Inverse Problems and Self-similarity in Imaging

by

Mehran Ebrahimi Kahrizsangi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Applied Mathematics

Waterloo, Ontario, Canada, 2008

© Mehran Ebrahimi Kahrizsangi 2008

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 examines the concept of image self-similarity and provides solutions to various associated inverse problems such as resolution enhancement and missing fractal codes.

In general, many real-world inverse problems are ill-posed, mainly because of the lack of existence of a unique solution. The procedure of providing acceptable unique solutions to such problems is known as *regularization*. The concept of *image prior*, which has been of crucial importance in image modelling and processing, has also been important in solving inverse problems since it algebraically translates to the regularization procedure.

Indeed, much recent progress in imaging has been due to advances in the formulation and practice of regularization. This, coupled with progress in optimization and numerical analysis, has yielded much improvement in computational methods of solving inverse imaging problems.

Historically, the idea of self-similarity was important in the development of fractal image coding. Here we show that the self-similarity properties of natural images may be used to construct image priors for the purpose of addressing certain inverse problems. Indeed, new trends in the area of *non-local image processing* have provided a rejuvenated appreciation of image self-similarity and opportunities to explore novel self-similarity-based priors.

We first revisit the concept of fractal-based methods and address some open theoretical problems in the area. This includes formulating a necessary and sufficient condition for the contractivity of the block fractal transform operator. We shall also provide some more generalized formulations of fractal-based self-similarity constraints of an image. These formulations can be developed algebraically and also in terms of the set-based method of Projection Onto Convex Sets (POCS).

We then revisit the traditional inverse problems of single frame image zooming and multi-frame resolution enhancement, also known as "super-resolution". Some ideas will be borrowed from newly developed non-local denoising algorithms in order to formulate self-similarity priors. Understanding the role of scale and choice of examples/samples is also important in these proposed models. For this purpose, we perform an extensive series of numerical experiments and analyze the results. These ideas naturally lead to the method of "self-examples," which relies on the regularity properties of natural images at different scales, as a means of solving the single-frame image zooming problem.

Furthermore, we propose and investigate a multi-frame super-resolution counterpart which does not require explicit motion estimation among video sequences.

Acknowledgements

I wish to thank my supervisor, Prof. Edward Vrscay, for giving me the chance to pursue my graduate studies at the University of Waterloo. Ed's notable support, friendly attitude, and encouragement in many ways during my PhD years are deeply appreciated.

I would like to acknowledge the Department of Applied Mathematics, Faculty of Mathematics, University of Waterloo, and the Province of Ontario (Ontario Graduate Scholarship) for various forms of financial support.

I wish to thank the members of my Advisory Committee – Profs. S.P. Lipshitz, J. Orchard and G. Freeman of the University of Waterloo – as well as the External Examiner, Prof. W. Kinsner of the University of Manitoba. Their valuable comments and suggestions are highly appreciated. I wish to express sincere appreciation to many faculty members, colleagues, and individuals who have offered support, inspiration, and encouragement throughout this research effort.

Dedication

This thesis is dedicated to my wife, Mojdeh, for her love, sacrifice, and patience.

This work is also dedicated to my parents, Ahmad and Tooran, who have raised me to be the person I am today. Their guidance and support throughout my entire educational process will always be remembered.

Contents

List of Tables

List of Figures

List of Symbols

Font and Style

Ω,Φ, $\mathfrak{C}$, $\mathfrak{P}$, $\mathscr{X}$,$\mathscr{A}$	Sets
$\mathbf{M}$, $\mathbf{H}$, $\mathbf{\Sigma}$	Matrices
$\mathbf{u}$, $\mathbf{v}$, $\mathbf{n}$	Images
$\underline{\mathbf{u}}$, $\underline{\mathbf{v}}$, $\underline{\mathbf{n}}$	Images in the vector order
T, G, $\mathcal{H}$, $\mathcal{D}$, $\mathcal{U}$	Operators acting on images
$\underline{T}$, $\underline{G}$, $\underline{\mathcal{H}}$, $\underline{\mathcal{D}}$, $\underline{\mathcal{U}}$	Operators acting on images in the vector order
x, M, $w(x,y)$, δ	Variables, functions, etc.

Set of Numbers

$\mathbb{N}$	Positive integer numbers
$\mathbb{Z}$	Integer numbers
$\mathbb{R}$	Real numbers

Standard

$\lceil \quad \rceil$	Ceiling function
$\lfloor \quad \rfloor$	Floor function
1_{ψ}	Characteristic function on Ψ
$\ell^p(\Xi)$	ℓ^p spaces
$\| \quad \|_p$	p-norm
$< \mathbf{u}, \mathbf{v} >$	Inner product
$\nabla \mathbf{x}$	Gradient of $\mathbf{x}$
$\nabla \cdot \underline{v}$	Divergence of $\underline{v}$

Probability

$P(\underline{\mathbf{x}})$	Probability density function
$E(\underline{\mathbf{x}})$	Expected value
$Var(\underline{\mathbf{x}})$	Variance
$Cov(\underline{\mathbf{x}}, \underline{\mathbf{y}})$	Covariance

Matrices

$\mathbf{M}^T$	Transpose of $\mathbf{M}$
$\mathbf{I}_{M \times M}$	$M \times M$ identity matrix
$\mathbf{M}^{-1}$	Inverse of $\mathbf{M}$
$Rank(\mathbf{M})$	Rank of $\mathbf{M}$
$Null(\mathbf{M})$	Null-space of $\mathbf{M}$
$Range(\mathbf{M})$	Range of $\mathbf{M}$
$\|\mathbf{M}\|$	Frobenius norm of $\mathbf{M}$
$\|\mathbf{M}\|_{2-norm}$	2-norm of $\mathbf{M}$
$\|\mathbf{M}\|_{\infty-norm}$	Infinity-norm of $\mathbf{M}$
$\rho(\mathbf{M})$	Spectral radius of $\mathbf{M}$
$\mathbf{diag}(d_i)$	Diagonal matrix

Operators

$\mathcal{A}_z$	Local averaging operator
$\mathcal{S}_z$	Down-sampling operator
$\mathcal{D}_z$	Decimation operator
$\mathcal{U}_z$	Up-sampling operator

Transforms

$DFT(\mathbf{u})$	Discrete Fourier transform
$DWT(\mathbf{u})$	Discrete wavelet transform

Common Sets

Υ	$[1,\ldots,N] \times [1,\ldots,N]$
Ω	$[1,\ldots,M] \times [1,\ldots,N]$
Φ	$[1,\ldots,K] \times [1,\ldots,L]$
Ψ	$[1,\ldots,P] \times [1,\ldots,Q]$

Note: These symbols are extensively described in Section 1.4 (Notations and Preparation).

Chapter 1

Introduction

1.1 Imaging and Inverse Problems

1.1.1 General Model of an Inverse Problem

In very many practical problems in the field of applied sciences, the features of most interest cannot be observed directly, but have to be inferred from other, observable quantities. In the simplest approximation, which works quite well in a wide range of cases, there is a linear relationship between the features of interest and the derived quantities. If we model the object by a function $\mathbf{x}$, and the observed quantities or image by another function $\mathbf{h}$, we can pose the problem of inferring $\mathbf{x}$ from $\mathbf{h}$ as an *inverse problem*, the task of which is to solve the equation

$$\mathcal{H}(\mathbf{x}) = \mathbf{h}. \tag{1.1}$$

This equation and the task of solving it is sensible only when placed in some appropriate framework. In this thesis, we shall assume that $\mathbf{x}$ and $\mathbf{h}$ belong to appropriate function spaces, typically Banach or Hilbert spaces, $\mathbf{x} \in \mathcal{B}_{OBJECT}$, $\mathbf{h} \in \mathcal{B}_{IMAGE}$, and that $\mathcal{H}$ is a (typically linear) operator from the space $\mathcal{B}_{OBJECT}$ to $\mathcal{B}_{IMAGE}$. The choice of the spaces must be appropriate for describing real-life situations.

The observations or *data*, which we model by yet another function, $\mathbf{u}$, are not exactly equal to the image $\mathbf{h} = \mathcal{H}(\mathbf{x})$, but rather to a distortion of $\mathbf{h}$, in a realistic framework. This distortion is often modelled by an additive noise or error term $\mathbf{n}$, i.e.

$$\mathbf{u} = \mathbf{h} + \mathbf{n} = \mathcal{H}(\mathbf{x}) + \mathbf{n}. \tag{1.2}$$

Moreover, one typically assumes that the "size" of the noise can be measured by its norm, $\|\mathbf{n}\|$ in its appropriate Banach or Hilbert space. Our only handle is thus via the observed $\mathbf{u}$, and we typically have little information on $\mathbf{n} = \mathbf{u} - \mathbf{h}$ beyond an upper bound on its norm $\|\mathbf{n}\|$ in the so-called *deterministic setting*. In the *stochastic setting*, further assumptions are taken into account instead, e.g., the variance of the components of $\mathbf{n}$ may be assumed known.

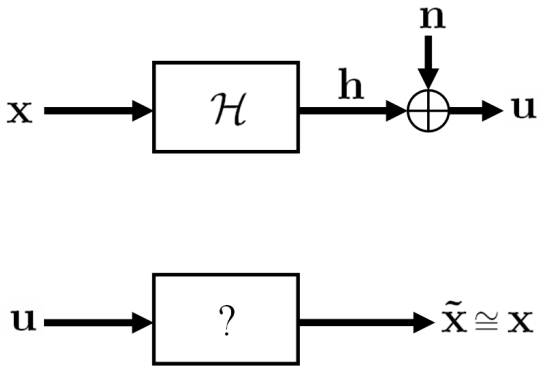


Figure 1.1: Inverse problem.

Finally, the **inverse problem** can be summarized as in the following statement: Given the observations $\mathbf{u}$ find an approximation of the object $\mathbf{x}$ denoted by $\tilde{\mathbf{x}}$ (see Figure 1.1).

1.1.2 Imaging Tasks as Inverse Problems

Many image processing tasks can be modelled as inverse problems. The classical problem of image denoising can be described as an inverse problem in which the operator $\mathcal{H}$ is the identity operator. Image zooming, which we consider in this thesis, is another inverse problem in which the operator $\mathcal{H}$ can be modelled as a composition of several linear operators, e.g., geometric warp, blur, and down-sampling. Many other important classical problems in imaging, including image reconstruction from samples, compression, deblurring, segmentation, and registration, can be modelled as inverse problems.

1.2 A Word on Self-similarity

Self-similarity has been an important and progressive concept in the imaging community in the past few decades. When speaking of self-similarity, we are not just referring to the traditional fractal coding context [BD85, Bar88, Fis95, Lu97], in which subblocks of an image are approximated using other subblocks of the same image in general. Indeed our discussions and results contain a rather wide and general class of techniques that involve self-similarity properties of images. This includes the recently developed emerging "non-local" methods that employ the concept of self-similarity as a key tool in solving the associated inverse problems, e.g., the classical inverse problems of image and video denoising [BCM05b, BCM05c, BCM05a].

In this thesis, we shall deal with various existing problems related to image self-similarity that can be modelled and addressed using inverse problems [EV06a, EV06b]. Furthermore, we will present various extensions of the current non-local methods [BCM05b, BCM05c, BCM05a] to address the inverse problems of image and video zooming [EV08b, EV08a, EV07a] by employing self-similarity properties of images or image sequences.

1.3 Thesis Organization

Throughout this thesis, we will be dealing with approximation of an object (image) $\mathbf{x}$. In general, such approximation will be represented as

$$\tilde{\mathbf{x}}_{Method,Parameters}$$

in which the *Method* and *Parameters* of the corresponding approximation process are denoted in the subscript.

In the next section, we will introduce the notations used throughout. Various sources are used in preparation of these materials including [NS82, AH77, Bra86, Jai01, LR99, Mal98, Vog02]. In Chapters 2 and 3, the background material on inverse theory techniques and self-similarity in imaging inverse problems will be covered. Chapters 4 to 10 will include the author's contribution to the research area. Finally, a summary of these contributions and the author's future perspectives will follow in Chapter 11.

1.4 Notations and Preparation

1.4.1 Familiar Concepts

It is assumed that the reader is familiar with elementary linear algebra, calculus, and the following concepts.

- Metric space, Field, Vector space, Normed linear space, Banach space, Hilbert space
- Measure space, Invariant measure, Borel probability measure, Uniform and Gaussian probability distributions

1.4.2 Consistent Font and Style

- **Sets**
  Sets are denoted by upper-case greek letters, upper-case fraktur or upper-case script fonts e.g., Ω,Φ,Δ,Ψ, Ξ, $\mathfrak{C}$, $\mathfrak{P}$, $\mathfrak{R}$, $\mathfrak{S}$, $\mathscr{X}$,$\mathscr{A}$. Number of elements of a finite set $\mathfrak{C}$ is denoted by $|\mathfrak{C}|$.
- **Matrices**
  Matrices are represented by bold upper-case normal or upper-case greek letters, e.g., $\mathbf{M}$, $\mathbf{A}$, $\mathbf{I}$, $\mathbf{H}$, $\mathbf{\Psi}$, $\mathbf{\Phi}$, $\mathbf{\Sigma}$.
- **Images**
  Images are represented by bold lower-case letters, e.g., $\mathbf{u}$, $\mathbf{v}$, $\mathbf{n}$. In this thesis, images are typically assumed as elements of $\ell^2(\Xi)$ for some finite set Ξ.
- **Images represented in the (column) vector order**
  Underlined bold lower-case letters will be used for images represented in the vector format, e.g., $\underline{\mathbf{u}}$, $\underline{\mathbf{v}}$, $\underline{\mathbf{n}}$. The notation $\underline{\overleftarrow{\cdot}}$ is used to represent the conversion from the vector order to the normal representation defined above, i.e., $\mathbf{u} = \underline{\overleftarrow{\mathbf{u}}}$.
- **Operators acting on images**
  Upper-case (normal or calligraphic) fonts will be used to represent operators acting on images, e.g., T, G, $\mathcal{H}$, $\mathcal{D}$, $\mathcal{U}$, $\mathcal{P}$.
- **Operators acting on images represented in the vector format**
  Underlined upper-case fonts will be used for operators acting on images represented in the vector format, e.g., $\underline{T}$, $\underline{G}$.
- **Variables, functions, etc.**
  Variables, functions, and other quantities will be denoted in terms of normal (not bold, nor underlined) english or greek characters, e.g, x, y, M, N, $w(x,y)$, ϕ, δ, λ. To avoid any possible overlap with some operators precise definitions of these expressions are presented.

1.4.3 Standard Notations

Various General Notations

- **Set of numbers**
  Positive integer numbers: $\mathbb{N}$, Integer numbers: $\mathbb{Z}$, Real numbers: $\mathbb{R}$

- **ℓ^p spaces**
  The ℓ^p space over a set Ξ is denoted by $\ell^p(\Xi)$, for $1 \leq p \leq \infty$. This is a Banach space in general and $\|\mathbf{f}\|_p$ denotes the p-norm of $\mathbf{f} \in \ell^p(\Xi)$. The space $\ell^2(\Xi)$ is a Hilbert space as well, equipped with an inner product. The inner product of $\mathbf{u}$ and $\mathbf{v}$ in $\ell^2(\Xi)$ is denoted by $< \mathbf{u}, \mathbf{v} >$.

- **Ceiling and floor functions**
  The ceiling function, denoted by $\lceil r \rceil$ for any $r \in \mathbb{R}$, returns the smallest integer not less than r, e.g., $\lceil 1.4 \rceil = \lceil 2 \rceil = 2$. The floor function, denoted by $\lfloor r \rfloor$, returns the highest integer less than or equal to r, e.g., $\lfloor 1.4 \rfloor = \lfloor 1 \rfloor = 1$.

- **Characteristic function**
  The characteristic (or indicator) function on a set Ψ is denoted by 1_ψ.

- **Gradient of a function**
  The vector derivative of a function $\mathbf{x} : \mathbb{R}^N \to \mathbb{R}$, if it exists, is called the gradient of $\mathbf{x}$ and is denoted by $\nabla \mathbf{x}$.

- **Divergence of a vector field**
  Divergence of a vector field $v : \mathbb{R}^N \to \mathbb{R}^N$, $\underline{v}(x_1, x_2, \ldots, x_N) = (v_1, v_2, \ldots, v_N)$ is denoted by $\nabla \cdot \underline{v}$ and is defined as in the following expression if it exists

$$\nabla \cdot \underline{v} = \sum_{i=1}^{N} \frac{\partial v_i}{\partial x_i}. \tag{1.3}$$

Notations Involving Random Vectors

- **Probability**
  Probability density function of a random vector $\underline{\mathbf{x}}$ is represented by $P(\underline{\mathbf{x}})$.

- **Expected value function**
  Expected value of a random vector $\underline{\mathbf{x}}$ is denoted by $E(\underline{\mathbf{x}})$.

- **Variance**
  Variance of a random vector $\underline{\mathbf{x}}$ is denoted by $Var(\underline{\mathbf{x}})$ and is defined as
  $Var(\underline{\mathbf{x}}) = E(\underline{\mathbf{x}}^2) - E(\underline{\mathbf{x}})^2$. The square root of $Var(\underline{\mathbf{x}})$ is called the standard deviation of $\underline{\mathbf{x}}$.

- **Covariance**
  The covariance between two random vectors $\underline{\mathbf{x}}$ and $\underline{\mathbf{y}}$ is denoted by $Cov(\underline{\mathbf{x}}, \underline{\mathbf{y}})$ and is defined as $Cov(\underline{\mathbf{x}}, \underline{\mathbf{y}}) = E(\underline{\mathbf{x}} \cdot \underline{\mathbf{y}}) - E(\underline{\mathbf{x}})E(\underline{\mathbf{y}})$.

Notations Involving Matrices

- **Transpose of a matrix**
  Transpose of a matrix $\mathbf{M}$ is denoted by $\mathbf{M}^T$. All of the matrices in this thesis are assumed to be real-valued. Hence, the conjugate transpose of any matrix equals its transpose.

- **Identity matrix**
  The $M \times M$ identity matrix will be denoted by $\mathbf{I}_{M\times M}$.

- **Inverse of a matrix**
  Inverse of a square matrix $\mathbf{M}$, if exists, is denoted by $\mathbf{M}^{-1}$.

- **Diagonal matrix**
  The notation $\mathbf{M} = \mathbf{diag}(d_i)$ means that $\mathbf{M}$ is a diagonal matrix with diagonal elements d_i placed on the row and column i.

- **Rank of a matrix**
  The rank of a matrix $\mathbf{M}$ denoted by $Rank(\mathbf{M})$ is defined as the maximal number of its linearly independent rows (or columns).

- **Null-space and range of a matrix**
  The null-space and range of a matrix $\mathbf{M}$ are respectively denoted by $Null(\mathbf{M})$ and $Range(\mathbf{M})$.

- **Frobenius norm**
  The Frobenius norm of a matrix $\mathbf{M}$ is defined as

$$\|\mathbf{M}\| = \Big\{ \sum_{i,j} {\mathbf{M}_{i,j}}^2 \Big\}^{\frac{1}{2}}. \tag{1.4}$$

  All of the norms in this thesis are assumed to be the Frobenius norm unless otherwise stated.

- **2-norm**
  The 2-norm of a matrix $\mathbf{M}$ is defined as

$$\|\mathbf{M}\|_{2-norm} = \sup_{\underline{\mathbf{x}}\neq\underline{\mathbf{0}}} \frac{\|\mathbf{M}\underline{\mathbf{x}}\|}{\|\underline{\mathbf{x}}\|}. \tag{1.5}$$

- **Infinity-norm**
  The infinity-norm of a matrix $\mathbf{M}$ is defined as

$$\|\mathbf{M}\|_{\infty-norm} = \max_i \sum_j |\mathbf{M}_{i,j}|. \tag{1.6}$$

- **Spectral radius**
  The spectral radius a matrix $\mathbf{M}$ denoted by $\rho(\mathbf{M})$ is defined as

$$\rho(\mathbf{M}) = \max\{|\lambda| : \lambda \text{ belongs to the set of eigenvalues of } \mathbf{M}\}. \tag{1.7}$$

Important Definitions and Propositions Involving Matrices

- **Normalized vector**
  A vector $\underline{\mathbf{x}}$ is called normalized if $\|\underline{\mathbf{x}}\| = 1$.

- **Rank and invertibility**
  Rank of any matrix $\mathbf{M}$ and its corresponding Gram matrix are equal, i.e.

$$rank(\mathbf{M}) = rank(\mathbf{M}^T\mathbf{M}) = rank(\mathbf{M}\mathbf{M}^T). \tag{1.8}$$

  Any $N \times N$ matrix $\mathbf{M}$ is invertible if and only if it is full rank, i.e., if and only if $rank(\mathbf{M}) = N$.

- **Positive definite and positive semidefinite matrix**
  A real matrix $\mathbf{M}$ is called positive definite if for any nonzero real vector $\underline{\mathbf{x}}$, $\underline{\mathbf{x}}^T\mathbf{M}\underline{\mathbf{x}} > 0$. Also, $\mathbf{M}$ is called positive semidefinite if for any nonzero real vector $\underline{\mathbf{x}}$, $\underline{\mathbf{x}}^T\mathbf{M}\underline{\mathbf{x}} \geq 0$.

  The eigenvalues of a symmetric matrix are all non-negative if and only if the matrix is positive semidefinite. Furthermore, the eigenvalues of a symmetric matrix are all positive if and only if the matrix is positive definite. This yields that a symmetric matrix is invertible if and only if the matrix is positive definite.

- **Matrix norm properties**
  Frobenius norm and 2-norm of any vector are the same. Also, for any $M \times N$ matrix $\mathbf{M}$ the folowing inequalities hold.

$$\frac{1}{\sqrt{N}}\|\mathbf{M}\|_{\infty-norm} \leq \|\mathbf{M}\|_{2-norm} \leq \|\mathbf{M}\| \leq \sqrt{N}\|\mathbf{M}\|_{2-norm} \leq \sqrt{MN}\|\mathbf{M}\|_{\infty-norm}. \tag{1.9}$$

- **Spectral radius properties**
  For any matrix $\mathbf{M}$ and any norm $\|.\|$, e.g., either Frobenius, 2-norm, or infinity-norm defined earlier, and any positive integer k

$$\rho(\mathbf{M}) = \lim_{k\to\infty} \|\mathbf{M}^k\|^{\frac{1}{k}} \leq \|\mathbf{M}^k\|^{\frac{1}{k}}. \tag{1.10}$$

- **Stochastic matrix**
  A real square matrix $\mathbf{M}$ with nonnegative elements is called a right(left) stochastic matrix if the sum of the elements of every row(column) is one. The matrix $\mathbf{M}$ is called doubly stochastic matrix if $\mathbf{M}$ is both right and left stochastic.

