac{\partial S_2}{\partial \alpha} = 0$ then clearly $\alpha^2 = \frac{s_u^2}{s_v^2}$ which implies that

$$\alpha = \pm\frac{s_u}{s_v}.$$

From (B.4.2), it is easy to see that $\alpha = \frac{s_u}{s_v}$ corresponds to a maximum of S_2 when $s_{uv} > 0$ and a minimum of S_2 when $s_{uv} < 0$. Further, since S_1 is maximized for $\beta = \bar{u} - \alpha\bar{v}$ regardless of the value of α, it follows that $\alpha = \frac{s_u}{s_v}$ corresponds to a maximum of S when $s_{uv} > 0$ and a minimum of S when $s_{uv} < 0$. Therefore, the parameters α and β that maximize $S(u, \alpha v + \beta)$ are

$$\alpha = \operatorname{sgn}(s_{uv})\frac{s_u}{s_v} \quad \text{and} \quad \beta = \bar{u} - \alpha\bar{v}.$$

B.5 Maximizing structural similarity for real-valued wavelet expansions

This section shows the derivation for the coefficient α that maximizes $S_{W,\mathbb{R}}(u, \alpha v)$. Let $x, y \in \mathbb{R}^N$ be two signals. The real-valued wavelet-based structural similarity function is given by

$$S_{W,\mathbb{R}}(x, y) = \frac{2\displaystyle\sum_{k=1}^{N} a_k c_k}{\displaystyle\sum_{k=1}^{N} [a_k^2 + c_k^2]}. \tag{B.5.1}$$

As it is not important in what follows, we omit any discussion of this formula here and direct the reader to Sec. 3.2.2 for details. Make the substitutions $x = u$ and $y = \alpha v$ in the formula for $S_{W,\mathbb{R}}(x, y)$. The expansion coefficients of αv are simply the coefficients of v scaled by α. Thus we have that

$$S_{W,\mathbb{R}}(u, \alpha v) = \frac{2\alpha\displaystyle\sum_{k=1}^{N} a_k c_k}{\displaystyle\sum_{k=1}^{N} [a_k^2 + (\alpha c_k)^2]}.$$

Now compute $\frac{dS_{W,\mathbb{R}}(u,\alpha v)}{d\alpha}$:

$$\frac{dS_{W,\mathbb{R}}(u, \alpha v)}{d\alpha} = \frac{\left[2\displaystyle\sum_{k=1}^{N} a_k c_k\right]\left[\displaystyle\sum_{k=1}^{N} a_k^2 - \alpha^2 c_k^2\right]}{\left[\displaystyle\sum_{k=1}^{N} a_k^2 + \alpha^2 c_k^2\right]^2}. \tag{B.5.2}$$

Setting $\frac{dS_{W,\mathbb{R}}}{d\alpha}(u, \alpha v) = 0$ and solving for α yields $\alpha^2 = \frac{\sum_{k=1}^{N} a_k^2}{\sum_{k=1}^{N} c_k^2}$ which implies that

$$\alpha_{\pm} = \pm \sum_{k=1}^{N} a_k^2 \Bigg/ \sum_{k=1}^{N} c_k^2 .$$

Upon examining (B.5.2), it is clear then that if $\sum_{k=1}^{N-1} a_k c_k > 0$ then α_+ corresponds to a maximum of $S_{W,\mathbb{R}}$ and that if $\sum_{k=1}^{N-1} a_k c_k < 0$ then α_+ corresponds to a minimum. Summarizing the above, the parameter α that maximizes $S_{W,\mathbb{R}}(u, \alpha v)$ is given by

$$\alpha = \operatorname{sgn}\left(\sum_{k=1}^{N} a_k c_k\right) \sum_{k=1}^{N} a_k^2 \Bigg/ \sum_{k=1}^{N} c_k^2 .$$