- **Perron-Frobenius theorem**
  For any right stochastic matrix $\mathbf{M}$ there exists a probability vector $\underline{\mathbf{p}}$, i.e., a vector with non-negative elements of sum one, such that

$$\underline{\mathbf{p}}\mathbf{M} = \underline{\mathbf{p}}. \tag{1.11}$$

  The j-th element of $\underline{\mathbf{p}}$ may be computed, independent of i, by taking the limit

$$\lim_{k\to\infty} (\mathbf{M}^k)_{(i,j)} = \underline{\mathbf{p}}_j. \tag{1.12}$$

Definitions Involving Fourier Theory

- **Discrete Fourier transform**
  For any image $\mathbf{u} \in \ell^2(\Omega)$, where $\Omega = [1, \ldots, M] \times [1, \ldots, N]$, the discrete Fourier transform of $\mathbf{u}$ denoted by $DFT(\mathbf{u})$ is defined in the following manner.

$$DFT(\mathbf{u})(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \mathbf{u}(x, y) \exp\left[-i2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)\right], \tag{1.13}$$

  where $i = \sqrt{-1}$, and periodic boundary conditions are assumed on $\mathbf{u}$, i.e., $\mathbf{u}(x, 0) = \mathbf{u}(x, N)$, for any $1 \leq x \leq M - 1$, $\mathbf{u}(0, y) = \mathbf{u}(M, y)$, for any $1 \leq y \leq N - 1$, and $\mathbf{u}(0, 0) = \mathbf{u}(M, N)$. It follows that,

$$\mathbf{u}(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} DFT(\mathbf{u})(u, v) \exp\left[i2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)\right]. \tag{1.14}$$

- **Fourier spectrum**
  The magnitude Fourier spectrum of $\mathbf{u}$ is defined as $|DFT(\mathbf{u})(u, v)|$, in which $|.|$ denotes magnitude of a complex number.

- **Bandlimited image**
  Image $\mathbf{u} \in \ell^2(\Omega)$ is called bandlimited of bandwidth b if there exists some real number b such that for any integers u, v that satisfy all three conditions of $|u| \leq \frac{M}{2}$, $|v| \leq \frac{N}{2}$, and $\sqrt{(\frac{u}{M})^2 + (\frac{v}{N})^2} \geq b$,

$$|DFT(\mathbf{u})(u, v)| = 0.$$

  It is assumed that both M and N are even integers. Different conditions are required in the odd case which are not discussed in this thesis.

Important Definitions and Propositions Involving Contraction Maps

- **Contraction map**
  A mapping $T : \ell^2(\Xi) \to \ell^2(\Xi)$ is called a *contraction* (or contractive) if there exists some real number $0 \leq c < 1$, such that for any $\mathbf{u}, \mathbf{v} \in \ell^2(\Xi)$,

  $$\|T(\mathbf{u}) - T(\mathbf{v})\| \leq c\|\mathbf{u} - \mathbf{v}\|.$$

  In this case, c is called a *contractivity factor* of T. The smallest contractivity factor of T (which can be shown to exist if T is a contraction) is called *the contractivity factor* of T.

- **Banach's contraction mapping principle (fixed point theorem)**
  If $T : \ell^2(\Xi) \to \ell^2(\Xi)$ is a contraction, then there exists a unique $\mathbf{u}^\star \in \ell^2(\Xi)$ (referred to as the *fixed point* of T) such that $T(\mathbf{u}^\star) = \mathbf{u}^\star$. Furthermore, the iteration sequence $\mathbf{u}_{n+1} = T(\mathbf{u}_n)$ converges to $\mathbf{u}^\star$, for any $\mathbf{u}_0 \in \ell^2(\Xi)$.

- **Collage theorem**
  If $T : \ell^2(\Xi) \to \ell^2(\Xi)$ is a contraction with a contractivity factor $con_{(T)}$, and the fixed point $\mathbf{u}^\star$, then for any $\mathbf{u} \in \ell^2(\Xi)$

  $$\|\mathbf{u} - \mathbf{u}^\star\| \ \leq \ \frac{1}{1 - con_{(T)}}\|\mathbf{u} - T(\mathbf{u})\|.$$

A note regarding the above definitions and propositions:
General counterparts of the above exist in more abstract spaces, e.g., in metric spaces. However, in this thesis, ℓ^2 spaces will be mainly used.

1.4.4 Various Sets and Operators used Throughout

Common Sets used Throughout

$\Upsilon = [1, \ldots, N] \times [1, \ldots, N]$, typically used for observed images of square size.
$\Omega = [1, \ldots, M] \times [1, \ldots, N]$, typically used for observed images.
$\Phi = [1, \ldots, K] \times [1, \ldots, L]$, typically used for the so-called *example* images.
$\Psi = [1, \ldots, P] \times [1, \ldots, Q]$, typically used for ideal images (objects) to be recovered.
In Chapters 9 and 10, $P = Mz$ and $Q = Nz$, i.e., $\Psi = [1, \ldots, Mz] \times [1, \ldots, Nz]$.
Ξ could be any of these defined sets, i.e., either Υ, Ω, Φ, or Ψ.

Common Operators used Throughout

- **Local averaging operator**
  The local averaging operator of length z, $\mathcal{A}_z : \ell^2(\Xi) \to \ell^2(\Xi)$ for any $(i, j) \in \Xi$ is defined as

$$(\mathcal{A}_z \mathbf{x})(i, j) = \frac{1}{z^2} \sum_{0 \le i' < z, 0 \le j' < z} \mathbf{x}(i + i', j + j'). \tag{1.15}$$

  Boundary conditions on $\mathbf{x}$ may be required for $\mathcal{A}_z$ to be well-defined.

- **Down-sampling operator**
  The down-sampling operator $\mathcal{S}_z : \ell^2(\Xi) \to \ell^2(\Xi)$ by a factor of $z \in \mathbb{N}$ is defined for any image $\mathbf{x} \in \ell^2(\Xi)$ such that for any $(i, j) \in \Xi$,

$$(\mathcal{S}_z \mathbf{x})(i, j) = \mathbf{x}\Big((i-1)z + 1, (j-1)z + 1\Big). \tag{1.16}$$

- **Coarser-scale projection (decimation) operator**
  The projection operator to a coarser-scale is denoted by $\mathcal{D}_z : \ell^2(\Xi) \to \ell^2(\Xi)$, where $\mathcal{D}_z = \mathcal{S}_z \circ \mathcal{A}_z$.

- **Up-sampling (pixel replication) operator**
  The up-sampling operator by a factor of z, acting on an image $\mathbf{x} \in \ell^2(\Xi)$, is denoted by $\mathcal{U}_z : \ell^2(\Xi) \to \ell^2(\Xi)$, where

$$(\mathcal{U}_z \mathbf{x})(i, j) = \mathbf{x}(\lceil \frac{i}{z} \rceil, \lceil \frac{j}{z} \rceil), \tag{1.17}$$

  for any $(i, j) \in \Xi$. It is typically assumed that the support of the image $\mathbf{x}$ is well contained in Ξ, unless part of the up-sampled image may be lost under this operation.

An important note regarding the above operators:
In the chapters of this thesis where the factor z is precisely defined, $\mathcal{A}_z$, $\mathcal{S}_z$, $\mathcal{D}_z$, and $\mathcal{U}_z$ will be denoted respectively by $\mathcal{A}$, $\mathcal{S}$, $\mathcal{D}$, and $\mathcal{U}$ for simplicity. This happens in Chapters 3 to 6. In Chapter 3 and 4 we will assume $z = s$. In Chapters 5 and 6, $z = 2$.

1.4.5 Image Quality Measures

We will mostly rely on the ℓ^2 distance of images when required in this thesis. There will be a few places that the peak signal-to-noise ratio (PSNR) will be employed instead. Here we give the corresponding definitions.

- **Mean Squared Error (MSE)**
  The MSE of an $M \times N$ image $\mathbf{u} \in \ell^2(\Omega)$ having the ideal image $\mathbf{x} \in \ell^2(\Omega)$ is defined as

$$MSE(\mathbf{u}, \mathbf{x}) = \frac{\|\mathbf{u} - \mathbf{x}\|^2}{MN}. \tag{1.18}$$

- **Root Mean Squared Error (RMSE)**
  The RMSE of $\mathbf{u}$ having $\mathbf{x}$ is defined as

$$RMSE(\mathbf{u}, \mathbf{x}) = \sqrt{MSE(\mathbf{u}, \mathbf{x})}. \tag{1.19}$$

- **Peak Signal-to-Noise Ratio (PSNR)**
  The PSNR of $\mathbf{u}$ having $\mathbf{x}$ is defined in decibels (dB) as

$$PSNR(\mathbf{u}, \mathbf{x}) = 20 \log_{10} \left(\frac{\max_{i,j} \{\mathbf{x}_{i,j}\}}{RMSE(\mathbf{u}, \mathbf{x})} \right) dB. \tag{1.20}$$

1.4.6 Singular Value Decomposition (SVD) Expansion

Given any $M \times N$ real matrix $\mathbf{H}$, it can be shown that the eigenvalues of $\mathbf{H}^T\mathbf{H}$ are all real and non-negative. Assume that among all these eigenvalues, r of them denoted by σ_m, $1 \leq m \leq r$, are positive and to each corresponds an $N \times 1$ normalized eigenvector $\mathbf{\Phi}_m$. Hence,

$$\mathbf{H}^T\mathbf{H}\mathbf{\Phi}_m = \sigma_m\mathbf{\Phi}_m, 1 \leq m \leq r. \tag{1.21}$$

Furthermore, without loss of generality, assume that the eigenvalues σ_m are arranged in decreasing order,

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0. \tag{1.22}$$

Multiplying Equation 1.21 by $\mathbf{H}$ from the left gives

$$\mathbf{H}\mathbf{H}^T(\mathbf{H}\mathbf{\Phi}_m) = \sigma_m(\mathbf{H}\mathbf{\Phi}_m), 1 \leq m \leq r. \tag{1.23}$$

If for any m, $1 \leq m \leq r$, we define the properly normalized $M \times 1$ vectors

$$\mathbf{\Psi}_m = \frac{\mathbf{H}\mathbf{\Phi}_m}{\sqrt{\sigma_m}}, \tag{1.24}$$

then Equation 1.23 can be written as

$$\mathbf{H}\mathbf{H}^T(\mathbf{\Psi}_m) = \sigma_m(\mathbf{\Psi}_m), 1 \leq m \leq r. \tag{1.25}$$

Hence, by this definition, the vectors $\mathbf{\Psi}_m$, $1 \leq m \leq r$, become the normalized eigenvectors of $\mathbf{H}\mathbf{H}^T$. It can be shown that each set of vectors $\{\mathbf{\Phi}_m\}_{1\leq m\leq r}$ and $\{\mathbf{\Psi}_m\}_{1\leq m\leq r}$ is (or can be chosen to be, in the case of some repeated eigenvalues σ_m) *orthonormal* and *complete* in, respectively, the space of $N \times N$ and $M \times M$ matrices, i.e., the following equalities hold,

$$\mathbf{\Phi}_m{}^T\mathbf{\Phi}_n = 1_{\{m\}}(n) \tag{1.26}$$

$$\sum_{m=1}^{r} \mathbf{\Phi}_m\mathbf{\Phi}_m{}^T = \mathbf{I}_{N\times N} \tag{1.27}$$

$$\mathbf{\Psi}_m{}^T\mathbf{\Psi}_n = 1_{\{m\}}(n) \tag{1.28}$$

$$\sum_{m=1}^{r} \mathbf{\Psi}_m\mathbf{\Psi}_m{}^T = \mathbf{I}_{M\times M}. \tag{1.29}$$

Now consider any $N \times 1$ matrix (vector) $\underline{\mathbf{x}}$. By orthonormality and completeness of $\mathbf{\Phi}_m$, or equivalently by multiplying Equation 1.27 with $\underline{\mathbf{x}}$ from the right side,

$$\underline{\mathbf{x}} = \sum_{m=1}^{r} \mathbf{\Phi}_m < \underline{\mathbf{x}}, \mathbf{\Phi}_m >= \sum_{m=1}^{r} \mathbf{\Phi}_m\mathbf{\Phi}_m{}^T\underline{\mathbf{x}}. \tag{1.30}$$

Multiplying the sides of this equality by $\mathbf{H}$ from the left yields

$$\mathbf{H}\underline{\mathbf{x}} = \sum_{m=1}^{r} \mathbf{H}\mathbf{\Phi}_m\mathbf{\Phi}_m{}^T\underline{\mathbf{x}} = \sum_{m=1}^{r} \sqrt{\sigma_m}\mathbf{\Psi}_m\mathbf{\Phi}_m{}^T\underline{\mathbf{x}}. \tag{1.31}$$

Hence, $\mathbf{H}$ can be represented as

$$\mathbf{H} = \sum_{m=1}^{r} \sqrt{\sigma_m} \mathbf{\Psi}_m {\mathbf{\Phi}_m}^T. \tag{1.32}$$

Now, define $\mathbf{\Sigma}$ to be an $r \times r$ diagonal matrix whose diagonal entries are the positive elements $\sqrt{\sigma_m}$, $1 \leq m \leq r$, i.e.,

$$\mathbf{\Sigma} = \ \mathbf{diag}(\sqrt{\sigma_m}), 1 \leq m \leq r. \tag{1.33}$$

Furthermore, take the matrices $\mathbf{\Psi}$ and $\mathbf{\Phi}$ to be respectively of size $M \times r$ and $N \times r$ such that their m-th columns are, respectively, the vectors $\mathbf{\Psi}_m$ and $\mathbf{\Phi}_m$, for $1 \leq m \leq r$. This leads to

$$\mathbf{H} = \mathbf{\Psi \Sigma \Phi}^T, \tag{1.34}$$

which is known as the *SVD expansion* of $\mathbf{H}$. As a result of the orthonormality and completeness of $\{\mathbf{\Phi}_m\}$ and $\{\mathbf{\Psi}_m\}$, $1 \leq m \leq r$, in the corresponding spaces and the definitions of $\mathbf{\Psi}$ and $\mathbf{\Phi}$, the following equalities can be indicated.

$$\mathbf{\Phi \Phi}^T = \mathbf{\Phi} \mathbf{I}_{r \times r} \mathbf{\Phi}^T = \mathbf{I}_{N \times N} \tag{1.35}$$

$$\mathbf{\Phi}^T \mathbf{\Phi} = \mathbf{\Phi}^T \mathbf{I}_{N \times N} \mathbf{\Phi} = \mathbf{I}_{r \times r} \tag{1.36}$$

$$\mathbf{\Psi \Psi}^T = \mathbf{\Psi} \mathbf{I}_{r \times r} \mathbf{\Psi}^T = \mathbf{I}_{M \times M} \tag{1.37}$$

$$\mathbf{\Psi}^T \mathbf{\Psi} = \mathbf{\Psi}^T \mathbf{I}_{M \times M} \mathbf{\Psi} = \mathbf{I}_{r \times r} \tag{1.38}$$

It can be shown that the SVD expansion of a matrix $\mathbf{H}$ is unique if the positive eigenvalues σ_m are all distinct for $1 \leq m \leq r$. If there are some repeated positive eigenvalues σ_m, $1 \leq m \leq r$, then the SVD expansion is no longer unique. For example if $\mathbf{H} = \mathbf{I}_{2,2}$, then $\mathbf{H} = \Psi \Lambda \Phi^T$ where $\Lambda = \mathbf{I}_{2,2}$, and

$$\Psi = \Phi = \begin{bmatrix} u & \sqrt{1-u^2} \\ \sqrt{1-u^2} & -u \end{bmatrix} \tag{1.39}$$

forms an SVD expansion of $\mathbf{H}$ for any $-1 \leq u \leq 1$.

Chapter 2

Inverse Theory Techniques

2.1 Introduction

2.1.1 Space of Images

Recall the linear degradation model previously introduced in Chapter 1,

$$\mathbf{u} = \mathcal{H}(\mathbf{x}) + \mathbf{n}. \tag{2.1}$$

For convenience throughout this thesis, we make the assumption that the observed image $\mathbf{u}$ lies in the Hilbert space $\ell^2(\Omega)$, i.e., $\mathbf{u} \in \mathcal{B}_{IMAGE} = \ell^2(\Omega)$ where

$$\Omega = [1, \ldots, M] \times [1, \ldots, N]. \tag{2.2}$$

Also, we assume that the object $\mathbf{x}$ (typically an image itself again) lies in the Hilbert space $\ell^2(\Omega)$, i.e., $\mathbf{x} \in \mathcal{B}_{OBJECT} = \ell^2(\Psi)$ where

$$\Psi = [1, \ldots, P] \times [1, \ldots, Q]. \tag{2.3}$$

These assumptions are equivalent to restricting ourselves to both a discrete image and object (an image as well) space. Hence, $\mathbf{x} \in \ell^2(\Psi)$ is the unknown object(image) of size $P \times Q$ to be recovered from the measurements $\mathbf{u} \in \ell^2(\Omega)$ of size $M \times N$. Also, $\mathbf{n} \in \ell^2(\Omega)$ is additive noise of the same size as the measurements, i.e., $M \times N$.

2.1.2 Linearity of the Operator

We assume that $\mathcal{H}$ is a bounded *linear* operator from $\ell^2(\Psi)$ to $\ell^2(\Omega)$. In many imaging applications, $\mathcal{H}$ is an integral operator with a kernel representing the response of the imaging device; in the special case where this linear device is translation-invariant, $\mathcal{H}$ reduces to a convolution operator. Everywhere in this thesis, we assume that $\mathcal{H}$ is precisely known or can be estimated.

As pointed out in the notation of Section 1.4, $\underline{\mathbf{u}}$ refers to the vector representation of an image $\mathbf{u}$. Hence, we may equivalently write the linear algebraic equation corresponding to Equation 2.1

$$\underline{\mathbf{u}} = \mathbf{H}\underline{\mathbf{x}} + \underline{\mathbf{n}}, \tag{2.4}$$

in which, $\mathbf{H}$ is the linear degradation matrix of size $MN \times PQ$ corresponding to $\mathcal{H}$. In addition, $\underline{\mathbf{x}}$ is the unknown vector of size $PQ \times 1$ to be recovered from the observation vector $\underline{\mathbf{u}}$ of size, $MN \times 1$. Similarly, $\underline{\mathbf{n}}$ stands for the additive noise vector of size $MN \times 1$.

Depending on the assumptions over $\underline{\mathbf{n}}$, i.e., deterministic or stochastic setting, different approaches may be pursued to address the corresponding inverse problem. Interestingly these two frameworks are highly related and both yield to the same results as will be reviewed later in this chapter.

2.2 Inverse Estimation and Ill-posedness

2.2.1 Pseudo-solutions

To find an estimate of $\underline{\mathbf{x}}$ from the observed image $\underline{\mathbf{u}}$, one can consider minimizing the *discrepancy function* $\underline{\Delta}(\underline{\mathbf{x}})$ defined as,

$$\underline{\Delta}(\underline{\mathbf{x}}) = \|\mathbf{H}\underline{\mathbf{x}} - \underline{\mathbf{u}}\|^2. \tag{2.5}$$

The minimizers of $\underline{\Delta}(\underline{\mathbf{x}})$ are called *pseudo-solutions* of the inverse problem. If the matrix $\mathbf{H}$ has a trivial null-space, i.e., if

$$Null(\mathbf{H}) = \{\mathbf{x} \in \ell^2(\Psi); \mathbf{H}\underline{\mathbf{x}} = \underline{\mathbf{0}}\} = \{\mathbf{0}\}, \tag{2.6}$$

the discrepancy function has a unique minimizer, given by the so-called *Least Squares (LS)* solution

$$\underline{\tilde{\mathbf{x}}_{LS}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\underline{\mathbf{u}}, \tag{2.7}$$

where $\mathbf{H}^T$ is the transpose of $\mathbf{H}$ (or equivalently the adjoint of the operator $\mathcal{H}$). In this case, $(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ is called the *pseudoinverse* of $\mathbf{H}$. The condition $Null(\mathbf{H}) = \{\mathbf{0}\}$ guarantees that the so-called *Gram matrix* $(\mathbf{H}^T\mathbf{H})$ *is* invertible.

2.2.2 Generalized Inverse

When the Gram matrix $(\mathbf{H}^T\mathbf{H})$ is *not* invertible, there exist infinitely many minimizers of the discrepancy function caused by the non-trivial null-space of the matrix $\mathbf{H}$. In this case, one can choose among the set of pseudo-solutions the unique element $\underline{\mathbf{x}}^\dagger$ of minimal norm, i.e.

$$\underline{\mathbf{x}}^\dagger = \underline{\tilde{\mathbf{x}}_{MNLS}} = \arg\min_{\underline{\mathbf{x}}}\{\|\underline{\mathbf{x}}\|; \underline{\mathbf{x}} \text{ minimizes } \underline{\Delta}(\underline{\mathbf{x}})\}. \tag{2.8}$$

This unique element is called the *generalized solution* or *the Minimum Norm Least Square (MNLS) solution* of the inverse problem. In this case, the map $\mathbf{H}^\dagger : \underline{\mathbf{u}} \to \underline{\mathbf{x}}^\dagger$ is called the *generalized inverse of* $\mathbf{H}$.

Even when $\mathbf{H}^T\mathbf{H}$ is not invertible, $\mathbf{H}^\dagger\underline{\mathbf{u}}$ is well-defined for all $\underline{\mathbf{u}}$ such that $\mathbf{H}^T\underline{\mathbf{u}} \in Range(\mathbf{H}^T\mathbf{H})$. It can be shown that the transformation between $\underline{\mathbf{u}}$ and $\underline{\mathbf{x}}^\dagger$ is linear and unique, and hence for all $\underline{\mathbf{u}}$ such that $\mathbf{H}^T\underline{\mathbf{u}} \in Range(\mathbf{H}^T\mathbf{H})$,

$$\underline{\mathbf{x}}^\dagger = \mathbf{H}^\dagger\underline{\mathbf{u}}. \tag{2.9}$$

Using the singular value decomposition (SVD) expansion of $\mathbf{H}$, covered in Section 1.4.6, one can precisely compute $\mathbf{H}^\dagger$. If $\mathbf{H}$ is of size $MN \times PQ$ and has the SVD expansion

$$\mathbf{H} = \sum_{m=1}^{r} \sqrt{\sigma_m}\boldsymbol{\Psi}_m{\boldsymbol{\Phi}_m}^T = \boldsymbol{\Psi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T \tag{2.10}$$

then $\mathbf{H}^\dagger$ is a $PQ \times MN$ matrix with the SVD expansion

$$\mathbf{H}^\dagger = \sum_{m=1}^{r} \frac{1}{\sqrt{\sigma_m}}\boldsymbol{\Phi}_m{\boldsymbol{\Psi}_m}^T = \boldsymbol{\Phi}\boldsymbol{\Sigma}'\boldsymbol{\Psi}^T, \tag{2.11}$$

where $\mathbf{\Phi}_m$ and $\mathbf{\Psi}_m$ are, respectively, the normalized eigenvectors of $\mathbf{H}^T\mathbf{H}$ and $\mathbf{H}\mathbf{H}^T$ corresponding to positive singular values $\{\sigma_m, 1 \leq m \leq r\}$. Also, $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are the matrices whose columns are respectively the vectors $\mathbf{\Phi}_m$ and $\mathbf{\Psi}_m$, $1 \leq m \leq r$. As well,

$$\mathbf{\Sigma} = \mathbf{diag}(\sqrt{\sigma_m}), \quad 1 \leq m \leq r, \tag{2.12}$$

is the diagonal matrix constructed using σ_m's and

$$\mathbf{\Sigma}' = \mathbf{diag}(\frac{1}{\sqrt{\sigma_m}}), \quad 1 \leq m \leq r, \tag{2.13}$$

is constructed using $\mathbf{\Sigma}$ with every nonzero entry replaced by its reciprocal.

Furthermore, using Equations 2.10 and 2.11, it can be shown that the following relations involving $\mathbf{H}^\dagger$ hold in general.

- $\mathbf{H}\mathbf{H}^\dagger = (\mathbf{H}\mathbf{H}^\dagger)^T$
- $\mathbf{H}^\dagger\mathbf{H} = (\mathbf{H}^\dagger\mathbf{H})^T$
- $\mathbf{H}\mathbf{H}^\dagger\mathbf{H} = \mathbf{H}$
- $\mathbf{H}^\dagger\mathbf{H}\mathbf{H}^T = \mathbf{H}^T$

In addition, if the Gram matrix $(\mathbf{H}^T\mathbf{H})$ is invertible then generalized inverse of $\mathbf{H}$ simply equals its pseudoinverse, i.e.,

$$\mathbf{H}^\dagger = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T. \tag{2.14}$$

2.2.3 Illposedness and Regularization

We mentioned that $\mathbf{H}^\dagger\mathbf{u}$ uniquely exists, even if the Gram matrix $(\mathbf{H}^T\mathbf{H})$ is not invertible. However, the generalized inverse operator $\mathbf{H}^\dagger$ may be unbounded (for the so-called *ill-posed problems*) or have a very large norm (for the so-called *ill-conditioned problems*). This situation happens when the Gram matrix has very small positive eigenvalues $\sigma_m > 0$, for some $1 \leq m \leq r$.

In such instances, the generalized inverse operator has to be replaced by bounded approximants or approximants with smaller norm, so that numerically stable solutions can be defined and used as meaningful approximations of the true solution corresponding to the exact data. This is the issue of regularization that we will be using a great deal of in this thesis.

The concept of regularization may be viewed and addressed in both deterministic and stochastic settings that will be discussed in the following sections.

2.3 Deterministic View

2.3.1 Tikhonov Regularization

From an algebraic point of view, regularization can be performed by adding a convex penalty function to the discrepancy function, thus enforcing the existence of a unique solution. More

technically, consider the following minimization problem

$$\underline{\tilde{\mathbf{x}}_{G,\lambda}} = \arg\min_{\underline{\mathbf{x}}} \|\mathbf{H}\underline{\mathbf{x}} - \underline{\mathbf{u}}\|^2 + \lambda \underline{G}(\underline{\mathbf{x}}). \tag{2.15}$$

In this minimization problem, $\underline{G}(\underline{\mathbf{x}})$ is a convex function and therefore $\underline{\tilde{\mathbf{x}}_{G,\lambda}}$ exists for any $\lambda > 0$. The positive value λ in this case is called the *regularization parameter*.

A simple way of achieving a unique global minimizer is the Tikhonov approach [TA77]. One considers the convex function

$$\underline{G}(\underline{\mathbf{x}}) = \underline{G_{Tikh}}(\underline{\mathbf{x}}) = \|\underline{\mathbf{x}}\|^2, \tag{2.16}$$

which leads to

$$\underline{\tilde{\mathbf{x}}_{Tikh,\lambda}} = (\mathbf{H}^T\mathbf{H} + \lambda \mathbf{I}_{PQ\times PQ})^{-1}\mathbf{H}^T\underline{\mathbf{u}}. \tag{2.17}$$

This is called the Tikhonov approximation of $\underline{\mathbf{x}}$ with the regularization parameter λ. It can easily be shown that the expression $\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}_{PQ\times PQ}$ is positive definite for $\lambda > 0$ and hence invertible. To verify that as $\lambda \to 0$, the regularized solution $\underline{\tilde{\mathbf{x}}_{Tikh,\lambda}}$ converges to the generalized solution $\underline{\mathbf{x}}^\dagger$, consider

$$\begin{aligned}
\underline{\tilde{\mathbf{x}}_{Tikh,\lambda}} &= (\mathbf{H}^T\mathbf{H} + \lambda \mathbf{I}_{PQ\times PQ})^{-1}\mathbf{H}^T\underline{\mathbf{u}} && (2.18)\\
&= [(\mathbf{\Psi\Sigma\Phi}^T)^T(\mathbf{\Psi\Sigma\Phi}^T) + \lambda \mathbf{I}_{PQ\times PQ}]^{-1}(\mathbf{\Psi\Sigma\Phi}^T)^T\underline{\mathbf{u}} && (2.19)\\
&= [(\mathbf{\Phi\Sigma}^T\mathbf{\Psi}^T)(\mathbf{\Psi\Sigma\Phi}^T) + \lambda\mathbf{\Phi}\mathbf{I}_{r\times r}\mathbf{\Phi}^T]^{-1}(\mathbf{\Phi\Sigma}^T\mathbf{\Psi}^T)\underline{\mathbf{u}} && (2.20)\\
&= [(\mathbf{\Phi\Sigma}^T\mathbf{\Sigma\Phi}^T) + \lambda\mathbf{\Phi}\mathbf{I}_{r\times r}\mathbf{\Phi}^T]^{-1}(\mathbf{\Phi\Sigma}^T\mathbf{\Psi}^T)\underline{\mathbf{u}} && (2.21)\\
&= \left[[(\mathbf{\Phi\Sigma}^T\mathbf{\Sigma\Phi}^T) + \lambda\mathbf{\Phi}\mathbf{I}_{r\times r}\mathbf{\Phi}^T]^{-1}\mathbf{\Phi}\right](\mathbf{\Sigma}^T\mathbf{\Psi}^T\underline{\mathbf{u}}) && (2.22)\\
&= \mathbf{\Phi}(\mathbf{\Sigma}^T\mathbf{\Sigma} + \lambda\mathbf{I}_{r\times r})^{-1}(\mathbf{\Sigma}^T\mathbf{\Psi}^T\underline{\mathbf{u}}) && (2.23)\\
&= \mathbf{\Phi}\big[(\mathbf{\Sigma}^T\mathbf{\Sigma} + \lambda\mathbf{I}_{r\times r})^{-1}\mathbf{\Sigma}^T\big]\mathbf{\Psi}^T\underline{\mathbf{u}} && (2.24)\\
&= \mathbf{\Phi}\ \mathbf{diag}[\frac{1}{\sigma_m + \lambda} \times \sqrt{\sigma_m}]\mathbf{\Psi}^T\underline{\mathbf{u}}. && (2.25)
\end{aligned}$$

Taking the limit,

$$\begin{aligned}
\lim_{\lambda\to 0}\underline{\tilde{\mathbf{x}}_{Tikh,\lambda}} &= \lim_{\lambda\to 0}\mathbf{\Phi}\ \mathbf{diag}[\frac{\sigma_m}{\sigma_m + \lambda} \times \frac{1}{\sqrt{\sigma_m}}]\mathbf{\Psi}^T\underline{\mathbf{u}} && (2.26)\\
&= \mathbf{\Phi}\ \mathbf{diag}[\frac{1}{\sqrt{\sigma_m}}]\mathbf{\Psi}^T\underline{\mathbf{u}} && (2.27)\\
&= \mathbf{\Phi\Sigma'\Psi}^T\underline{\mathbf{u}} && (2.28)\\
&= \mathbf{H}^\dagger\underline{\mathbf{u}} && (2.29)\\
&= \underline{\mathbf{x}}^\dagger. && (2.30)
\end{aligned}$$

This result [TA77] is of fundamental importance. Although the generalized inverse $\mathbf{H}^\dagger$ may be ill-posed or ill-conditioned if there are some eigenvlaues σ_m near zero, it is approximated by a nearby set of well-posed operators $\{(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}_{PQ\times PQ})^{-1}\mathbf{H}^T\}_\lambda$ as $\lambda \to 0$.

2.3.2 Generalized Tikhonov Regularization

One can generalize the Tikhonov approach by assuming

$$\underline{G}(\underline{\mathbf{x}}) = \underline{G_{Gen}}(\underline{\mathbf{x}}) = \lambda \| \mathbf{C}\underline{\mathbf{x}} - \mathbf{D} \|^2, \tag{2.31}$$

such that $\mathbf{C}^T\mathbf{C}$ is positive semidefinite. In this case, the approximation becomes

$$\underline{\tilde{\mathbf{x}}_{Gen,\lambda}} = (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{C}^T\mathbf{C})^{-1}(\mathbf{H}^T\underline{\mathbf{u}} + \lambda\mathbf{C}^T\mathbf{D}). \tag{2.32}$$

The positive semidefiniteness of $\mathbf{C}^T\mathbf{C}$ guarantees that $\mathbf{H}^T\mathbf{H} + \lambda\mathbf{C}^T\mathbf{C}$ is a non-singular matrix if $\lambda > 0$.

2.3.3 Regularization, only for Ill-posed Problems ?

Although the term $\underline{G_{Tikh}}(\underline{\mathbf{x}})$ added in the Tikhonov approach removed the ill-posedness of the original problem as an algebraic stabilizer, it is not clear at all whether it helps in getting a proper result. Furthermore, the above discussion might lead to the wrong impression that regularization is necessary only if the problem is ill-posed. Considering the image denoising problem, where the matrix $\mathbf{H} = \mathbf{I}_{PQ\times PQ}$, i.e., equals the identity operator, the matrix $\mathbf{H}^T\mathbf{H} = \mathbf{I}_{PQ\times PQ}$ is invertible and thus, the problem is well-posed. However, the generalized solution or the MNLS solution equals the observed image itself, i.e.,

$$\underline{\mathbf{x}}^\dagger = \underline{\tilde{\mathbf{x}}_{MNLS}} = (\mathbf{I}^T_{PQ\times PQ}\mathbf{I}_{PQ\times PQ})^{-1}\mathbf{I}^T_{PQ\times PQ}\underline{\mathbf{u}} = \underline{\mathbf{u}}. \tag{2.33}$$

This means that although the denoising problem is well-posed, the MNLS solution equals the noisy input image, and no denoising is performed on the image by approximating the MNLS solution. Hence, we have to point out that regularization not only acts as an *algebraic stabilizer* in estimating the solutions of ill-posed inverse problems (i.e., if $\mathbf{H}^T\mathbf{H}$ is singular), but it may be required to *improve* the solutions of well-posed problems in many situations.

2.4 Stochastic View

Again remember the problem of recovery of the unknown $\underline{\mathbf{x}}$ of size $PQ \times 1$, given the observation

$$\underline{\mathbf{u}} = \mathbf{H}\underline{\mathbf{x}} + \underline{\mathbf{n}}, \tag{2.34}$$

of size $MN \times 1$, and recall that $\mathbf{H}$ was a linear degradation matrix of size $MN \times PQ$. In the stochastic setting, we make the assumption that the noise vector $\underline{\mathbf{n}}$ is a random vector of size $MN \times 1$. Similarly, we assume that the vectors $\underline{\mathbf{x}}$ and $\underline{\mathbf{u}}$ are random vectors as well.

It is customary to believe that the value at each entry of the noise random vector $\underline{\mathbf{n}}$ is realized from a random variable with Gaussian distribution, zero mean and some known standard deviation σ. We also assume that this the random noise vector is independent of $\underline{\mathbf{x}}$ and is white, i.e., it is a zero mean vector and its autocorrelation matrix is a multiple of the identity matrix. By these

assumptions the Probability Density Function (PDF) of the random noise vector $\underline{\mathbf{n}}$ can be written as [AH77, Jai01]

$$P(\underline{\mathbf{n}}) = \frac{1}{(2\pi)^{\frac{MN}{2}}\sigma^{MN}} \exp\{-\frac{\underline{\mathbf{n}}\underline{\mathbf{n}}^T}{2\sigma^2}\}. \tag{2.35}$$

2.4.1 The Maximum-Likelihood Estimator (MLE)

The Maximum-Likelihood Estimator (MLE) suggests choosing $\underline{\mathbf{x}}$ that maximizes the conditional probability $P(\underline{\mathbf{u}} \mid \underline{\mathbf{x}})$, known as the *likelihood*. This is equivalent to choosing the unknown image $\underline{\mathbf{x}}$ that makes the measurements most likely to occur.

Considering the Gaussianity of the noise in Equation 2.35 and the fact that $\underline{\mathbf{x}}$ is assumed to be known, the measurement vector $\underline{\mathbf{u}} = \mathbf{H}\underline{\mathbf{x}} + \underline{\mathbf{n}}$ is also a Gaussian random vector with a mean that is shifted by the constant value $\mathbf{H}\underline{\mathbf{x}}$. Thus, the likelihood function becomes

$$P(\underline{\mathbf{u}} \mid \underline{\mathbf{x}}) = \frac{1}{(2\pi)^{\frac{MN}{2}}\sigma^{MN}} \exp\{-\frac{\|\underline{\mathbf{u}} - \mathbf{H}\underline{\mathbf{x}}\|^2}{2\sigma^2}\}. \tag{2.36}$$

Therefore, the MLE approximation is given by

$$\underline{\tilde{\mathbf{x}}}_{ML} = \arg\max_{\underline{\mathbf{x}}} P(\underline{\mathbf{u}} \mid \underline{\mathbf{x}}) = \arg\min_{\underline{\mathbf{x}}} \|\underline{\mathbf{u}} - \mathbf{H}\underline{\mathbf{x}}\|^2. \tag{2.37}$$

This is, however, the discrepancy functional that we minimized to find the pseudosolutions in the deterministic framework! Hence again if the Gram matrix $\mathbf{H}^T\mathbf{H}$ is invertible, there is a unique solution given by

$$\underline{\tilde{\mathbf{x}}}_{ML} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\underline{\mathbf{u}}, \tag{2.38}$$

which equals the LS solution $\underline{\tilde{\mathbf{x}}}_{LS}$ we found in the deterministic setting, i.e., $\underline{\tilde{\mathbf{x}}}_{ML} = \underline{\tilde{\mathbf{x}}}_{LS}$. Again in this case, if the Gram matrix is singular there are infinitely many possible solutions. Therefore, more information is required to tune the reconstruction toward a unique solution. This again leads naturally to the notion of regularization.

2.4.2 The Bayesian Approach and Regularization

The Bayesian approach [Bay64] starts with the replacement of the likelihood function with the *posterior* probability $P(\underline{\mathbf{x}} \mid \underline{\mathbf{u}})$. This minor change significantly changes the view of the approach. The Bayes rule [Bay64] relates the above two conditional probabilities, the posterior and likelihood, by

$$P(\underline{\mathbf{x}} \mid \underline{\mathbf{u}}) = \frac{P(\underline{\mathbf{u}} \mid \underline{\mathbf{x}}) \times P(\underline{\mathbf{x}})}{P(\underline{\mathbf{u}})}, \tag{2.39}$$

or equivalently in the same order

$$\textbf{Posterior} = \frac{\textbf{Likelihood} \times \textbf{Prior}}{P(\underline{\mathbf{u}})}. \tag{2.40}$$

Generally speaking, there are two ways to practise the Bayesian approach using the posterior in estimating $\underline{\mathbf{x}}$. These methods are known as (1) Maximum *A posteriori* Probability (MAP), and (2) Minimum Mean-Squared Error (MMSE).

Maximum *A-Posteriori* Probability (MAP) Estimation

The simpler method is the MAP method, choosing the $\underline{\mathbf{x}}$ that *maximizes* the posterior $P(\underline{\mathbf{x}} \mid \underline{\mathbf{u}})$. Using Equation 2.39, this reads

$$\begin{aligned} \underline{\tilde{\mathbf{x}}_{MAP}} &= \arg\max_{\underline{\mathbf{x}}} P(\underline{\mathbf{x}} \mid \underline{\mathbf{u}}) && (2.41)\\ &= \arg\max_{\underline{\mathbf{x}}} \frac{P(\underline{\mathbf{u}} \mid \underline{\mathbf{x}})P(\underline{\mathbf{x}})}{P(\underline{\mathbf{u}})} && (2.42)\\ &= \arg\max_{\underline{\mathbf{x}}} P(\underline{\mathbf{u}} \mid \underline{\mathbf{x}})P(\underline{\mathbf{x}}). && (2.43) \end{aligned}$$

Observe that the denominator $P(\underline{\mathbf{u}})$ has been removed from consideration, since it is considered as a constant with respect to the optimization objective. A convenient way to describe the PDF of $\underline{\mathbf{x}}$, known as the *prior*, is the *Gibbs distribution*, which represents $P(\underline{\mathbf{x}})$ in an exponential form [ZM97]

$$P(\underline{\mathbf{x}}) = \text{Const} \cdot \exp\{-\alpha \underline{G}(\underline{\mathbf{x}})\}. \tag{2.44}$$

Such a description loses no generality, as every non-negative function can be written in such a format. The constant in front of the exponential is a normalization factor, guaranteeing that the integral of prior over all $\underline{\mathbf{x}}$ is 1, and α is a positive parameter. The term $\underline{G}(\underline{\mathbf{x}})$ is a non-negative *energy function*, supposed to be low for highly probable signals and high otherwise. Using the Gibbs distribution expression in hand, and the expression derived for the likelihood function in Equation 2.36, we obtain

$$\begin{aligned} \underline{\tilde{\mathbf{x}}_{MAP}} &= \arg\min_{\underline{\mathbf{x}}} \|\underline{\mathbf{u}} - \mathbf{H}\underline{\mathbf{x}}\|^2 - \log\{P(\underline{\mathbf{x}})\} && (2.45)\\ &= \arg\min_{\underline{\mathbf{x}}} \|\underline{\mathbf{u}} - \mathbf{H}\underline{\mathbf{x}}\|^2 + 2\sigma^2 \alpha \underline{G}(\underline{\mathbf{x}}). && (2.46) \end{aligned}$$

The expression we just obtained is the same as what had earlier in Equation 2.15 by assuming that the parameter $\alpha = \frac{\lambda}{2\sigma^2}$ is a multiple of the regularization parameter λ in the deterministic setting. We see that the MAP method leads naturally to the concept of regularization explained in the deterministic setting, this time only giving a probabilistic meaning to the additional expression $\underline{G}(\underline{\mathbf{x}})$ rather than settling with the gained algebraic stability.

Minimum Mean Squared Error (MMSE) Estimation

A second way to exploit the posterior in a Bayesian approach is the MMSE estimation. This option chooses the *expected value* of $\underline{\mathbf{x}}$ based on its conditional density $P(\underline{\mathbf{x}} \mid \underline{\mathbf{u}})$, i.e.

$$\underline{\tilde{\mathbf{x}}_{MMSE}} = E(\underline{\mathbf{x}} \mid \underline{\mathbf{u}}) = \int_{\underline{\mathbf{x}}_1} \int_{\underline{\mathbf{x}}_2} \cdots \int_{\underline{\mathbf{x}}_{PQ}} \underline{\mathbf{x}} P(\underline{\mathbf{x}} \mid \underline{\mathbf{u}}) d\underline{\mathbf{x}} \tag{2.47}$$

Since the integral is PQ-dimensional (as the dimension of $\underline{\mathbf{x}}$), such an approach is typically prohibitive for non-scalar cases. Whichever method chosen, MMSE or MAP, the estimation of $\underline{\mathbf{x}}$ using the Bayesian approach requires a clear definition of the energy function $\underline{G}(\underline{\mathbf{x}})$. When dealing with images, this energy function is essentially describing how natural images behave.

In the next section, we describe the main choices made for $\underline{G}(\underline{\mathbf{x}})$ in the past few decades, showing the evolution of ideas on this matter [ED07].

2.5 Evolution of Image Priors

2.5.1 Priors from the "Good Old Days"

Assuming the Gibbs distribution for images, what should $\underline{G}(\underline{\mathbf{x}})$ be so as to reflect the distribution of natural images? This question poses one of the most fundamental problems in image processing. This enigma has drawn a considerable research attention in the past few decades, and is still considered an open question [ZM97, ED07, DE06]. In this section, we briefly describe the main milestones in this arena, showing how priors are getting "smarter" and more complex, all in the attempt to better describe image content. The Tikhonov regularization [TA77] presented earlier was among the first to be practised. Choosing $\underline{G}(\underline{\mathbf{x}}) = \|\mathbf{L}\underline{\mathbf{x}}\|^2$ in which $\mathbf{L}$ is the matrix associated to the Laplacian operator, promotes uniform spatial smoothness across the image. The uniformity is a key feature of this choice, as it leads to numerical convenience in the classic deblurring problem, where $\mathbf{H}$ is a linear space-invariant blurring operation. This becomes the well-known Wiener filter algorithm, which for many years was the leading approach in image deblurring [AH77].

By the late 1980s and early 1990s, it became clear that the Wiener filter does not produce sufficiently good results, and that better are within reach when the enforced spatial uniformity is avoided. This basic idea of forcing smoothness adaptively across the image found many manifestations in various proposed image priors [ED07]. One of the simplest ways was the weighted least-squares expression $\underline{G}(\underline{\mathbf{x}}) = (\mathbf{L}\underline{\mathbf{x}})^T\mathbf{W}(\mathbf{L}\underline{\mathbf{x}})$ in which $\mathbf{W}$ is a diagonal matrix with positive entries along the main diagonal being close to 1 for smooth regions, and close to 0 for edge or texture zones.

2.5.2 Total Variation Regularization and PDE-based Techniques

A vast amount of activity in image processing, which seems to be independent of the above discussion, is the introduction of PDE-based filtering techniques into image processing. Contributions such as the total variation (TV) by Rudin et al. [ROF92] and its variants to date, are considered among the best regularization techniques available and are often used in image processing. The edge-preserving regularization terms formed using various partial differential equations (PDE) methods [Wei98] are also among other techniques that lie in the same category.

Very briefly and in the simplest case of the TV-based regularization [ROF92], the functional $\underline{G}(\underline{\mathbf{x}})$ for any image $\mathbf{x}(x, y)$ is defined as

$$\underline{G}(\underline{\mathbf{x}}) = TV(\underline{\mathbf{x}}) = \int_{x,y} |\nabla\underline{\mathbf{x}}| dxdy. \tag{2.48}$$

It can be shown that

$$\nabla TV(\underline{\mathbf{x}}) = -\nabla \cdot \left(\frac{\nabla\underline{\mathbf{x}}}{|\nabla\underline{\mathbf{x}}|}\right), \tag{2.49}$$

which will be required in minimizing the corresponding functional.

2.5.3 Sparsity of Transform Domain Coefficients

In parallel to the impressive progress made on the use of PDEs in image processing in defining regularization expressions, the field of approximation theory contributed its own techniques for this purpose, in particular the use of the wavelet transform. Empirical observations suggested that wavelet transform coefficients of signals tend to be sparse, i.e., many of them are zero or near zero [DJ94]. This led to a proposed regularization expression of the form $\underline{G}(\underline{\mathbf{x}}) = \|\mathbf{DWT}(\underline{\mathbf{x}})\|_p^p$, where **DWT** is the discrete wavelet transform operator in matrix form, and the p-norm (with $p \geq 1$) sums over these coefficients in a way that promotes sparsity [Mal98].

More recent signal recovery techniques, e.g., basis pursuit [CDS98] and some of the methods in the newly developed area of compressed sensing, e.g. [CR07], consider priors in terms of the ℓ^1 norm, e.g., $\underline{G}(\underline{\mathbf{x}}) = \|\underline{\mathbf{x}}\|_1$. Although this gives *non-smooth* objective functions in the corresponding minimization schemes, interior point methods or convex programming techniques are typically employed for a computationally stable and efficient recovery of the image.

2.5.4 Example-based Regularization

The quest for better regularization expressions for images is very much active today, with many new contributions that extend the options and improve on them. Using the above rationale in forming regularization, one must question the fundamental ability of a simple analytical expression $\underline{G}(\underline{\mathbf{x}})$ to grasp the complexity and wealth of general image content [ED07, DE06].

An emerging powerful regularization methodology that has been drawing research attention in recent years is regularization based on image examples. Rather than guessing the image PDF and forcing a simple expression to be used to describe it, we let example images guide us in the construction of the prior. Examples can be used in a variety of ways, and the various proposed methods can be roughly divided into three categories extensively described in [ED07, DE06]: (1) Learning prior parameters: If we are generally pleased with the above-described analytical priors, those can be further improved by learning their parameters [CPT04, EL99, FPC00, FJP02, NK03, WL00]. (2) Learning the posterior directly: Rather than learn the image prior and then plug it in a MAP/MMSE reconstruction penalty term, one can use the examples to directly learn the posterior PDF, and then use it for the reconstruction [HT03, RB05, ZM97, BCM05b, BCM05c]. (3) Building a regularization expression with examples: This is a fusion of the above two techniques, where examples are found as part of the reconstruction process, and then inserted directly into an explicit regularization expression [BK02, DE06, ED07, FPC00, FJP02].

Considering the vast progress made on the formation of regularization expressions, we will be exploiting some of these methods in this thesis.

2.6 Iterative Methods in Inverse Problems

In the previous sections, we introduced various abstract methods for solving ill-posed problems. The practical implementation of these requires the minimization of some functionals or perhaps

inverting a large matrix.

If the functional is non-quadratic, nonlinear iterative methods such as the *steepest descent method* or variants of Newton's method can be used to compute the regularized solution. If the functional is quadratic, its minimizer satisfies a system of linear equations. If this system is large, iterative methods, e.g., the *conjugate gradient method*, may be required to efficiently compute the minimizer [Vog02]. In this section, we briefly review the methods of steepest descent and conjugate gradient. However, before we continue, we introduce the traditional *Method of Projection onto Convex Sets (POCS)*, and *Projected Landweber Iterations* which are important *set-based* approaches in inverse problems. POCS and Projected Landweber methods implicitly find an approximate solution of inverse problems based on the notion of *sets* and *projections*.

We will be using some of these methods in the computational results presented in the following chapters of this thesis.

2.6.1 Method of Projection onto Convex Sets (POCS)

In this section, we consider the image restoration problem by defining a set of constraints which must be satisfied by candidate solutions [YW82]. In the POCS formulation, constraint sets are used to define the feasible solution space for the restoration. These constraints are defined as convex and closed sets that represent desirable characteristics of the solution. These sets encapsulate properties such as similarity to the observed data, positivity, bounded energy, smoothness and so on. The solution space of the restoration problem is, by design, the intersection of the convex constraint sets. POCS is an iterative procedure which, given any element in the corresponding space, locates an element which simultaneously satisfies all the constraint sets.

Assume that $\mathbf{x} \in \ell^2(\Upsilon)$ is known to lie in m given sets Ψ_i, $i = 1, 2, \ldots, m$ where each of the sets represents a constraint on the image. If the sets Ψ_i are closed and convex, we associate projection operators $\mathcal{P}_i$, $i = 1, 2, \ldots, m$, to each Ψ_i. The projection of $\mathbf{h}$ onto Ψ_i is defined as $\mathbf{p} = \mathcal{P}_i\mathbf{h}$, with $\mathbf{p} \in \Psi_i$ and

$$\|\mathbf{p} - \mathbf{h}\| = \inf_{\mathbf{y}\in\Psi_i} \|\mathbf{y} - \mathbf{h}\|. \tag{2.50}$$

The closedness and convexity of the Ψ_i's guarantee that this projection is well-defined and uniquely exists. Youla et al.[YW82] proved that if the sets Ψ_i, $i = 1, 2, \ldots, m$ are closed and convex and $\bigcap_i \Psi_i \neq \emptyset$, then the sequence $\{\mathbf{x}_n\}$ defined recursively as

$$\mathbf{x}_{n+1} = \mathcal{P}_m \ldots \mathcal{P}_2\mathcal{P}_1\mathbf{x}_n, \tag{2.51}$$

converges to a fixed point in the intersection of all Ψ_is. In other words, $\mathbf{x}_n$ converges to a point $\tilde{\mathbf{x}}_{POCS}$, where $\tilde{\mathbf{x}}_{POCS} \in \bigcap_i \Psi_i$ (see Figure 2.1).

In [YW82], a detailed theoretical discussion of the POCS method can be found. It is imperative to note that the limit point $\tilde{\mathbf{x}}_{POCS}$ is *non-unique*, in general. It is only known that the recovered image lies in the intersection of the constraint sets. Furthermore, the limiting point $\tilde{\mathbf{x}}_{POCS}$ is also dependent on the initial guess in general.

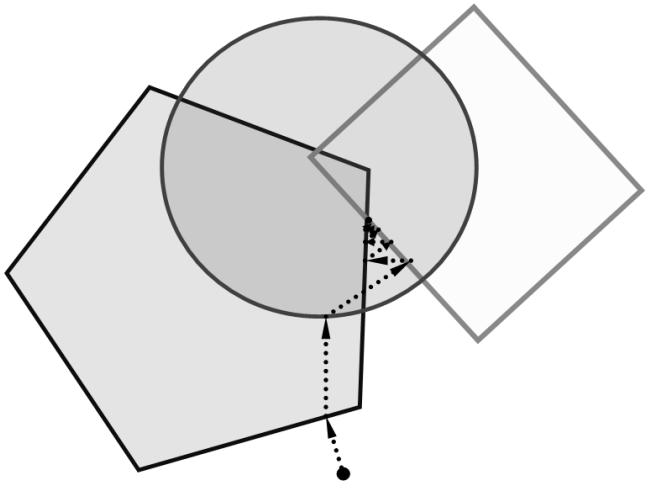


Figure 2.1: A schematic example of POCS.

POCS attracted very much attention in a multitude of image restoration and reconstruction applications over the years. The main reasons for the attention were the simplicity, flexibility, and power in the inclusion of *a priori* information. It is generally simple to define new convex constraint sets which incorporate desired solution characteristics. These sets may impose restrictions such as positivity or bounded energy which are difficult to represent in terms of cost functions. The only potential source of difficulty in applying POCS is to determine the projection operators. If the convex and closed sets are constructed and the projection operators are found, the iterations can be easily implemented. We will be using this idea later with regard to fractal image coding.

2.6.2 Projected Landweber Iterations

Assume that Ψ_c is a closed, convex subset of $\ell^2(\Upsilon)$ and consider the minimization problem

$$\underline{\tilde{\mathbf{x}}_{Land,\Psi_c}} = \arg \min_{\underline{\mathbf{x}} \in \Psi_c} \|\mathbf{H}\underline{\mathbf{x}} - \underline{\mathbf{u}}\|^2. \tag{2.52}$$

Furthermore, assume that this minimization problem has a non-empty set of solutions. Therefore, some (not necessarily unique minimizer) $\underline{\tilde{\mathbf{x}}_{Land,\Psi_c}}$ exists. Also, assume that $\mathcal{P}_{\Psi_c}$ denotes the projection operator on Ψ_c. The Projected Landweber Algorithm, described in the following iterative scheme, evaluates some unique minimizer $\underline{\tilde{\mathbf{x}}_{Land,\Psi_c}}$. This situation is sometimes called as *hard constrained* case, because the solution is assumed to lie strictly in the set Ψ_c. This case along with the POCS scheme covered in the previous section can be thought in the category of "set-based regularization" frameworks. A very interesting unified formulation of such regularization schemes has recently been given in [CW05].

Algorithm 2.6.1. ***Projected Landweber Iterations***
To find a minimizer $\underline{\tilde{\mathbf{x}}_{Land,\Psi_c}}$,

Choose some fixed step-size $\mu > 0$ *and a threshold* $\epsilon > 0$;
$n := 0$;
$\underline{\mathbf{x}_0}$:=*initial guess;*
begin projected Landweber iterations

$$\begin{aligned} \underline{\mathbf{x}_{n+1}} &:= \mathcal{P}_{\Psi_c}\big(\underline{\mathbf{x}_n} - \mu\mathbf{H}^T(\mathbf{H}\underline{\mathbf{x}_n} - \underline{\mathbf{u}})\big); \qquad \%\ \textit{update approximate solution} \\ n &:= n+1; \end{aligned}$$

end projected Landweber iterations if $\|\underline{\mathbf{x}_n} - \underline{\mathbf{x}_{n-1}}\| < \epsilon$.

Note that choosing a step-size, $0 < \mu < \frac{1}{\|\mathbf{H}\|^2_{2-norm}}$, where $\|\mathbf{H}\|_{2-norm}$ is the matrix 2-norm of $\mathbf{H}$, guarantees the convergence of this algorithm.

2.6.3 Steepest Descent

Steepest descent or *gradient descent* is a simple minimization algorithm for a real-valued differentiable objective function F. The algorithm is based on the observation that if the real-valued function F is defined and differentiable in a neighbourhood of a point $\mathbf{a}$, then it decreases most rapidly in the direction of the negative gradient $\mathbf{p} = -\nabla F(\mathbf{a})$. Here we summarize the algorithm as an outcome of this observation.

Algorithm 2.6.2. ***Steepest Descent with Variable Step-Size***
To minimize a given smooth function $F(\mathbf{x})$,

Choose some threshold $\epsilon > 0$;
$n := 0$;
$\mathbf{x}_0$:=*initial guess;*
begin steepest descent iterations

$$\begin{aligned} \mathbf{p}_n &:= -\nabla F(\mathbf{x}_n) && \%\ \textit{update search direction} \\ \mu_n &:= \arg\min_{\mu>0} F(\mathbf{x}_n + \mu\mathbf{p}_n); && \%\ \textit{line search step-size} \\ \mathbf{x}_{n+1} &:= \mathbf{x}_n + \mu_n\mathbf{p}_n; && \%\ \textit{update approximate solution} \\ n &:= n+1; \end{aligned}$$

end steepest descent iterations if $\|\mathbf{x}_n - \mathbf{x}_{n-1}\| < \epsilon$.

The above algorithm relies on finding a line search step-size which may be more complicated to solve than the original minimization algorithm [Vog02]. An alternative strategy is to modify the algorithm by choosing an appropriate *fixed step-size* and hope that it will converge to an appropriate solution.

Algorithm 2.6.3. ***Steepest Descent with Fixed Step-Size***
To minimize a given smooth function $F(\mathbf{x})$*,*

Choose some fixed step-size $\mu > 0$ *and a threshold* $\epsilon > 0$;
$n := 0$*;*
$\mathbf{x}_0$:=*initial guess;*
begin steepest descent iterations

$$\begin{aligned} \mathbf{x}_{n+1} &:= \mathbf{x}_n - \mu \nabla F(\mathbf{x}_n); \qquad \% \text{ update approximate solution} \\ n &:= n+1; \end{aligned}$$

end steepest descent iterations if $\|\mathbf{x}_n - \mathbf{x}_{n-1}\| < \epsilon$*.*

Note that choosing a very large step-size μ will affect the convergence of the algorithm to the appropriate solution. On the other hand, if μ is very small the rate of convergence is very slow.

2.6.4 Conjugate Gradient (CG)

For the extremely large linear systems direct solution methods are not practical. On the other hand, the slow convergence rate of the steepest descent iteration limits its usefulness [Vog02]. In this case, an iterative technique known as *Conjugate Gradient (CG)* method provides a very efficient framework for solving symmetric positive definite linear systems which are of large dimension

Algorithm 2.6.4. ***The Conjugate Gradient Method***
To solve the linear equation $\mathbf{A}\underline{\mathbf{x}} = \underline{\mathbf{b}}$*, in which* $\mathbf{A}$ *is symmetric positive definite,*
Choose some threshold $\epsilon > 0$;
$n := 0$*;*
$\underline{\mathbf{x}_0}$:=*initial guess;*
$\underline{\mathbf{r}_0} := \mathbf{A}\underline{\mathbf{x}_0} - \underline{\mathbf{b}}$*;* *% initial gradient*
$\underline{\mathbf{p}_0} := -\underline{\mathbf{r}_0}$*;* *% initial search direction*
$\epsilon_0 := \|\underline{\mathbf{r}_0}\|^2$;
begin conjugate gradient iterations

$$\begin{aligned} \underline{\mathbf{h}_n} &:= \mathbf{A}\underline{\mathbf{p}_n}; \\ \mu_n &:= \Big(\frac{\epsilon_n}{\underline{\mathbf{p}_n}^T \underline{\mathbf{h}_n}}\Big); \qquad \% \text{ line search step-size} \\ \underline{\mathbf{x}_{n+1}} &:= \underline{\mathbf{x}_n} + \mu_n \underline{\mathbf{p}_n}; \qquad \% \text{ update approximate solution} \\ \underline{\mathbf{r}_{n+1}} &:= \underline{\mathbf{r}_n} + \mu_n \underline{\mathbf{h}_n}; \qquad \% \text{ update gradient} \\ \epsilon_{n+1} &:= \|\underline{\mathbf{r}_{n+1}}\|^2; \\ \underline{\mathbf{p}_{n+1}} &:= -\underline{\mathbf{r}_{n+1}} + \Big(\frac{\epsilon_{n+1}}{\epsilon_n}\Big)\underline{\mathbf{p}_n}; \qquad \% \text{ update search direction} \\ n &:= n+1; \end{aligned}$$

end conjugate gradient iterations if $\epsilon_n < \epsilon$*.*

Chapter 3

Self-similarity in Imaging Inverse Problems

3.1 Introduction

Since the appearance of B. Mandelbrot's classic work, *The Fractal Geometry of Nature* [Man83], the idea of self-similarity has played a very important role in mathematics and physics. In the late-1980's, M. Barnsley of Georgia Tech, with coworkers and students, showed that sets of contractive maps with associated probabilities, called *Iterated Function Systems (IFS)*, could be used not only to generate fractal sets and measures but also to approximate natural objects and images [Bar88]. This gave birth to *fractal image compression*, which would become a hotbed of research activity over the next decade. Indeed, twenty years have passed since *Fractals Everywhere* [Bar88], Barnsley's beautiful exposition of IFS theory and its applications, was first published.

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

In this chapter, we outline the evolution of the idea of self-similarity from Mandelbrot's "generators" to Iterated Function Systems and fractal image coding. Furthermore, a detailed explanation of the fractal image coding concept will be given along with a review of its applications other than image compression, e.g., denoising and zooming. Finally, we introduce some newly developed nonlocal image denoising filters [BCM05b, BCM05c] inspired by the self-similarity properties of images.

3.2 History: From Self-similarity to Fractal Image Coding

In *The Fractal Geometry of Nature* [Man83], B. Mandelbrot showed how "fractal" sets could be viewed as limits of iterative schemes involving *generators*. In the simplest case, a generator $\mathcal{G}$ acts on a set $\mathscr{U}$ (for example, a line segment) to produce N affinely-contracted copies of $\mathscr{U}$ and then arranges these copies in space according to a prescribed rule. Starting with an appropriate "initiator set" $\mathscr{U}_0$, the iteration procedure $\mathscr{U}_{n+1} = \mathcal{G}(\mathscr{U}_n)$ converges, in the limit $n \to \infty$, to a fractal set $\mathscr{U}^\star$.

Last, but certainly not least, the set $\mathscr{U}^\star$ is "self-similar," meaning that arbitrarily small pieces of $\mathscr{U}^\star$ are scaled-down copies of $\mathscr{U}^\star$. Consequently, $\mathscr{U}^\star$ can be expressed as a union of contracted copies of itself.

Some of Mandelbrot's examples were certainly not new. For example, in the classical construction of the ternary Cantor set, the usual "middle-thirds" dissection procedure is represented by a generator $\mathcal{G}$ that, acting on a line segment of length l, produces two contracted copies of length $l/3$ which are separated by a distance $l/3$. From this, the Cantor set is viewed as a union of two

contracted copies of itself. (And, in turn, we may obtain its *fractal dimension* to be $\ln 2/\ln 3$.) These simple examples, however, led to more general constructions, including random fractals. The result was a rather unified treatment of fractal geometry.

The next leap in fractal analysis and construction came with the work of J. Hutchinson [Hut81]. One of the main ingredients was a set of N contractive maps $w_i : \mathfrak{X} \to \mathfrak{X}$, $1 \leq i \leq N$ on a complete metric space $(\mathfrak{X}, d)$. Associated with these maps was a set-valued mapping $\mathcal{W}$, defined as the union of the set-valued actions of the w_i. (As such, $\mathcal{W}$ performed the shrink-and-placement procedure of Mandelbrot's generator $\mathcal{G}$.) Hutchinson showed that $\mathcal{W}$ is contractive in $(\mathscr{H}(\mathfrak{X}), h)$, the complete metric space of non-empty compact subsets of $\mathfrak{X}$ with Hausdorff metric h. From Banach's *fixed point theorem* (also known as the contraction mapping theorem or contraction mapping principle), there exists a unique set $\mathscr{U}^\star \in \mathscr{H}(\mathfrak{X})$, the *attractor* of the IFS $\mathcal{W}$, which satisfies the fixed point relation

$$\mathscr{U}^\star = \mathcal{W}(\mathscr{U}^\star) = \bigcup_{i=1}^{N} w_i(\mathscr{U}^\star). \tag{3.1}$$

$\mathscr{U}^\star$ is self-similar since it can be expressed as a union of copies of itself.

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

$$\mu^\star(\mathscr{X}) = \mathcal{M}\mu^\star(\mathscr{X}) = \sum_{i=1}^{N} p_i \mu^\star(w_i^{-1}(\mathscr{X})), \quad \forall \mathscr{X} \in \mathscr{H}(\mathfrak{X}). \tag{3.2}$$

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

M. Barnsley and S. Demko [BD85] independently discovered the use of such systems of mappings and associated probabilities for the construction of fractal sets and measures, coining the term *Iterated Function Systems (IFS)*. Their analysis was framed in a more probabilistic setting and gave rise to the well-known *chaos game algorithm* for generating pictures of attractors and invariant measures.

In the Barnsley/Demko paper was the first suggestion that IFS might be useful for the *approximation of natural objects*. This was the seed for the *inverse problem of fractal-based approximation*: Given a "target" set $\mathscr{U}$, for example, a leaf, can one find an IFS with attractor $\mathscr{U}^\star$ that approximates $\mathscr{U}$ to some desired degree of accuracy?

Subsequently, Barnsley and students [BEHL85] showed how the inverse problem could be reformulated/simplified in terms of the following *Collage Theorem*, a simple consequence of Banach's fixed point theorem: Given a contraction map $\mathcal{T} : \mathfrak{X} \to \mathfrak{X}$ with contraction factor $c \in [0, 1)$ and fixed point $\mathscr{U}^\star$, then

$$d(\mathscr{U}, \mathscr{U}^\star) \leq \frac{1}{1-c} d(\mathscr{U}, \mathcal{T}\mathscr{U}), \qquad \forall \mathscr{U} \in \mathfrak{X}. \tag{3.3}$$

If $\mathscr{U} \in \mathfrak{X}$ is a "target" (set or measure) that we wish to approximate with the fixed point $\mathscr{U}^\star$ of a contraction map $\mathcal{T}$, then instead of searching for a $\mathcal{T}$ (i.e., an IFS) that makes the approximation error $d(\mathscr{U}, \mathscr{U}^\star)$ small, one looks for a $\mathcal{T}$ such that the *collage distance* $d(\mathscr{U}, \mathcal{T}\mathscr{U})$ is small – in other words a transform $\mathcal{T}$ that maps $\mathscr{U}$ as close as possible to itself. For IFS on sets, cf. Equation 3.1, this amounts to looking for ways in which a set $\mathscr{U}$ can be approximated by a union of contracted and distorted copies of itself. This procedure was illustrated very nicely for a generic leaf-shape in [BEHL85].

In [BEHL85] was also presented the infamous "Barnsley fern" – the attractor of a four-map *IFS with probabilities (IFSP)* shown in Figure 3.1. But this "fern" was more than an IFS attractor set $\mathscr{U}^\star$ – it was an invariant measure μ which could be represented on a computer screen as a shaded image. In other words, the construction of *sets* by IFS now becomes the construction of *images* by IFSP.

Figure 3.1: The infamous "Barnsley fern" IFS.

Naturally, the next question was: "Can IFS(P) be used to approximate other natural objects?" But an even more ambitious question was: "Can IFS(P) be used to approximate images?" There were various attempts, and Barnsley himself announced success to the world in the January 1988 issue of *BYTE* magazine [BS88], claiming that astronomical rates of data compression could be achieved with IFS image coding. However, the details of the IFS compression method were not revealed at that time for proprietary reasons. (Barnsley had been granted a software patent and subsequently founded Iterated Systems Incorporated.)

The thesis of A. Jacquin [Jac89], one of Barnsley's Ph.D. students, however, removed much of the mystery behind IFS image coding. Indeed, Jacquin's seminal paper [Jac92] described in sufficient detail the method of *block-based fractal image coding* which is still the basis of most, if not all, fractal-based image coding methods. It is, of course, overly ambitious to expect that an image

can be well approximated with smaller parts of itself. Therefore, in the spirit of standard block-based image coding methods (e.g., Discrete Cosine Transform (DCT), Vector Quantization (VQ)), *range* or *child* subblocks of an image were approximated by affine greyscale transformations of decimated versions of larger *domain* or *parent* subblocks. Although Jacquin's original formulation was in terms of measures, the method is easily interpreted in terms of image functions. Such interpretations will be reviewed in the following section.

3.3 Block-based Fractal Image Coding

More details on fractal image coding can be found in many places [Bar88, BH93, Fis95, Lu97, Fis98, GFV03]. In this section, we outline the most important features of fractal image coding. Fractal image coding seeks to approximate an image by a union of spatially-contracted and greyscale-modified copies of subblocks of itself.

3.3.1 Fractal Image Encoding

Let an image of interest be represented by an image function $\mathbf{u}(x, y)$, denoted by $\mathbf{u} \in \ell^2(\Upsilon)$, where

$$\Upsilon = [1, \ldots, N] \times [1, \ldots, N]. \tag{3.4}$$

Briefly, the result of the coding procedure is a contractive mapping T, the so-called *fractal transform* operator. The fixed point $\mathbf{u}^\star$ of T provides an approximation to $\mathbf{u}$. In other words,

$$\mathbf{u} \cong \mathbf{u}^\star = T(\mathbf{u}^\star). \tag{3.5}$$

We need to take a few steps to define the operator T and study its properties.

Partitioning

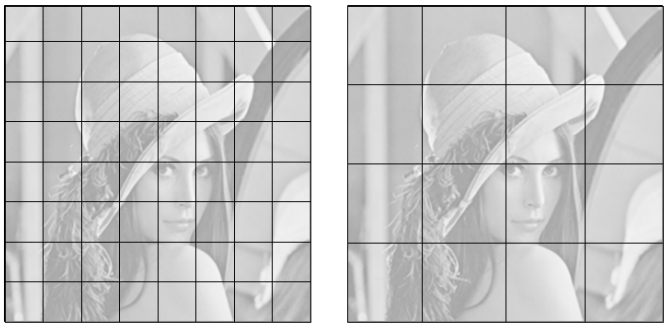
First, consider a partition of Υ into nonoverlapping subblocks $\{C_i\}$, $i \in \mathfrak{C}$, referred to as *range (or child) subblocks*, such that

$$\Upsilon = \bigcup_{i \in \mathfrak{C}} C_i. \tag{3.6}$$

Also, consider a new partition of Υ by typically larger subblocks $\{P_j\}$, $j \in \mathfrak{P}$, referred to as *domain (or parent) subblocks*, such that

$$\Upsilon = \bigcup_{j \in \mathfrak{P}} P_j. \tag{3.7}$$

In this thesis, we assume that the elements $\{P_j\}$ do not mutually overlap, however this is not always a requirement in many texts. For example, we may assume that the square blocks C_i and $P_{J(i)}$ are, respectively of size $K \times K$ and $sK \times sK$, where each of these blocks belongs to the corresponding square grid partitions of Υ, for positive integers K and s. In this specific case, we need to assume sK divides N to guarantee the existence of these grids (see Figure 3.2).



(a) (b)

Figure 3.2: Partitioning of the an image into range and domain blocks, respectively in (a) and (b).

Block and Parameter Search

For each $i \in \mathfrak{C}$, associated with each range (or child) subblock C_i one searches for the index $J(i) \in \mathfrak{P}$ of a corresponding domain (or parent) subblock $P_{J(i)}$ in a manner that $\mathbf{u}(C_i) = \mathbf{u}\Big|_{C_i}$ is well approximated by a spatially-contracted and greyscale-modified copy of $\mathbf{u}(P_{J(i)}) = \mathbf{u}\Big|_{P_{J(i)}}$, i.e.,

$$\mathbf{u}\Big|_{C_i} \cong \phi_i\Big(\mathcal{D}\Big(\mathbf{u}\Big|_{P_{J(i)}}\Big)\Big). \tag{3.8}$$

Here, $\phi_i : \mathbb{R} \to \mathbb{R}$ are *greyscale maps* that operate on pixel intensities and are usually assumed to be affine, i.e., $\phi_i(t) = \alpha_i t + \beta_i$. Also, $\mathcal{D}$ is the decimation operator by a factor of s introduced in Chapter 1.

Therefore, for each $i \in \mathfrak{C}$ one finds $J(i)$, α_i, and β_i that minimize the following expression

$$(J(i), \alpha_i, \beta_i) = \arg \min_{j \in \mathfrak{P}, (\alpha,\beta) \in \Pi} \left\| \mathbf{u}\Big|_{C_i} - \Big\{\alpha \mathcal{D}\Big(\mathbf{u}\Big|_{P_j}\Big) + \beta\Big\} \right\|. \tag{3.9}$$

In this expression, $\Pi \subset \mathbb{R}^2$ denotes the feasible (α, β) parameter space, which will be suitably restricted. Note that such a minimization problem may have non-unique solutions. The solution of the above minimization problem is performed by exhaustive searching over all $j \in \mathfrak{P}$ for each $i \in \mathcal{C}$. For a domain-range block pair P_j/C_i, the optimal value of the α and β parameters may depend on the parameter space Π. Typically this may be accomplished by means of least-squares or constrained least squares method. If $\Pi = \mathbb{R}^2$, i.e., when no constraint is assumed on the parameters α and β and the pair of blocks $\mathbf{u}\Big|_{C_i}$, $\mathcal{D}\Big(\mathbf{u}\Big|_{P_j}\Big)$ are respectively represented by vectors $\underline{\mathbf{u}}_{c_i}$ and $\underline{\mathbf{u}}_{p_j}$ of the same size, it is easy to show that (e.g. see [Gha04]) the minimizing parameters of the above expression for some fixed $i \in \mathfrak{C}$ and $j \in \mathfrak{P}$ is given by

$$\alpha_i = \frac{Cov(\underline{\mathbf{u}}_{p_j}, \underline{\mathbf{u}}_{c_i})}{Var(\underline{\mathbf{u}}_{p_j})} \tag{3.10}$$

$$\beta_i = E(\underline{\mathbf{u}}_{c_i}) - \alpha_i E(\underline{\mathbf{u}}_{p_j}), \tag{3.11}$$

in the case that $Var(\underline{\mathbf{u}}_{p_j}) \neq 0$. Furthermore, in the case that $Var(\underline{\mathbf{u}}_{p_j}) = 0$ and the elements of $\mathbf{u}_{p_j}$ are all non-negative,

$$\alpha_i = 0 \tag{3.12}$$

$$\beta_i = E(\underline{\mathbf{u}}_{c_i}) \tag{3.13}$$

are some non-unique minimizing parameters.

Fractal Transform Operator

We can summarize that the *fractal code* of the image $\mathbf{u}$ approximated in the outlined fashion consists of

$$\{J(i), \alpha_i, \beta_i\}, \tag{3.14}$$

for all $i \in \mathfrak{C}$.

To introduce the *fractal transform operator* using the obtained fractal code, it is required to define one more ingredient, the so-called *block mapping*. For each $i \in \mathfrak{C}$, define a *block mapping* w_i, from $P_{J(i)}$ to C_i such that

$$C_i = w_i(P_{J(i)}). \tag{3.15}$$

In our discrete case, if C_i is a block of size $K \times K$, and $P_{J(i)}$ is of size $sK \times sK$ then w_i relates every $s \times s$ block of $P_{J(i)}$ to the corresponding 1×1 pixel in C_i. More precisely, for any $i \in \mathfrak{C}$ and $1 \leq p_x, p_y \leq sK$,

$$w_i(P_{J(i)}(p_x, p_y)) = C_i(\lceil \frac{p_x}{s} \rceil, \lceil \frac{p_y}{s} \rceil). \tag{3.16}$$

This is equivalent to state that for any $i \in \mathfrak{C}$, $(x, y) \in \Upsilon$, and $1 \leq c_x, c_y \leq s$,

$$w_i(P_{J(i)}(s(x-1) + c_x, s(y-1) + c_y)) = C_i(x, y). \tag{3.17}$$

Such a w_i is not 1-1 and by notations $w_i^{-1}(C_i)$ and $w_i^{-1}(x, y)$ for some $(x, y) \in C_i$, we respectively mean the inverse images of C_i and (x, y) under w. The former would simply be $P_{J(i)}$ and the latter is an $s \times s$ block in the corresponding $P_{J(i)}$. Hence, using the newly defined block mapping notion we can write,

$$\mathbf{u}\Big|_{C_i} \cong \phi_i\Big(\mathcal{D}\Big(\mathbf{u}\Big|_{P_{J(i)}}\Big)\Big) = \phi_i\Big(\mathcal{D}\Big(\mathbf{u}\Big|_{w_i^{-1}(C_i)}\Big)\Big) \tag{3.18}$$

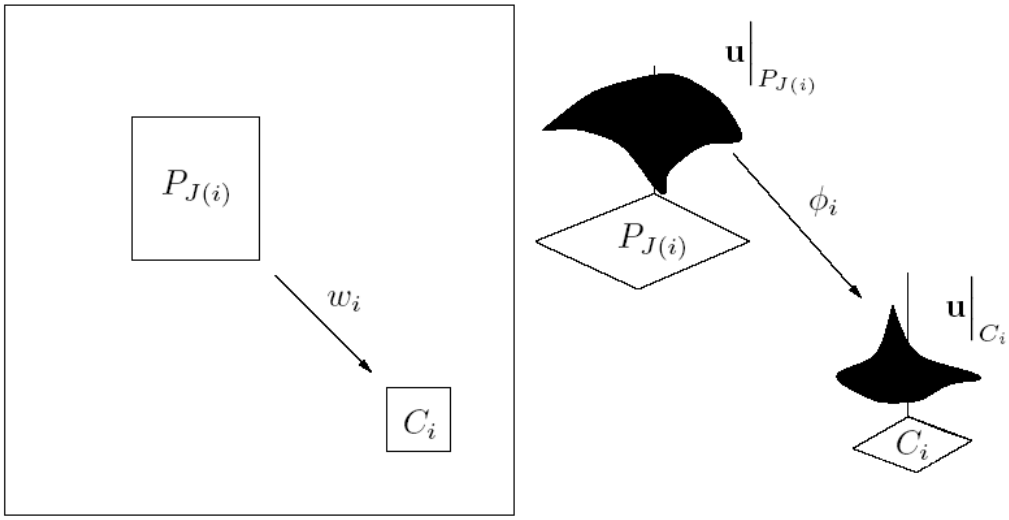
(see Figure 3.3).

Let us now assume that we have computed the fractal code of an image function $\mathbf{u}$ according to Equation 3.9. Because of the nonoverlapping nature of the partition by the range subblocks C_i's, we may define T,

$$\mathbf{u}(x, y) \cong (T(\mathbf{u}))(x, y) = \sum_{i \in \mathfrak{C}} \phi_i\Big(\mathcal{D}\big(\mathbf{u}(w_i^{-1}(x, y))\big)\Big), \tag{3.19}$$

for every $(x, y) \in \Upsilon$. The image function $\mathbf{u}$ is thus approximated as a sum of spatially-contracted and greyscale-distorted (ϕ_i) copies of its blocks. This T will be referred to as the *fractal transform operator* of $\mathbf{u}$.

Under suitable conditions outlined (and further extended by the author) in the next chapter, the operator T is contractive in $\ell^2(\Upsilon)$ [BEHL85]. As such, using Banach's contraction mapping principle, there will exist a unique fixed point $\mathbf{u}^\star \in \ell^2(\Upsilon)$ such that $T(\mathbf{u}^\star) = \mathbf{u}^\star$.



$$\mathbf{u}\Big|_{C_i} \cong \phi_i\Big(\mathcal{D}\Big(\mathbf{u}\Big|_{P_{J(i)}}\Big)\Big) = \phi_i\Big(\mathcal{D}\Big(\mathbf{u}\Big|_{w_i^{-1}(C_i)}\Big)\Big)$$

Figure 3.3: Fractal coding, mapping of a domain block to a range block.

Collage Theorem

If the above approximation is a "good one", then the so-called *collage distance* $\| \mathbf{u} - T(\mathbf{u}) \|$ is small. From the so-called "Collage Theorem" [BEHL85],

$$\| \mathbf{u} - \mathbf{u}^\star \| \leq \frac{1}{1 - con_{(T)}} \| \mathbf{u} - T(\mathbf{u}) \|, \tag{3.20}$$

it then follows that if $\mathbf{u}$ is "close" to $T(\mathbf{u})$, then $\mathbf{u}$ is also close to $\mathbf{u}^\star$, the fixed point of T. Here, $con_{(T)} \in [0, 1)$ denotes the contraction factor of T. The quantity $\| \mathbf{u} - \mathbf{u}^\star \|$ is the error of approximation of $\mathbf{u}$ by $\mathbf{u}^\star$.

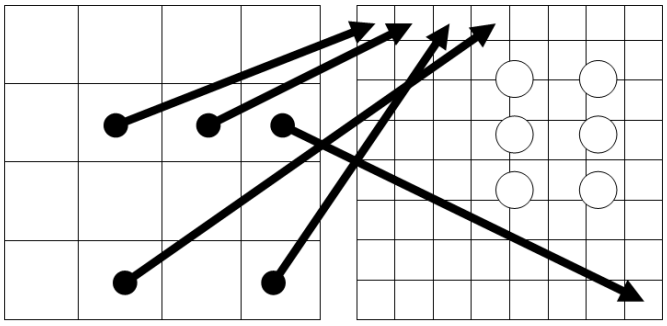
3.3.2 Fractal Image Decoding

Once we have a fractal transform T, we may generate its fixed point $\mathbf{u}^\star$ by a simple iteration. Starting with an arbitrary image $\mathbf{u}_0$, one forms the iteration sequence $\mathbf{u}_{n+1} = T(\mathbf{u}_n)$. In this decoding procedure, the image subblocks $\mathbf{u}_n(C_i)$ of $\mathbf{u}_n$ are replaced by modified copies $\phi_i\Big(\mathcal{D}\big(\mathbf{u}_n(P_{J(i)})\big)\Big)$ according to Equation 3.18 (see Figure 3.4). This yields,

$$\mathbf{u}_{n+1}\Big|_{C_i} = \Big(T(\mathbf{u}_n)\Big)\Big|_{C_i} = \alpha_i \mathcal{D}\Big(\mathbf{u}_n\Big|_{P_{J(i)}}\Big) + \beta_i. \tag{3.21}$$

starting with an arbitrary image $\mathbf{u}_0$.

Banach's contraction mapping theorem guarantees that the sequence of images $\mathbf{u}_n$ converges to $\mathbf{u}^\star$, if T is a contraction.



$$\mathbf{u}_{n+1}\Big|_{C_i} = \Big(T(\mathbf{u}_n)\Big)\Big|_{C_i} = \alpha_i \mathcal{D}\Big(\mathbf{u}_n\Big|_{P_{J(i)}}\Big) + \beta_i.$$

Figure 3.4: One iteration of fractal decoding.

3.3.3 Example: Fractal Coding Applied to an Image

In Figure 3.5 is presented the fixed point approximation $\mathbf{u}^\star$ to the standard 512×512 pixel Lena image (8 bits/pixel) using a partition of 8×8 nonoverlapping pixel blocks ($64^2 = 4096$ in total). The domain pool was the set of $32^2 = 1024$, 16×16 non-overlapping pixel blocks. (This is clearly not optimal.) This image was obtained by starting with the seed image $\mathbf{u}_0(x) = 255$ (plain white image) and iterating $\mathbf{u}_{n+1} = T(\mathbf{u}_n)$ to $n = 15$. Iterates $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$ are also shown in this figure.

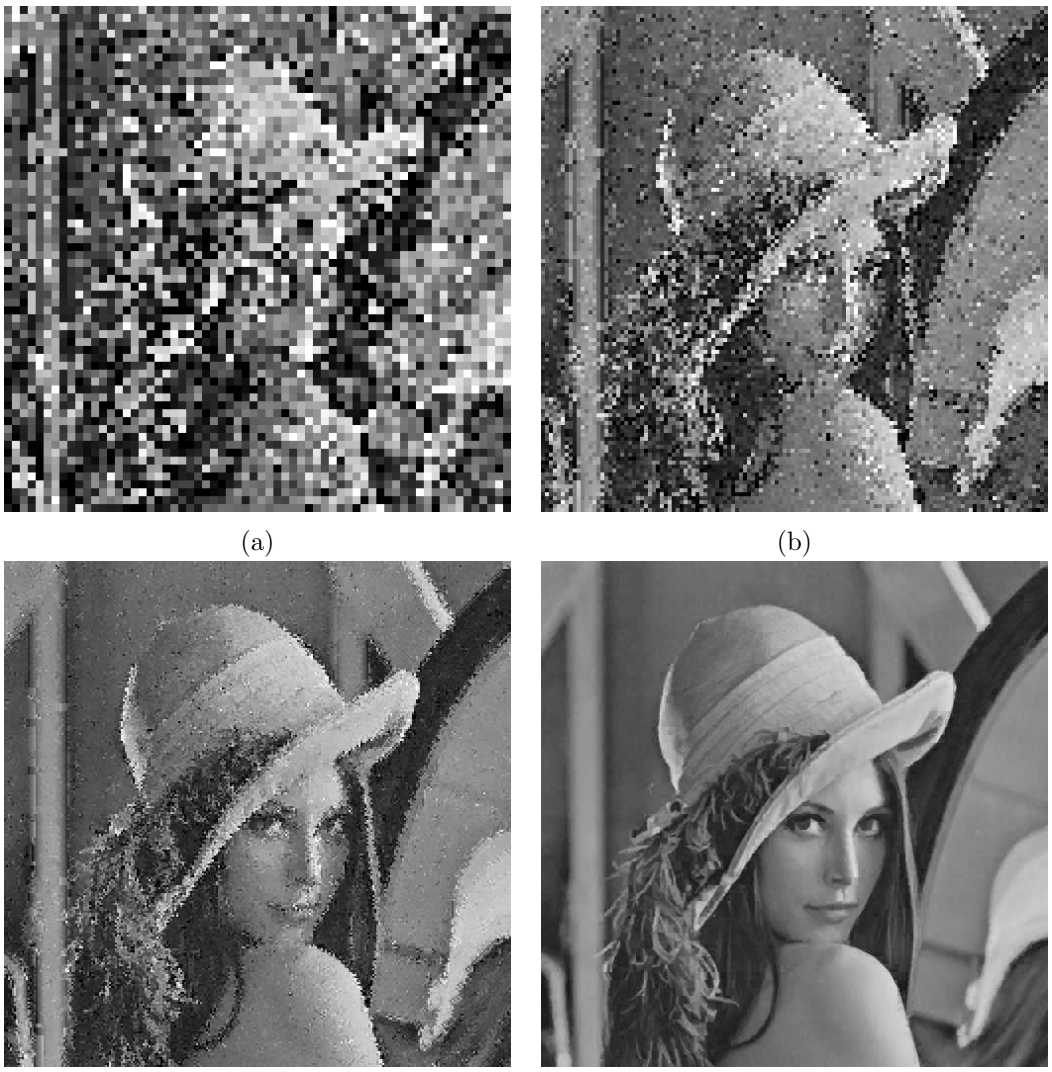
3.3.4 Matrix Notation of the Fractal Transform Operator

Given T, we can define a corresponding $\underline{T}$ acting on $\underline{\mathbf{x}}$ such that $\underline{T}(\underline{\mathbf{x}})$ represents $T(\mathbf{x})$ in the vector format. We take advantage of writing $\underline{\mathbf{x}}$ in the vector format, so that operator $\underline{T}$ can be written as

$$\underline{T}(\underline{\mathbf{x}}) = \mathbf{M}\underline{\mathbf{x}} + \mathbf{B}. \tag{3.22}$$

A similar type of matrix representation of the fractal transform operator was introduced in [Ham98]. Here, $\mathbf{M}$ is an $N^2 \times N^2$ matrix and both $\underline{\mathbf{x}}$ and $\mathbf{B}$ are vectors of dimension $N^2 \times 1$. Matrix $\mathbf{M}$ carries the α_i information, where $\mathbf{B}$ contains the information of β_i, all in the appropriate locations. Remember that α_i and β_i were the parameters of the greyscale maps. A nominal row of the matrix $\mathbf{M}$ corresponding to a pixel in the range block C_i has the following form.

$$\begin{pmatrix} \cdots\cdots & \cdot & \cdot & \cdots\cdots & \cdot & \cdots & \cdots\cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0\ldots\ldots 0 & \frac{\alpha_i}{4} & \frac{\alpha_i}{4} & 0\ldots 0 & \frac{\alpha_i}{4} & \frac{\alpha_i}{4} & 0\ldots\ldots 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots\cdots & \cdot & \cdot & \cdots\cdots & \cdot & \cdots & \cdots\cdots \end{pmatrix}$$



(a) (b)

(c) (d)

Figure 3.5: The iterates $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$, are respectively plotted in (a), (b), and (c). The fixed point $\mathbf{u}^\star$ of the fractal transform operator T designed to approximate the standard 512×512 (8 bpp) "Lena" image is shown in (d). The "seed" image was $\mathbf{u}_0(x, y) = 255$ (plain white). The fractal transform T was obtained by "fratcal coding" using 4096 nonoverlapping 8×8 pixel range blocks. The domain pool consisted of the set of 1024 nonoverlapping 16×16 pixel blocks.

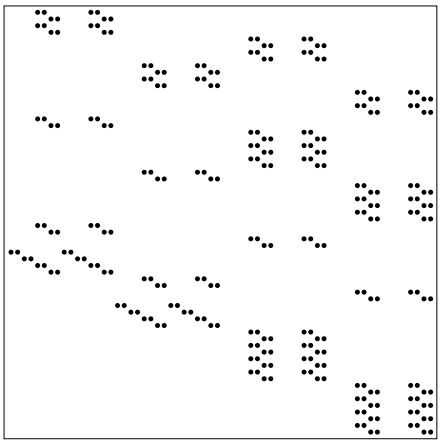


Figure 3.6: Typical sparsity of matrix $\mathbf{M}$, for an 8×8 image, i.e., $N = 8$. $\mathbf{M}$ is of dimension 64×64, i.e., $N^2 = 64$. Range and domain blocks are respectively of size 2×2 and 4×4. Number of nonzeros of $\mathbf{M}$ are $4N = 256$ out of the total of $(N^2)^2 = 4096$ elements.

Typically, $\mathbf{M}$ is very large and sparse with the ratio of non-zeros to total of $4/N^2$, for an $N \times N$ image. Figure 3.6 shows the sparsity structure of the matrix $\mathbf{M}$ corresponding to some small image.

By this notation , $\underline{\mathbf{u}}^\star$ is the solution of a linear equation of the form

$$\mathbf{M}\underline{\mathbf{x}} + \mathbf{B} = \underline{\mathbf{x}}, \tag{3.23}$$

or equivalently

$$\mathbf{A}\underline{\mathbf{x}} = \mathbf{B}, \tag{3.24}$$

where $\mathbf{A} = \mathbf{I}_{N^2 \times N^2} - \mathbf{M}$, and $\mathbf{I}_{N^2 \times N^2}$ is the identity matrix of size $N^2 \times N^2$. In the case that the fractal transform is fully known, this linear system has a unique solution $\underline{\mathbf{u}}^\star$, indeed by the structure of A. This unique solution is in fact the attractor of the fractal transform T in the vector order, i.e., $\underline{\mathbf{u}}^\star$. It is also true that for any $\underline{\mathbf{x}}$

$$\mathbf{A}\underline{\mathbf{x}} - \mathbf{B} = \underline{\mathbf{x}} - \underline{T}(\underline{\mathbf{x}}). \tag{3.25}$$

We will be using this matrix representation of the fractal transform operator in the some of the following chapters.

3.4 Variations and Applications of Fractal Image Coding

As was mentioned earlier, the original motivation of fractal coding was image compression [Jac92, Bar88, Lu97]. Jacquin's original paper [Jac92] launched an intensive activity in fractal image

compression. One of the main drawbacks of fractal coding was the time required to search for optimal domain blocks. As such, there was much investigation on how to achieve the best quality with as little searching as possible. A discussion of some of these methods can be found in [Fis95, Fis98, Lu97].

Furthermore, some of the most important variations and improvements of fractal coding include: variable partitioning schemes [Fis98], redundant representation and multiple-domain methods [Bon97, AVT08, XF], lapped approximations [HC97], and fractal coding on wavelet coefficient trees (IFSW) [Vrs98].

More recently, the ability of fractal coding to solve inverse problems other than compression has been investigated which will be described in the following sections.

3.4.1 Fractal Image Denoising

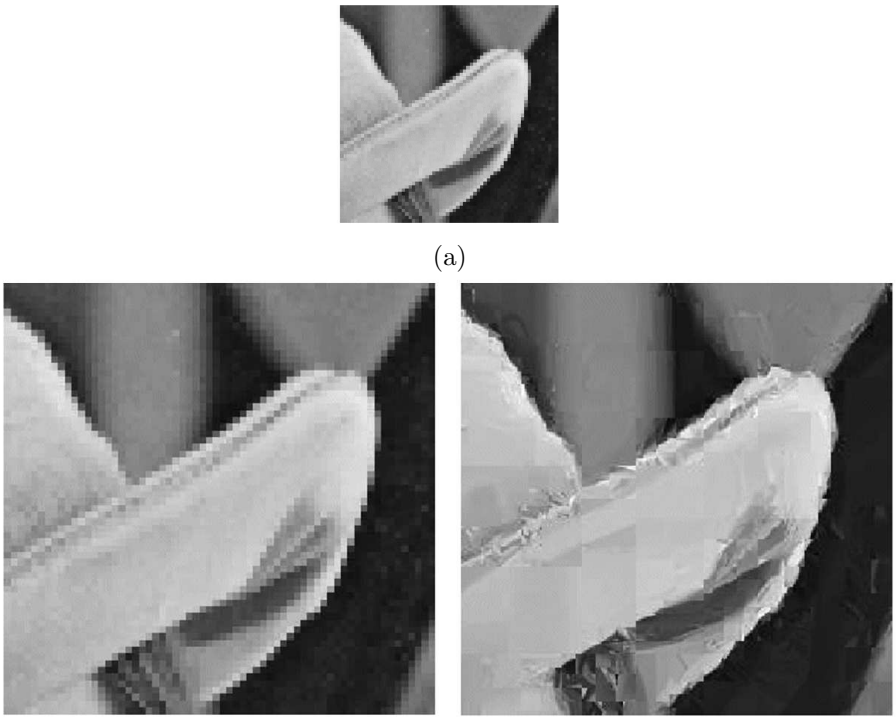
In [GFV03, GFV06, Gha04], it was first observed that fractal-based methods have denoising capabilities. There are two principal reasons: (i) the affine greyscale fitting between domain and range blocks causes some variance reduction in the noise and (ii) the spatial contraction/pixel decimation involved in mapping domain blocks to range blocks provides further variance reduction. Additional denoising can be achieved by using estimates of the noise variance to estimate the fractal code of the noiseless image [GFV03]. Furthermore, the fact that each range block may be well approximated by a number of domain blocks can be exploited to perform denoising by using multiple parents [Ale05].

3.4.2 Fractal Zoom

Re-sizing of an image does not translate into an increase in resolution. In fact, re-sizing should be accompanied by approximations for frequencies higher than those representable at original size, and at a higher signal-to-noise ratio. This process of re-sizing, called single-frame image zooming, will be precisely defined and addressed in Chapter 9.

In the traditional method of single-frame image zooming, interpolation functions are used. The original data are estimated by a function, usually continuous, and then resampled at a finer grid. The simplest interpolation algorithm is the so-called *nearest neighbour algorithm* where each unknown pixel is assigned the greyvalue of the sample closest to it. This method provides blocky results. More pleasant results can be obtained using *bilinear interpolation*, and *bicubic interpolation* schemes [Cha01]. Most of these algorithms provide overly smoothed results. Most of the difficulties arise in the areas around edges and sharp changes. Around edges, many interpolation methods tend to smooth and blur image details. Also, performing sharpening operators e.g., *un-sharp masking*, on these over-smoothed results does not usually recover the lost details of the image [Cha01].

It is believed that most image information is often found around edges and areas of high variation. This property can be used to predict the missing details from a sampled image. Several



(a)

(b) (c)

Figure 3.7: Fractal zoom example: (a) Original, (b) Pixel replication, (c) Fractal zoom.

methods have been used to address this problem. Prediction of the image details using evolution of wavelet coefficients across scales, and regularity-preserving image interpolation are examples of these approaches [CCH99, Mal98, MZ92].

Fractal-based zooming is another approach to achieve "superior" image details compared to conventional image enlargement techniques, e.g., bilinear interpolation. Due to the resolution-independent nature of the derived fractal transform operator, interpolation algorithms known as "fractal zoom" have been developed in the literature [Fis95, Fis98, GAaYTH97, Lu97, PD97] (see Figure 3.7). Very recently, it has been shown that such *superresolution* can be accomplished in the frequency domain using IFS-type methods [MV07].

In general, conventional fractal coding/decoding is a lossy operation, i.e., image details are partially lost in the coding/decoding process. More recent approaches such as fractal coding on wavelet coefficient trees (IFSW) [Vrs98] may preserve the image details and produce sharper results by employing the similarity of the wavelet coefficients as a *prior* model.

Figure 3.8 compares the result of pixel replication, bilinear interpolation, and the IFSW in the Haar basis as described in [Vrs98]. The bilinearly interpolated image is overly smoothed, while IFSW produces sharper yet noisy image.

The quality of the single-frame image zooming techniques is limited by the reliability and the amount of data available in the image. Single image zooming can not produce high frequency

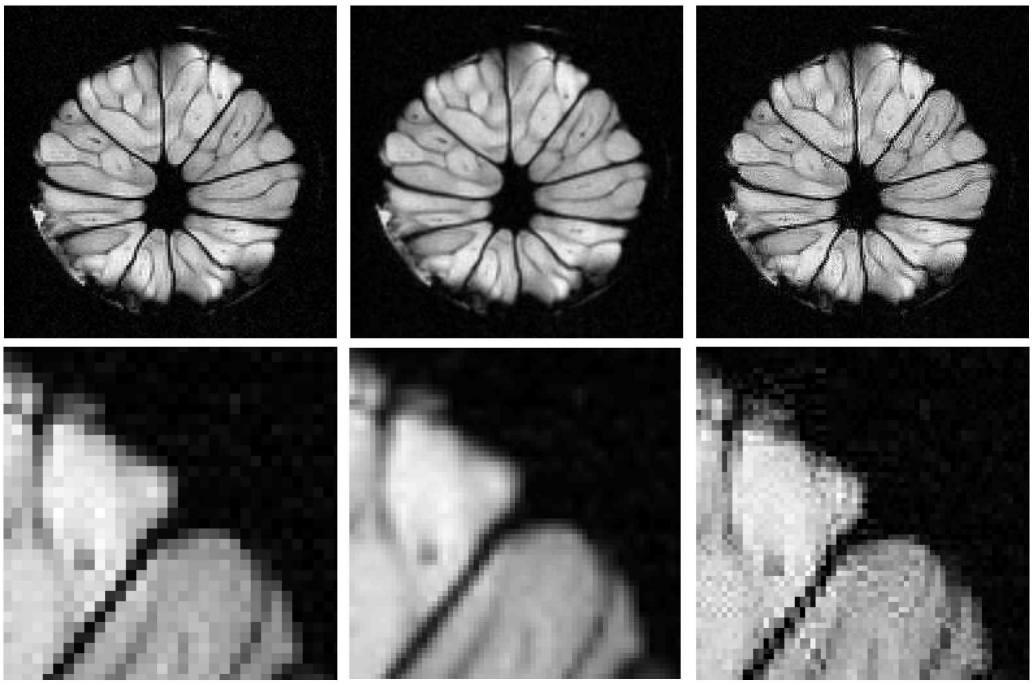


Figure 3.8: Top left to right: Pixel replication, Bilinear, IFSW, closer view of each image in the bottom.

components lost during the low-resolution sampling process unless a suitable model for zooming can be established. The prediction of details in an image requires *a priori* knowledge or assumption on the unknown image.

3.5 Newer Trends in Self-similarity

3.5.1 Statistical Studies on Image Self-similarity

Some recent investigations [Ale05, AVT08] have shown that images generally possess a great deal of local (affine) self-similarity: Given a subimage $\mathbf{u}|_{C_i}$ there are often a good number of domain blocks P_j whose subimages $\mathbf{u}|_{P_j}$ approximate it as well as the "best" domain block. This feature, which never seems to have been quantified previously, accounts for the rather small degradations that are experienced when the size of the *domain pools* – the domain blocks P_j to be examined – is decreased.

In [AVT08] was presented numerical evidence that the distributions of domain-range approximation errors can be used to characterize the relative degree of self-similarity of images. For many natural images, the distributions of errors demonstrate significant peaking near zero error. For others, the distributions are broader and peak farther away from zero. The latter images are judged to be less self-similar than the former. As noise of increasing variance is added to an image, the distributions broaden and move outward as well. In [EV08a], we have also examined

the distribution of neighbourhood distances at various scales for various natural images. It is observed that for each test image, the similarity between the same-scale and cross-scale histograms is striking. The details in producing the histograms and some of the results of [EV08a] will discussed in Chapter 8.

3.5.2 Non-local Methods

A new and important denoising method which involves self-similarity is nonlocal-means (NL-means) image denoising [BCM05b, BCM05c]. The authors have demonstrated that this algorithm has the ability to outperform classical denoising methods, including Gaussian smoothing, Wiener filter, TV filter, wavelet thresholding, and anisotropic diffusion. Furthermore, an extension of the algorithm has been developed [BCM05a] to address the problem of image denoising of image sequences. Due to the importance of the NL-means filter and the fact that we will be using it again in some of the following chapters we dedicate the next section to describe the method.

3.6 NL-means Image Denoising Filter

3.6.1 NL-means Image Denoising Filter

Consider the following image denoising problem [BCM05b, BCM05c],

$$\mathbf{u} = \mathbf{x} + \mathbf{n},$$

where $\mathbf{u} \in \ell^2(\Omega)$ is a given observation, $\mathbf{n} \in \ell^2(\Omega)$ is additive white independent Gaussian noise with zero-mean and variance σ^2, and $\mathbf{x} \in \ell^2(\Omega)$ is the image to be recovered where

$$\Omega = [1, \ldots, M] \times [1, \ldots, N].$$

For any $x \in \Omega$, define the approximation of $\mathbf{x}$ denoted by $\widetilde{\mathbf{x}_{NL}}$ as

$$\widetilde{\mathbf{x}_{NL}}(x) = \frac{1}{C(x)} \sum_{y \in \Omega} w(x,y)\mathbf{u}(y), \text{ such that} \tag{3.26}$$

$$w(x,y) = \exp\Big(- \frac{\| \mathbf{u}(\mathcal{N}^d\{x\}) - \mathbf{u}(\mathcal{N}^d\{y\}) \|_{2,a}^2}{h^2}\Big), \text{ and}$$

$$C(x) = \sum_{y \in \Omega} w(x,y), \tag{3.27}$$

where the expressions $\mathcal{N}^d\{\ldots\}$ and $\| \, . \, \|_{2,a}^2$ are defined in the following way.

Neighbourhoods: For any point in the domain of observation $(i,j) \in \Omega$, define

$$\mathcal{N}^d\big\{(i,j)\big\} = \Big\{(i+i', j+j') \mid (i',j') \in \mathbb{Z}^2, \max\{|i'|, |j'|\} \leq d\Big\}. \tag{3.28}$$

Gaussian-weighted-semi-norm: For any image patch

$$\mathbf{y} \in \ell^2\big([1,\ldots,2d+1]\times[1,\ldots,2d+1]\big),$$

define $\| \, . \, \|^2_{2,a}$ as

$$\| \, \mathbf{y} \, \|^2_{2,a} = \sum_{-d\le i\le d, -d\le j\le d} G_a(i,j)|\mathbf{y}(i+d+1, j+d+1)|^2$$

in which G_a is a two-dimensional Gaussian kernel of standard deviation a, centred at $(0,0)$, and of the same dimension as $\mathbf{y}$.

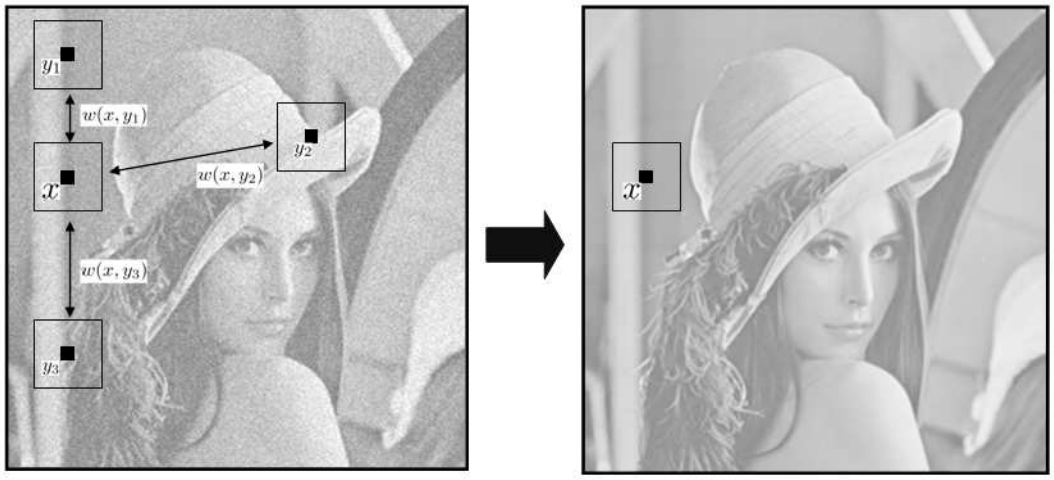


Figure 3.9: NL-means denoising, The intensity of each pixel in the noisy image is replaced with the weighted average of all of the pixel intensities in the same image. The weights are determined based on neighbourhood similarity of the pixels. Neighbourhoods are typically assumed as square patches of odd length.

The idea of the NL-means algorithm is that given a discrete noisy image $\mathbf{u}$, the estimated noiseless value $\widetilde{\mathbf{x}_{NL}}(x)$ is computed as a weighted average of all pixel intensities in the observed image, $\mathbf{u}(y)$, where the weights $w(x,y)$ depend on the similarity of neighbourhoods of the pixels x and y, and w is a decreasing function of the weighted Euclidean distance of the neighbourhoods. The NL-means denoising algorithm can be thought as an example-based technique in which examples are taken from the image itself. The examples, i.e., the patches, will be directly used to construct the posterior, which is used directly in the reconstruction process targeting the MMSE estimator, as was explained in Chapter 2.

The parameter h in the algorithm acts as a degree of filtering and controls decay of the exponential function and therefore the decay of the weights as a function of the Euclidean distances. Such an h plays the role of the regularization parameter of the inverse problem as was described earlier in Chapter 2. The NL-means algorithm not only compares the grey level in a single point but the geometrical configuration in a whole neighbourhood [BCM05b, BCM05c].

Restricting the Search-Window in the NL-means Filter

It turns out that the denoising algorithm above is computationally intensive. A possibility to overcome the computational complexity introduced in [BCM05b, BCM05c] is to restrict $y \in \Omega \cap \mathcal{N}^r\{x\}$, i.e., in a neighbourhood of x rather than the whole field of Ω in the corresponding equations. In [OEW08, OEW07], we have also examined other approaches to overcome the high computational cost of the NL-means algorithm.

NL-means Filter in the Matrix Notation

When $\mathbf{u}$ is represented in the vector format as $\underline{\mathbf{u}}$, the output of the NL-means denoising can be written as $\mathbf{W}\underline{u}$, where $\mathbf{W}$ is a right stochastic matrix containing the associated non-negative weights. This form of notation will be used in Chapter 7.

3.6.2 NL-means vs. Fractal Image Denoising

As mentioned ealier, the denoising properties of the fractal transform operator have been analyzed in [GFV03, GFV06, Ale05]. Here we briefly mention some of the main differences between fractal denoising and NL-means denoising:

- The blocks in fractal-based methods are typically taken at two different scales (coarse and fine), while the NL-means algorithm introduced in [BCM05b, BCM05c] considers only same-scale similarity.
- In the NL-means algorithm, blocks are compared solely in terms of the ℓ^2 distance between them. In fractal coding, an additional affine greyscale transformation is used.
- The NL-means is not an iterative scheme. One application of the operator completes the denoising task. However, the fractal denoising scheme [GFV03] is iterative. In [EV07a] it is shown how one may construct a contractive operator along with an iterative scheme for NL-means denoising. However, the iteration procedure is not computationally feasible since the matrix representation of the NL-means operator is not sparse, unlike the case for fractal transforms. This contribution will be presented in Chapter 7.
- The NL-means scheme involves the replacement of one pixel at a time. This produces a kind of continuity across pixels, in contrast to standard fractal decoding, which generally suffers from blockiness across range block boundaries, since entire range blocks are being replaced.
- In the NL-means scheme, each pixel is replaced by a weighted average of all pixel intensities in the image, where blocks of greater similarity have larger weights. Such a counterpart exists in fractal image coding in the form of multiparent transforms [Ale05], but has not been extensively employed.

Chapter 4

A Necessary and Sufficient Contractivity Condition for the Fractal Transform Operator

4.1 Introduction

In the previous chapter, we defined the fractal transform operator T and stated that under suitable conditions the operator T is contractive. Hence, using Banach's contraction mapping principle, T has a unique fixed point $\mathbf{u}^\star$ such that $T(\mathbf{u}^\star) = \mathbf{u}^\star$.

There are a few existing *sufficient* contractivity conditions for the fractal transform operator in the literature [BEHL85, FV98, Fis95, Fis98]. These conditions are *only sufficient* and *not necessary*. Furthermore in the existing results, images are typically assumed to be *continuous* and not *discrete* in nature. Therefore, it is not precisely clear how the theory would relate to practice, using these results.

In [Ale05], it is stated that such contractivity conditions are never checked in practice. In [Fis95, Fis98] the author suggests that based on "computational experiments" the operator T is converging if the coefficients α_i are in the interval $[-\sqrt{2}, \sqrt{2}]$. Furthermore, in [Gha99, Gha04] it is stated that "In fact, for block-based fractal schemes, it is indeed very difficult to derive a tight necessary contractivity requirement on the IFS coefficients." In this chapter, we will discuss that the existing sufficient contractivity conditions almost never hold and are probably useless to be applied in practice. These sufficient conditions, as we will show, are far from being necessary.

The first step would be to check that we can at least obtain the same sufficient contractivity conditions for the fractal transform operator, reported in [FV98] for the continuous case, now for our discrete setting. Note that all computations of fractal coding experiments are performed in the discrete setting. Therefore, in this chapter, as before we assume the space of images as $\ell^2(\Upsilon)$, for

$$\Upsilon = [1, \ldots, N] \times [1, \ldots, N] \tag{4.1}$$

i.e., we remain in a space of discrete images. Our advantage here would be the fact that we remain consistent in both theory and practice, i.e., both our propositions and experiments would be in the same discrete space. This first step will be accomplished in Section 4.2.

In Section 4.3, we further improve this existing sufficient condition and yield a necessary and sufficient condition for the contractivity of the fractal transform operator. This will be an improvement of the existing condition by a factor of 100 or so for many typical experiments.

In Section 4.4, we ask the question whether contractivity of T is necessary at all for the convergence of the fractal decoding scheme. The answer is negative and we state analytical conditions under which the fractal decoding iteration converges. Furthermore, we derive a theorem in this *non-contracting yet converging* case, corresponding to the Collage Theorem (Equation 3.20). We will explain how this result generalizes the Collage Theorem and will refer to the result as "The generalized Collage Theorem".

In Section 4.5, we consider a very special case of the fractal transform operator which has the same domain and range block size.

Finally, some concluding remarks will be given in Section 4.6.

4.2 Existing Contractivity Conditions for the Fractal Transform Operator

In this section, we verify the sufficient contractivity conditions of the fractal transform operator, reported in [FV98] for the continuous case. Here we remain in a discrete space $\ell^2(\Upsilon)$. As mentioned earlier, implementations of fractal coding are typically performed in the discrete space while the existing theory is developed to address the continuous case. The definitions of the down-sampling operator and the block mappings are of crucial importance in this context. We follow a mathematically rigorous setting to ensure the clarity of the statements.

Again recall that we defined the fractal transform operator T as

$$(T(\mathbf{u}))(x,y) = \sum_{i\in\mathfrak{C}} \phi_i\Big(\mathcal{D}\big(\mathbf{u}(w_i^{-1}(x,y))\big)\Big), \tag{4.2}$$

for every $(x,y)\in\Upsilon$, and also assumed that the greyscale map $\phi_i = \alpha_i t + \beta_i$. Given $\mathbf{u}\in\ell^2(\Upsilon)$, we briefly outlined how to compute its associated fractal transform operator T. We explained that $\mathbf{u}$ is approximated by the fixed point $\mathbf{u}^\star$ of T.

Proposition 4.2.1. *For any $\mathbf{y}\in\ell^2(\Upsilon)$ and $i\in\mathfrak{C}$,*

$$\Big\|\mathcal{D}\big(\mathbf{y}\big|_{P_{J(i)}}\big)\Big\| \le \frac{1}{s}\Big\|\mathbf{y}\big|_{P_{J(i)}}\Big\|. \tag{4.3}$$

Proof. Assume that $P_{J(i)}$ is an $sK\times sK$ block and $\mathbf{y}\big|_{P_{J(i)}}$ is represented by K^2 blocks each of size $s\times s$ by $y_{p,q}^{(m)}$ for $1\le p,q\le s$, and $1\le m\le K^2$. By this representation,

$$\Big\|\mathcal{D}\big(\mathbf{y}\big|_{P_{J(i)}}\big)\Big\|^2 = \sum_{m=1}^{K^2}\big[\frac{1}{s^2}\sum_{1\le p,q\le s} y_{p,q}^{(m)}\big]^2 \tag{4.4}$$

For every $1\le m\le K^2$, taking $a_{p,q}=y_{p,q}^{(m)}$ and $b_{p,q}=\frac{1}{s^2}$ in the Cauchy-Schwarz inequality

$$\Big(\sum_{1\le p,q\le s^2} a_{p,q}b_{p,q}\Big)^2 \le \Big(\sum_{1\le p,q\le s^2} a_{p,q}^2\Big)\Big(\sum_{1\le p,q\le s^2} b_{p,q}^2\Big), \tag{4.5}$$

leads to

$$\begin{aligned}\big[\frac{1}{s^2}\sum_{1\le p,q\le s} y_{p,q}^{(m)}\big]^2 &\le \Big(\sum_{1\le p,q\le s^2} {y_{p,q}^{(m)}}^2\Big)\Big(\sum_{1\le p,q\le s^2}\frac{1}{s^4}\Big) &(4.6)\\ &= \Big(\sum_{1\le p,q\le s^2} {y_{p,q}^{(m)}}^2\Big)\Big(\frac{1}{s^2}\Big). &(4.7)\end{aligned}$$

Hence, taking the sum over m yields

$$\sum_{m=1}^{K^2}\big[\frac{1}{s^2}\sum_{1\le p,q\le s} y_{p,q}^{(m)}\big]^2 \le \frac{1}{s^2}\sum_{m=1}^{K^2}\Big(\sum_{1\le p,q\le s^2} {y_{p,q}^{(m)}}^2\Big). \tag{4.8}$$

Substituting the equivalent values for the left and right sides gives,

$$\Big\|\mathcal{D}\big(\mathbf{y}\big|_{P_{J(i)}}\big)\Big\|^2 \le \frac{1}{s^2}\Big\|\mathbf{y}\big|_{P_{J(i)}}\Big\|^2. \tag{4.9}$$

Finally taking the square root of both sides of this expression completes the proof. □

Proposition 4.2.2. *T is a contraction on $\ell^2(\Upsilon)$ if*

$$\sqrt{\sum_{i\in\mathfrak{C}}(\frac{\alpha_i}{s})^2}<1 \tag{4.10}$$

where, the size of range and domain subblocks are respectively $K\times K$ and $sK\times sK$.

Proof. For any $\mathbf{u}$,$\mathbf{v}$ in $\ell^2(\Upsilon)$,

$$\begin{aligned}\|T(\mathbf{u}) &- T(\mathbf{v})\|^2 && (4.11)\\ &= \sum_{i\in\mathfrak{C}}\sum_{x,y\in C_i}\Big|\phi_i\Big(\mathcal{D}\big(\mathbf{u}(w_i^{-1}(x,y))\big)\Big)-\phi_i\Big(\mathcal{D}\big(\mathbf{v}(w_i^{-1}(x,y))\big)\Big)\Big|^2 && (4.12)\\ &= \sum_{i\in\mathfrak{C}}\Big\|\phi_i\Big(\mathcal{D}\big(\mathbf{u}(w_i^{-1}(C_i))\big)\Big)-\phi_i\Big(\mathcal{D}\big(\mathbf{v}(w_i^{-1}(C_i))\big)\Big)\Big\|^2. && (4.13)\end{aligned}$$

Assuming $\phi_i(t)=\alpha_i(t)+\beta_i$ as before yields

$$\begin{aligned}\|T(\mathbf{u})-T(\mathbf{v})\|^2 &= \sum_{i\in\mathfrak{C}}{\alpha_i}^2\Big\|\mathcal{D}\big(\mathbf{u}(w_i^{-1}(C_i))\big)-\mathcal{D}\big(\mathbf{v}(w_i^{-1}(C_i))\big)\Big\|^2 && (4.14)\\ &= \sum_{i\in\mathfrak{C}}{\alpha_i}^2\Big\|\mathcal{D}\big(\mathbf{u}\big|_{P_{J(i)}}\big)-\mathcal{D}\big(\mathbf{v}\big|_{P_{J(i)}}\big)\Big\|^2 && (4.15)\\ &= \sum_{i\in\mathfrak{C}}{\alpha_i}^2\Big\|\mathcal{D}\big((\mathbf{u}-\mathbf{v})\big|_{P_{J(i)}}\big)\Big\|^2. && (4.16)\end{aligned}$$

Hence, by the previous proposition taking $\mathbf{y}=\mathbf{u}-\mathbf{v}$,

$$\begin{aligned}\|T(\mathbf{u})-T(\mathbf{v})\|^2 &\le \sum_{i\in\mathfrak{C}}\frac{\alpha_i^2}{s^2}\Big\|(\mathbf{u}-\mathbf{v})\big|_{P_{J(i)}}\Big\|^2 && (4.17)\\ &\le \sum_{i\in\mathfrak{C}}(\frac{\alpha_i}{s})^2\Big\|(\mathbf{u}-\mathbf{v})\Big\|^2 && (4.18)\\ &= [\sum_{i\in\mathfrak{C}}(\frac{\alpha_i}{s})^2]\Big\|(\mathbf{u}-\mathbf{v})\Big\|^2. && (4.19)\end{aligned}$$

Hence,

$$\|T(\mathbf{u})-T(\mathbf{v})\|\le\sqrt{\sum_{i\in\mathfrak{C}}(\frac{\alpha_i}{s})^2}\Big\|(\mathbf{u}-\mathbf{v})\Big\|, \tag{4.20}$$

and the condition

$$\sqrt{\sum_{i\in\mathfrak{C}}(\frac{\alpha_i}{s})^2}<1 \tag{4.21}$$

is sufficient for T to be a contraction. □

The sufficient condition

$$\sqrt{\sum_{i\in\mathfrak{C}}(\frac{\alpha_i}{s})^2}<1 \tag{4.22}$$

has the same form as the sufficient condition given in [FV98]. As explained earlier our advantage here is the fact that we here remain consistent in both theory and practice. We deduced this result for the discrete case, which is exactly consistent with what is being done computationally.

The following is just a simple result of the above proposition.

Proposition 4.2.3. *As a result of the previous proposition, a sufficient condition for contractivity of T on $\ell^2(\Upsilon)$, where as before $\Upsilon = [1,\ldots,N]\times[1,\ldots,N]$, is*

$$\max_{i\in\mathfrak{C}}\{|\alpha_i|\} < \frac{s}{\sqrt{|\mathfrak{C}|}} = \frac{sK}{N}. \tag{4.23}$$

Proof. First assuming that there are $(\frac{N}{K})^2$ range blocks in $\mathfrak{C}$,

$$\sqrt{|\mathfrak{C}|} = \sqrt{\left(\frac{N}{K}\right)^2} = \frac{N}{K}, \tag{4.24}$$

which proves the equality part. To prove the sufficiency, assuming that the inequality holds, for any $i \in \mathfrak{C}$,

$$|\alpha_i| \le \max_{i\in\mathfrak{C}}\{|\alpha_i|\} < \frac{s}{\sqrt{|\mathfrak{C}|}}, \tag{4.25}$$

hence,

$$\frac{|\alpha_i|}{s} < \frac{1}{\sqrt{|\mathfrak{C}|}}. \tag{4.26}$$

Therefore,

$$\sum_{i\in\mathfrak{C}}(\frac{\alpha_i}{s})^2 < \sum_{i\in\mathfrak{C}}\frac{1}{|\mathfrak{C}|} = \frac{|\mathfrak{C}|}{|\mathfrak{C}|} = 1, \tag{4.27}$$

which is the same sufficient condition as in the previous proposition. □

4.3 Extending the Contractivity Conditions for the Fractal Transform Operator

In this section, we improve the existing sufficient condition stated in the previous section to produce a both necessary and sufficient condition for the contractivity of the fractal transform operator.

The proposition below is a stronger statement, i.e., a more relaxed sufficient condition for the contractivity of T.

Proposition 4.3.1. *A sufficient condition for the contractivity of T on $\ell^2(\Upsilon)$ is*

$$\max_{j\in\mathfrak{P}}\left\{\sqrt{\sum_{i\in\mathfrak{C}_j}(\frac{\alpha_i}{s})^2}\right\} < 1, \tag{4.28}$$

where the sets $\mathfrak{C}_j = \{i \in \mathfrak{C} | J(i) = j\}$ for any $j \in \mathfrak{P}$.

Proof. Back in the middle of the proof of the first contractivity Proposition 4.2.2, we had for any $\mathbf{u}$,$\mathbf{v}$ in $\ell^2(\Upsilon)$,

$$\|T(\mathbf{u}) - T(\mathbf{v})\|^2 \leq \sum_{i\in\mathfrak{C}} (\frac{\alpha_i}{s})^2 \Big\| (\mathbf{u}-\mathbf{v})\big|_{P_{J(i)}} \Big\|^2. \tag{4.29}$$

Note that $\{\mathfrak{C}_j\}_{j\in\mathfrak{P}}$ forms a partition for $\mathfrak{C}$, i.e., $\mathfrak{C}_{j_1} \cap \mathfrak{C}_{j_2} = \emptyset$ for distinct $j_1, j_2 \in \mathfrak{P}$ and

$$\mathfrak{C} = \bigcup_{j\in\mathfrak{P}} \mathfrak{C}_j \tag{4.30}$$

(see Figure 4.1).

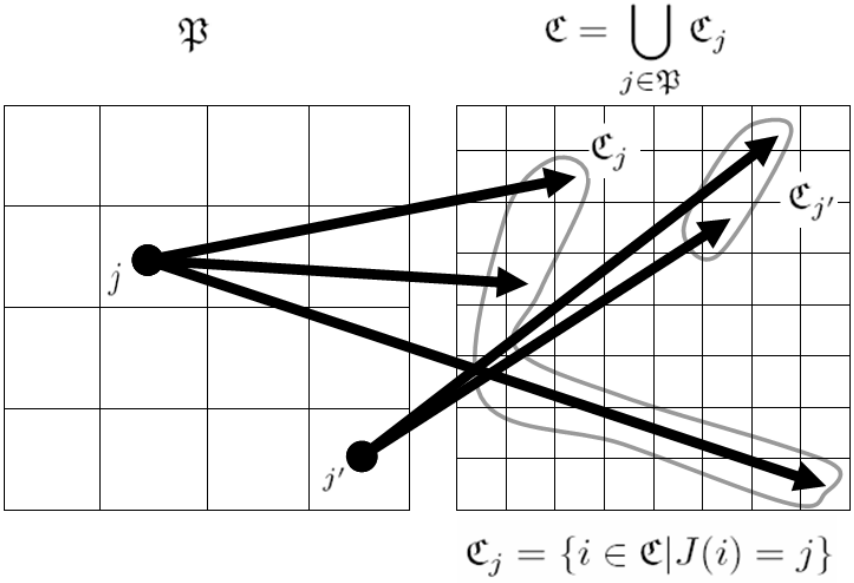


Figure 4.1: Partitioning of $\mathfrak{C}$. Two elements of the partition are shown on the right.

Hence,

$$\begin{aligned}
\|T(\mathbf{u}) - T(\mathbf{v})\|^2 &\leq \sum_{j\in\mathfrak{P}} \sum_{i\in\mathfrak{C}|J(i)=j} (\frac{\alpha_i}{s})^2 \Big\| (\mathbf{u}-\mathbf{v})\big|_{P_j} \Big\|^2 && (4.31)\\
&= \sum_{j\in\mathfrak{P}} \sum_{i\in\mathfrak{C}_j} (\frac{\alpha_i}{s})^2 \Big\| (\mathbf{u}-\mathbf{v})\big|_{P_j} \Big\|^2 && (4.32)\\
&= \sum_{j\in\mathfrak{P}} \left[\Big\| (\mathbf{u}-\mathbf{v})\big|_{P_j} \Big\|^2 \sum_{i\in\mathfrak{C}_j} (\frac{\alpha_i}{s})^2 \right] && (4.33)\\
&\leq \sum_{j\in\mathfrak{P}} \left[\Big\| (\mathbf{u}-\mathbf{v})\big|_{P_j} \Big\|^2 \right] \max_{j\in\mathfrak{P}} \Big\{ \sum_{i\in\mathfrak{C}_j} (\frac{\alpha_i}{s})^2 \Big\} && (4.34)\\
&\leq \Big\| (\mathbf{u}-\mathbf{v}) \Big\|^2 \max_{j\in\mathfrak{P}} \Big\{ \sum_{i\in\mathfrak{C}_j} (\frac{\alpha_i}{s})^2 \Big\}. && (4.35)
\end{aligned}$$

Taking the square root yields,

$$\|T(\mathbf{u}) - T(\mathbf{v})\| \leq \Big\|(\mathbf{u}-\mathbf{v})\Big\| \max_{j\in\mathfrak{P}} \Big\{ \sqrt{\sum_{i\in\mathfrak{C}_j} (\frac{\alpha_i}{s})^2} \Big\}. \tag{4.36}$$

Therefore the condition

$$\max_{j\in\mathfrak{P}} \Big\{ \sqrt{\sum_{i\in\mathfrak{C}_j} (\frac{\alpha_i}{s})^2} \Big\} < 1 \tag{4.37}$$

satisfies the convergence of T. □

What we just proved shows that a much weaker condition results in the convergence of the fractal transform operator. This is because the summation of contractivity condition is now restricted to $i \in \mathfrak{C}_j$, which seems a much smaller portion of values in $\mathfrak{C}_j \subset \mathfrak{C}$ compared to the unconditional case of $i \in \mathfrak{C}$. This is an improvement of the condition by a factor of 100 or so in many typical experiments.

Again, the following is just a simple result of the above proposition .

Proposition 4.3.2. *A sufficient condition for contractivity of T on $\ell^2(\Upsilon)$ is*

$$\max_{i\in\mathfrak{C}} \{|\alpha_i|\} < \frac{s}{\sqrt{\mathfrak{C}_{max}}}, \tag{4.38}$$

where $\mathfrak{C}_{max}$ is the maximum number of times any single domain block is mapped to some range block, i.e., $\mathfrak{C}_{max} = \max_{j\in\mathfrak{P}} |\mathfrak{C}_j|$.

Proof. Define $\alpha_{max} = \max_{i\in\mathfrak{C}}\{|\alpha_i|\}$, and assume $\alpha_{max} < \frac{s}{\sqrt{\mathfrak{C}_{max}}}$. This gives,

$$\begin{aligned}
\sqrt{\sum_{i\in\mathfrak{C}_j} \big(\frac{\alpha_i}{s}\big)^2} &\leq \sqrt{\sum_{i\in\mathfrak{C}_j} \big(\frac{\alpha_{max}}{s}\big)^2} && (4.39)\\
&= \big(\frac{\alpha_{max}}{s}\big)\sqrt{|\mathfrak{C}_j|} && (4.40)\\
&< \big(\frac{s}{\sqrt{\mathfrak{C}_{max}}}\frac{1}{s}\big)\sqrt{|\mathfrak{C}_j|} && (4.41)\\
&< \frac{\sqrt{|\mathfrak{C}_j|}}{\sqrt{|\mathfrak{C}_{max}|}} && (4.42)\\
&\leq 1. && (4.43)
\end{aligned}$$

Therefore, the sufficient contractivity condition of the previous proposition is satisfied. □

In the worst case that all of the range blocks are mapped from a single domain block, $\mathfrak{C}_{max}$ would be equal to the number of all range blocks, or $|\mathfrak{C}|$. This leads to

$$\max_{i\in\mathfrak{C}} \{|\alpha_i|\} < \frac{s}{\sqrt{\mathfrak{C}_{max}}} = \frac{s}{\sqrt{|\mathfrak{C}|}}, \tag{4.44}$$

which is equivalent to the Proposition 4.2.3. However, in a typical situation where each domain block is related to only a few range blocks $\mathfrak{C}_{max}$ is small and α_is can be relaxed and take larger values in magnitude up to $\frac{s}{\sqrt{\mathfrak{C}_{max}}}$.

Now we would like to prove that the sufficient condition in Proposition 4.3.1 is also necessary.

Proposition 4.3.3. *A necessary and sufficient condition for contractivity of T on $\ell^2(\Upsilon)$ is*

$$\max_{j\in\mathfrak{P}}\Big\{\sqrt{\sum_{i\in\mathfrak{C}_j}(\frac{\alpha_i}{s})^2}\Big\}<1, \tag{4.45}$$

where the sets $\mathfrak{C}_j=\{i\in\mathfrak{C}|J(i)=j\}$ for any $j\in\mathfrak{P}$.

Proof. We previously proved the sufficiency. To prove the necessity, assume that T is a contraction. Hence, there exists for $c<1$ such that for any $\mathbf{u}$,$\mathbf{v}$ in $\ell^2(\Upsilon)$,

$$\|T(\mathbf{u})-T(\mathbf{v})\|\le c\Big\|\mathbf{u}-\mathbf{v}\Big\|. \tag{4.46}$$

For any $j\in\mathfrak{P}$ take the pair of images $\mathbf{u}=\mathbf{u}^{(j)}$ and $\mathbf{v}=\mathbf{v}^{(j)}$ defined in the following manner. $\mathbf{u}^{(j)}(x,y)=1_{P_j}(x,y)$ for any $(x,y)\in\Upsilon$, where 1_{P_j} represents the characteristic or indicator function on P_j. Also, take $\mathbf{v}^{(j)}(x,y)=0$ for any $(x,y)\in\Upsilon$. Hence,

$$\begin{aligned}
&\|T(\mathbf{u}^{(j)})-T(\mathbf{v}^{(j)})\|^2 && (4.47)\\
=\;&\sum_{i\in\mathfrak{C}}\sum_{x,y\in C_i}\Big|\phi_i\Big(\mathcal{D}\big(\mathbf{u}^{(j)}(w_i^{-1}(x,y))\big)\Big)-\phi_i\Big(\mathcal{D}\big(\mathbf{v}^{(j)}(w_i^{-1}(x,y))\big)\Big)\Big|^2 && (4.48)\\
=\;&\sum_{i\in\mathfrak{C}}\Big\|\phi_i\Big(\mathcal{D}\big(\mathbf{u}^{(j)}(w_i^{-1}(C_i))\big)\Big)-\phi_i\Big(\mathcal{D}\big(\mathbf{v}^{(j)}(w_i^{-1}(C_i))\big)\Big)\Big\|^2. && (4.49)
\end{aligned}$$

Assuming $\phi_i(t)=\alpha_i(t)+\beta_i$ as before yields

$$\begin{aligned}
\|T(\mathbf{u}^{(j)}) &- T(\mathbf{v}^{(j)})\|^2 && (4.50)\\
&=\sum_{i\in\mathfrak{C}}{\alpha_i}^2\Big\|\mathcal{D}\big(\mathbf{u}^{(j)}(w_i^{-1}(C_i))\big)-\mathcal{D}\big(\mathbf{v}^{(j)}(w_i^{-1}(C_i))\big)\Big\|^2 && (4.51)\\
&=\sum_{i\in\mathfrak{C}}{\alpha_i}^2\Big\|\mathcal{D}\big(\mathbf{u}^{(j)}\Big|_{P_{J(i)}}\big)-\mathcal{D}\big(\mathbf{v}^{(j)}\Big|_{P_{J(i)}}\big)\Big\|^2 && (4.52)\\
&=\sum_{i\in\mathfrak{C}}{\alpha_i}^2\Big\|\mathcal{D}\big((\mathbf{u}^{(j)}-\mathbf{v}^{(j)})\Big|_{P_{J(i)}}\big)\Big\|^2 && (4.53)\\
&=\sum_{i\in\mathfrak{C}}{\alpha_i}^2\Big\|\mathcal{D}\big((1_{P_j}-0)\Big|_{P_{J(i)}}\big)\Big\|^2 && (4.54)\\
&=\sum_{i\in\mathfrak{C}}{\alpha_i}^2\Big\|\mathcal{D}\big((1_{P_j})\Big|_{P_{J(i)}}\big)\Big\|^2. && (4.55)
\end{aligned}$$

Breaking the sum $i \in \mathfrak{C}$ over two disjoint cases of $i \in \mathfrak{C}, J(i) = j$ and $i \in \mathfrak{C}, J(i) \neq j$ yields

$$\begin{aligned}
& \|T(\mathbf{u}^{(j)}) - T(\mathbf{v}^{(j)})\|^2 && (4.56)\\
= & \sum_{i\in\mathfrak{C}, J(i)=j} {\alpha_i}^2 \Big\| \mathcal{D}\big((1_{P_j})\Big|_{P_{J(i)}}\big) \Big\|^2 + \sum_{i\in\mathfrak{C}, J(i)\neq j} {\alpha_i}^2 \Big\| \mathcal{D}\big((1_{P_j})\Big|_{P_{J(i)}}\big) \Big\|^2 && (4.57)\\
= & \sum_{i\in\mathfrak{C}, J(i)=j} {\alpha_i}^2 \Big\| \mathcal{D}\big(1_{P_j}\big) \Big\|^2 && (4.58)\\
= & \sum_{i\in\mathfrak{C}, J(i)=j} {\alpha_i}^2 \frac{1}{s^2} \Big\| 1_{P_j} \Big\|^2 && (4.59)\\
= & \Big\| 1_{P_j} \Big\|^2 \sum_{i\in\mathfrak{C}, J(i)=j} \big(\frac{\alpha_i}{s}\big)^2 && (4.60)\\
= & \Big\| 1_{P_j} \Big\|^2 \sum_{i\in\mathfrak{C}_j} \big(\frac{\alpha_i}{s}\big)^2. && (4.61)
\end{aligned}$$

Also,

$$\|\mathbf{u}^{(j)} - \mathbf{v}^{(j)}\|^2 = \Big\| 1_{P_j} - 0 \Big\|^2 = \Big\| 1_{P_j} \Big\|^2. \tag{4.62}$$

Hence for any $j \in \mathfrak{P}$, replacing $\mathbf{u} = \mathbf{u}^{(j)}$ and $\mathbf{v} = \mathbf{v}^{(j)}$ in

$$\|T(\mathbf{u}) - T(\mathbf{v})\| \leq c \Big\| \mathbf{u} - \mathbf{v} \Big\|, \tag{4.63}$$

gives

$$\|T(\mathbf{u}^{(j)}) - T(\mathbf{v}^{(j)})\| \leq c \Big\| \mathbf{u}^{(j)} - \mathbf{v}^{(j)} \Big\|, \tag{4.64}$$

and replacing the equivalent values we just found yields that for any $j \in \mathfrak{P}$

$$\Big\| 1_{P_j} \Big\| \sqrt{\sum_{i\in\mathfrak{C}_j} \big(\frac{\alpha_i}{s}\big)^2} \leq c \Big\| 1_{P_j} \Big\|, \tag{4.65}$$

for some $c < 1$. Finally cancelling the positive value $\Big\| 1_{P_j} \Big\|$ from both sides of the inequality gives that for any $j \in \mathfrak{P}$

$$\sqrt{\sum_{i\in\mathfrak{C}_j} \big(\frac{\alpha_i}{s}\big)^2} \leq c < 1. \tag{4.66}$$

Hence for any $j \in \mathfrak{P}$,

$$\sqrt{\sum_{i\in\mathfrak{C}_j} \big(\frac{\alpha_i}{s}\big)^2} < 1, \tag{4.67}$$

or equivalently

$$\max_{j\in\mathfrak{P}} \Big\{ \sqrt{\sum_{i\in\mathfrak{C}_j} \big(\frac{\alpha_i}{s}\big)^2} \Big\} < 1. \tag{4.68}$$

□

The following proposition states that the smallest contractivity factor of T can be obtained using the developed necessary and sufficient condition.

Proposition 4.3.4. *If T is a contraction on $\ell^2(\Upsilon)$ and $con_{(T)}$ is defined as the infimum taken over all of the possible contractivity factors of T then*

$$con_{(T)} = \max_{j\in\mathfrak{P}}\Big\{\sqrt{\sum_{i\in\mathfrak{C}_j}(\frac{\alpha_i}{s})^2}\Big\} < 1, \tag{4.69}$$

where the sets $\mathfrak{C}_j = \{i \in \mathfrak{C} | J(i) = j\}$ for any $j \in \mathfrak{P}$.

Proof. If T is a contraction, the necessity condition we just proved satisfies that

$$\max_{j\in\mathfrak{P}}\Big\{\sqrt{\sum_{i\in\mathfrak{C}_j}(\frac{\alpha_i}{s})^2}\Big\} < 1. \tag{4.70}$$

Now, we know that this inequality holds and observe the Equation 4.36 in the last paragraph of the proof of sufficiency Proposition 4.3.1, in which we had

$$\|T(\mathbf{u}) - T(\mathbf{v})\| \leq \Big\|(\mathbf{u}-\mathbf{v})\Big\| \max_{j\in\mathfrak{P}}\Big\{\sqrt{\sum_{i\in\mathfrak{C}_j}(\frac{\alpha_i}{s})^2}\Big\}. \tag{4.71}$$

This shows that the expression

$$\max_{j\in\mathfrak{P}}\Big\{\sqrt{\sum_{i\in\mathfrak{C}_j}(\frac{\alpha_i}{s})^2}\Big\} \tag{4.72}$$

is a contractivity factor of T, and hence $con_{(T)}$ which is the infimum over all possible contractivity factor of T satisfies

$$con_{(T)} \leq \max_{j\in\mathfrak{P}}\Big\{\sqrt{\sum_{i\in\mathfrak{C}_j}(\frac{\alpha_i}{s})^2}\Big\}. \tag{4.73}$$

It is very easy to prove and generally known that the infimum of all contractivity factors of T, denoted by $con_{(T)}$, is itself a contractivity factor of T. Hence, using the statement we had in the last paragraph of the necessity condition taking $c = con_{(T)}$ gives

$$\max_{j\in\mathfrak{P}}\Big\{\sqrt{\sum_{i\in\mathfrak{C}_j}(\frac{\alpha_i}{s})^2}\Big\} \leq con_{(T)}. \tag{4.74}$$

Hence,

$$con_{(T)} = \max_{j\in\mathfrak{P}}\Big\{\sqrt{\sum_{i\in\mathfrak{C}_j}(\frac{\alpha_i}{s})^2}\Big\} < 1. \tag{4.75}$$

This shows that the smallest possible contractivity factor of T is

$$con_{(T)} = \max_{j\in\mathfrak{P}}\Big\{\sqrt{\sum_{i\in\mathfrak{C}_j}(\frac{\alpha_i}{s})^2}\Big\}, \tag{4.76}$$

which will be referred to as *the contractivity factor of T*. □

In the fractal coding literature, the fractal transform operator T is sometimes only defined in terms of downsampling $\mathcal{S}$ as opposed to the operator $\mathcal{D}$, i.e., averaging followed by downsampling. We have given the precise definitions of these operators in Section 1.4. The following proposition addresses the difference between these two cases, in terms of contractivity conditions.

Proposition 4.3.5. *Assume that sampling without averaging is used in the definition of T, i.e., $\mathcal{S}$ instead of $\mathcal{D}$ [see the notations in Section 1.4 for precise definition] resulting in the operator $T_{\mathcal{S}}$, given as follows*

$$\mathbf{u}(x,y) \cong (T_{\mathcal{S}}(\mathbf{u}))(x,y) = \sum_{i\in\mathfrak{C}} \phi_i\Big(\mathcal{S}\big(\mathbf{u}(w_i^{-1}(x,y))\big)\Big). \tag{4.77}$$

Then $T_{\mathcal{S}}$ is a contraction on $\ell^2(\Upsilon)$ and a necessary and sufficient contractivity condition is

$$\max_{j\in\mathfrak{P}}\Big\{\sqrt{\sum_{i\in\mathfrak{C}_j}{\alpha_i}^2}\Big\} < 1. \tag{4.78}$$

In addition, the contractivity factor of $T_{\mathcal{S}}$ is

$$con_{T_{\mathcal{S}}} = \max_{j\in\mathfrak{P}}\Big\{\sqrt{\sum_{i\in\mathfrak{C}_j}{\alpha_i}^2}\Big\}. \tag{4.79}$$

Proof. The proof is very similar to the previous proofs. This is especially because we can show for any $\mathbf{y}\in\ell^2(\Upsilon)$,

$$\Big\|\mathcal{S}\big(\mathbf{y}|_{P_{J(i)}}\big)\Big\| \le \Big\|\mathbf{y}|_{P_{J(i)}}\Big\|, \tag{4.80}$$

as opposed to

$$\Big\|\mathcal{D}\big(\mathbf{y}|_{P_{J(i)}}\big)\Big\| \le \frac{1}{s}\Big\|\mathbf{y}|_{P_{J(i)}}\Big\|. \tag{4.81}$$

Hence, the factor s is eliminated from the expressions in the corresponding propositions. Furthermore, similar arguments lead to the fact that $T_{\mathcal{S}}$ is a contraction in this case if

$$\max\{|\alpha_i|\}_{i\in\mathfrak{C}} < \frac{1}{\sqrt{\mathfrak{C}_{max}}}. \tag{4.82}$$

Apparently this suggests that it would be safer to use $\mathcal{D}$ than $\mathcal{S}$ in terms of convergence results. □

4.4 Convergence Results for the Fractal Transform Operator

The propositions proved in the previous sections of this chapter may give the incorrect impression that the contractivity of T is necessary for the convergence of the fractal decoding scheme. In this section, we derive analytical results on the convergence of the fractal decoding. It turns out that the fractal decoding scheme may be convergent to a unique limit regardless of the starting point, under certain conditions even when T is not necessarily a contraction.

Given T, we can define a corresponding $\underline{T}$ acting on $\underline{\mathbf{x}}$ such that $\underline{T}(\underline{\mathbf{x}})$ is $T(\mathbf{x})$ represented in the vector format, as done earlier in Section 3.3.4. We also take advantage of writing $\underline{\mathbf{x}}$ in the vector format, so that operator $\underline{T}$ can be written as

$$\underline{T}(\underline{\mathbf{x}}) = \mathbf{M}\underline{\mathbf{x}} + \mathbf{B}. \tag{4.83}$$

As mentioned earlier in Section 3.3.4, a similar type of matrix representation of the fractal transform operator was introduced in [Ham98]. Here, $\mathbf{M}$ is an $N^2 \times N^2$ matrix and both $\underline{\mathbf{x}}$ and $\mathbf{B}$ are vectors of dimension $N^2 \times 1$. Matrix $\mathbf{M}$ carries the α_i information, and $\mathbf{B}$ contains the information of β_i, all in the appropriate locations. Recall that α_i and β_i were the parameters of the greyscale maps.

We would like to find conditions under which the iterations $\mathbf{u}_{n+1} = T(\mathbf{u}_n)$ converge regardless of the initial arbitrary image $\mathbf{u}_0$. In the vector-matrix notation, the iterative scheme reads

$$\underline{\mathbf{u}}_{n+1} = \mathbf{M}\underline{\mathbf{u}}_n + \mathbf{B}. \tag{4.84}$$

The following proposition is a known general result for linear iterative schemes.

Proposition 4.4.1. *[You71, Ham98] The linear iterative scheme in Equation 4.84 converges to a limit independent of $\underline{\mathbf{u}}_0$ if and only if $\rho(\mathbf{M}) < 1$, where $\rho(\mathbf{M})$ represents the spectral radius of $\mathbf{M}$.*

The following proposition proved in [Ham98] gives a very useful sufficient condition for the convergence of the fractal decoding scheme.

Proposition 4.4.2. *[Ham98] A sufficient condition for the convergence of the fractal decoding scheme $\mathbf{u}_{n+1} = T(\mathbf{u}_n)$ regardless of the starting initial arbitrary image is that $|\alpha_i| < 1$ for all $i \in \mathfrak{C}$.*

Proof. Consider the decoding scheme in the vector-matrix notation. The necessary and sufficient condition for the convergence of the fractal decoding scheme was $\rho(\mathbf{M}) < 1$. Also, $\rho(\mathbf{M}) \leq \|\mathbf{M}\|$, where $\|.\|$ is any matrix norm [You71]. Therefore, a sufficient condition for the convergence of the fractal decoding scheme would be $\|\mathbf{M}\|_{\infty-norm} < 1$. Noting that each row of $\mathbf{M}$ contains exactly s^2 nonzero elements of value $\frac{\alpha_i}{s^2}$ for some $i \in \mathfrak{C}$, this shows that the infinity norm of $\mathbf{M}$ is $\max_{i\in\mathfrak{C}}\{|\alpha_i|\}$. Hence, if $|\alpha_i| < 1$ for all $i \in \mathfrak{C}$, the fractal decoding scheme converges to a limit regardless of the starting initial arbitrary image. □

Although this sufficient condition is very useful, it is not necessary at all for the convergence of the fractal decoding scheme. It can also be shown that the sufficient condition and the necessary and sufficient condition we found for the contractivity of T in the previous section do not necessarily contradict each other.

Therefore, the concluding point is that the convergence of the fractal decoding scheme to a limit regardless of the starting image is given by the condition $\rho(\mathbf{M}) < 1$, which is not necessarily in alignment with the contractivity of T. The fact that the fractal decoding process converges to a limit for many cases, even if none of the conditions $\max_{i\in\mathfrak{C}}\{|\alpha_i|\} < 1$ and $\max_{j\in\mathfrak{P}}\left\{\sqrt{\sum_{i\in\mathfrak{C}_j}(\frac{\alpha_i}{s})^2}\right\} < 1$ is satisfied, lies in the fact that $\rho(\mathbf{M}) < 1$ for the associated fractal code.

It would be interesting to verify the fact that we can generalize the Collage Theorem (Equation 3.20) to include cases for which the fractal decoding scheme is converging to a limit although T is not a contraction in those cases.

Proposition 4.4.3. *The Generalized Collage Theorem*
Assume that the sequence $\mathbf{u}_{n+1} = T(\mathbf{u}_n)$ *converges to* $\mathbf{u}^\star$ *regardless of* $\mathbf{u}_0$*, i.e.,* $\rho(\mathbf{M}) < 1$*. Then*

$$\|\mathbf{u} - \mathbf{u}^\star\| \leq \|\mathbf{u} - T(\mathbf{u})\| \cdot \|(\mathbf{I}_{N^2 \times N^2} - \mathbf{M})^{-1}\|_{2-norm}, \tag{4.85}$$

where $\|.\|_{2-norm}$ *represents the matrix* 2*-norm.*

Proof. Considering the fractal transform operator in matrix notation, and using induction for any $k \in \mathbb{N}$, yields

$$\underline{T}^k(\underline{\mathbf{u}}) = \mathbf{M}^k \underline{\mathbf{u}} + (\sum_{m=0}^{k-1} \mathbf{M}^m)\mathbf{B}. \tag{4.86}$$

Hence for any $k \in \mathbb{N}$

$$\begin{aligned}
\|\underline{\mathbf{u}} - \underline{T}^k(\underline{\mathbf{u}})\| &= \|\underline{\mathbf{u}} - \mathbf{M}^k \underline{\mathbf{u}} - (\sum_{m=0}^{k-1} \mathbf{M}^m)\mathbf{B}\| && (4.87)\\
&= \|\sum_{m=0}^{k-1} (\mathbf{M}^m - \mathbf{M}^{m+1})\underline{\mathbf{u}} - (\sum_{m=0}^{k-1} \mathbf{M}^m)\mathbf{B}\| && (4.88)\\
&= \|\sum_{m=0}^{k-1} \mathbf{M}^m \big[(\mathbf{I}_{N^2 \times N^2} - \mathbf{M})\underline{\mathbf{u}} - \mathbf{B}\big]\| && (4.89)\\
&\leq \|(\mathbf{I}_{N^2 \times N^2} - \mathbf{M})\underline{\mathbf{u}} - \mathbf{B}\| \cdot \|\sum_{m=0}^{k-1} \mathbf{M}^m\|_{2-norm}. && (4.90)
\end{aligned}$$

If $\rho(\mathbf{M}) < 1$, then the limit of $\sum_{m=0}^{k-1} \mathbf{M}^m$ as $k \to \infty$ exists and equals $(\mathbf{I}_{N^2 \times N^2} - \mathbf{M})^{-1}$ [NS82]. Hence, taking the limit $k \to \infty$ and converting back to the non-vector notation yields

$$\|\mathbf{u} - \mathbf{u}^\star\| \leq \|\mathbf{u} - T(\mathbf{u})\| \cdot \|(\mathbf{I}_{N^2 \times N^2} - \mathbf{M})^{-1}\|_{2-norm}. \tag{4.91}$$

□

Proposition 4.4.4. *The Generalized Collage Theorem in Proposition 4.4.3, generalizes the Collage Theorem in Equation 3.20.*

Proof. First we show that if T is a contraction with contractivity factor $con_{(T)}$, $0 \leq con_{(T)} < 1$, then $\|\mathbf{M}\|_{2-norm} \leq con_{(T)}$. Note that for any $\underline{\mathbf{x}} \neq \underline{\mathbf{0}}$

$$\|\mathbf{M}\underline{\mathbf{x}}\| = \|\mathbf{M}\underline{\mathbf{x}} - \mathbf{M}\underline{\mathbf{0}}\| = \|\mathbf{M}\underline{\mathbf{x}} + \mathbf{B} - (\mathbf{M}\underline{\mathbf{0}} + \mathbf{B})\| = \|\underline{T}(\underline{\mathbf{x}}) - \underline{T}(\underline{\mathbf{0}})\| \leq con_{(T)}\|\underline{\mathbf{x}}\|. \tag{4.92}$$

Therefore, for any $\underline{\mathbf{x}} \neq \underline{\mathbf{0}}$

$$\frac{\|\mathbf{M}\underline{\mathbf{x}}\|}{\|\underline{\mathbf{x}}\|} \leq con_{(T)}. \tag{4.93}$$

and

$$\|\mathbf{M}\|_{2-norm} = \sup_{\underline{\mathbf{x}} \neq \underline{\mathbf{0}}} \frac{\|\mathbf{M}\underline{\mathbf{x}}\|}{\|\underline{\mathbf{x}}\|} \leq con_{(T)}. \tag{4.94}$$

Hence,

$$\begin{aligned} \|(\mathbf{I}_{N^2\times N^2} - \mathbf{M})^{-1}\|_{2-norm} &= \|\sum_{m=0}^{\infty} \mathbf{M}^m\|_{2-norm} && (4.95)\\ &\leq \sum_{m=0}^{\infty} \|\mathbf{M}\|_{2-norm}^m && (4.96)\\ &\leq \sum_{m=0}^{\infty} con_{(T)}^m && (4.97)\\ &= \frac{1}{1-con_{(T)}}. && (4.98) \end{aligned}$$

Using this inequality in the Generalized Collage Theorem in Proposition 4.4.3 yields the collage Theorem

$$\|\mathbf{u} - \mathbf{u}^\star\| \leq \frac{1}{1-con_{(T)}} \cdot \|\mathbf{u} - T(\mathbf{u})\|. \quad (4.99)$$

□

In the case that T is a contraction a smaller upper-bound on $\|\mathbf{u} - \mathbf{u}^\star\|$ is given by the above proposition.

4.5 Fractal Code with the Equal Range and Domain Block Size

Recall from Chapter 3, that the fractal transform operator T was defined in terms of domain and range partitions, and domain-to-range mappings. We also mentioned that the domain blocks of size $sK \times sK$ are typically larger than the range blocks of size $K \times K$. Here, we would like to emphasize that it is also possible to construct a fractal transform for the case that $s = 1$. Because we are not interested in compression capabilities of the fractal transform, this seems to be a rather interesting case to consider.

Note that if $s = 1$, for each $i \in \mathfrak{C}$ and $(x, y) \in \Upsilon$, the *block mapping* w_i in Equation 3.17 simply becomes

$$w_i(P_{J(i)}(x, y) = C_i(x, y). \quad (4.100)$$

In our setting, we did not necessarily choose mappings w_i to be contractive. It may be a wrong convention among fractal imaging experts who think w_i should necessarily be contractive to yield a contractive T.

If, during the coding stage, i.e., the search for the "best" domain block, we allow each range block be approximated by any domain block, we arrive at a trivial case: the case that each domain block approximates itself. An interesting situation arises if we do not allow this situation. Hence, we exclude the range block being approximated from the search pool. Interestingly, all of the contractivity and convergence results we considered in this chapter are still valid in our setting for this case where $s = 1$.

4.6 Concluding Remarks

In this chapter, new contractivity and convergence conditions for the fractal transform operator were developed. We addressed a number of open questions in the area.

Chapter 5

Generalized Fractal Image Coding as Projections onto Convex Sets

5.1 Introduction

In this chapter, we show how fractal image coding can be viewed and generalized in terms of the method of projections onto convex sets (POCS). In our approach, the fractal code is considered as defining a set of spatial domain similarity constraints. In this framework, we also show how POCS can provide the opportunity to apply the fractal code of an image along with additional constraints at the decoding stage. The mathematical foundation of this approach along with some interesting applications are presented.

As mentioned earlier, in traditional fractal-based methods, images are thought of as elements of a complete metric space, e.g., $\ell^2(\Upsilon)$, for $\Upsilon = [1, \ldots, N] \times [1, \ldots, N]$. A contractive operator T is estimated at the coding stage, such that its fixed point is an approximation of the original image. At the decoding stage, this contractive operator or *fractal transform* is applied iteratively to an initial "seed" image. Banach's contraction mapping theorem guarantees the convergence of the iterative scheme to the fixed point of T. This decoding process is too restrictive, however, since additional knowledge about the original image (e.g., bounding constraints on the pixel values or smoothness along block boundaries) cannot be applied simultaneously with the fractal transform. As a result, post-processing algorithms have traditionally been applied to the fixed point of the contraction map T in order to incorporate additional constraints. This post-processing stage usually translates to applying some projections on the fixed point. For example, thresholding the data in order to enforce certain bounds is simply a projection of the data onto the space of images that lie within those bounds.

In our approach, a fractal code is considered as a set of spatial domain similarity constraints. Fortunately these constraints are convex and closed in $\ell^2(\Upsilon)$ and therefore satisfy the requirement of setting up a POCS model.

In this framework, we indicate how POCS can provide the opportunity to apply the fractal code of an image along with additional constraints at the decoding stage. These constraints may include other spatial, statistical, spectral and pattern properties of the unknown image, e.g., bounds on pixel values, bounded energy, smoothness, similarity to the observed data, etc.

A feature of this interpretation is that we can move away from the traditional view of fractal coding as simply a mapping of domain image subblocks onto range image subblocks. Instead, we consider the application of the projections on a closed and convex set that is constructed using the similarity constraint between the domain and range subblocks. This type of projection translates to a *simultaneous* alteration of domain-range block pairs. In this chapter, we shall introduce such projections explicitly and study some of their interesting implications.

Let us qualify that we are not interested in the compression capabilities of fractal image coding here. Our study picks up on the thesis of T. Puh on set-theoretic coding [Puh96]. In that work, there were no explicit details on how the self-similarity properties of an image can be translated into POCS-type inequality constraints.

The layout of the remainder of this chapter is as follows. The formulation of the fractal-based self-similarity constraints is presented in Sections 5.2 and 5.3. In Section 5.4, an application of

this formulation will be presented as the solution to the problem of image reconstruction given a missing fractal code. In Section 5.5, the potential application of the prescribed method in image denoising is discussed. Finally, some concluding remarks will be given in Section 5.6.

5.2 Self-Similarity Constraints Using Collage Distances

In this section, we consider a number of self-similarity constraints on image blocks which relate fractal image coding to the POCS concept. Collage distance is an important measure of self-similarity in this context. Below we consider three different types of collage distances, using which similarity constraints can be built. These are (a) total collage distance, (b) collage distance for fixed domain-range blocks, and (c) point-wise collage distance.

Definition 5.2.1. *For a given image* $\mathbf{u} \in \ell^2(\Upsilon)$*, we introduce the following collage distance definitions.*
a) The total collage distance *is defined as*

$$\Delta(\mathbf{u}) = \Big\| T(\mathbf{u}) - \mathbf{u} \Big\| \tag{5.1}$$

b) Given any $i \in \mathfrak{C}$*, the collage distance for a fixed pair, range block* C_i *and domain block* $P_{J(i)}$*, is*

$$\begin{aligned} \Delta^{(i)}(\mathbf{u}) &= \Big\| \phi_i\Big(\mathcal{D}\Big(\mathbf{u}\big(w_i^{-1}(C_i)\big)\Big)\Big) - \mathbf{u}(C_i) \Big\| & (5.2)\\ &= \Big\| \phi_i\Big(\mathcal{D}\Big(\mathbf{u}\big(P_{J(i)}\big)\Big)\Big) - \mathbf{u}(C_i) \Big\|, & (5.3) \end{aligned}$$

where, as before, w_i^{-1} *is the inverse mapping from* $P_{J(i)}$ *to* C_i*.*
c) Given any $i \in \mathfrak{C}$ *and a pixel* $(m,n) \in C_i$*, the pointwise collage distance is defined as*

$$\Delta^{(i)}_{(m,n)}(\mathbf{u}) = \Big| \phi_i\Big(\mathcal{D}\Big(\mathbf{u}\big(w_i^{-1}(m,n)\big)\Big)\Big) - \mathbf{u}(m,n) \Big|. \tag{5.4}$$

Although fractal image coding relies on perfect self-similarity between elements in the range and domain pool, in natural images the assumption of perfect self-similarity does not hold. Therefore, to design a model that is more consistent with the physical world we enforce a criterion that the collage distances as measures of similarity must lie within some threshold. This threshold can be determined based on the application. This framework allows more flexibility to incorporate additional knowledge about the image and to relax the condition of perfect self-similarity in order to get a better approximation of an image in the reconstruction process. That being said, our model can be made consistent with the perfect self-similarity assumption case, by setting the threshold to be zero. This translates to the fact that our model is in complete consistency with traditional fractal image coding.

We may use each of the three collage distances described above to define the following associated self-similarity constraints in $\ell^2(\Upsilon)$.

a) Based on the total collage distance define

$$\Psi = \Big\{ \mathbf{u} \in \ell^2(\Upsilon) : \Delta(\mathbf{u}) \leq \delta \Big\}. \tag{5.5}$$

b) Based on range based collage distance define

$$\Psi^{(i)} = \Big\{\mathbf{u} \in \ell^2(\Upsilon) : \Delta^{(i)}(\mathbf{u}) \leq \delta^{(i)}\Big\}. \tag{5.6}$$

c) Finally, define the set $\Psi^{(i)}_{(m,n)}$ based on the pointwise collage distance as

$$\Psi^{(i)}_{(m,n)} = \Big\{\mathbf{u} \in \ell^2(\Upsilon) : \Delta^{(i)}_{(m,n)}(\mathbf{u}) \leq \delta^{(i)}_{(m,n)}\Big\}. \tag{5.7}$$

5.3 Discrete Pointwise Collage Constraints and Associated Projections

Note that we defined $\Psi^{(i)}_{(m,n)}$, where (m,n) is a pixel with integer coordinates over a fixed C_i. We base our model on the sets $\Psi^{(i)}_{(m,n)}$ constructed by the pointwise collage distances.

$$\begin{aligned}\Psi^{(i)}_{(m,n)} &= \Big\{\mathbf{u} \in \ell^2(\Upsilon) : \Delta^{(i)}_{(m,n)}(\mathbf{u}) \leq \delta^{(i)}_{(m,n)}\Big\} \\ &= \Big\{\mathbf{u} \in \ell^2(\Upsilon) : \Big|\phi_i\Big(\mathcal{D}\Big(\mathbf{u}\big(w_i^{-1}(m,n)\big)\Big)\Big) - \mathbf{u}(m,n)\Big| \leq \delta^{(i)}_{(m,n)}\Big\}.\end{aligned}$$

Assuming that the domain blocks are twice as large as the range blocks in each direction then, in our discrete setting, (m,n) corresponds to four pixels (p,q), $(p,q+1)$, $(p+1,q)$, and $(p+1,q+1)$ under the inverse mapping $w_i^{-1}(m,n)$, i.e.,

$$w_i^{-1}(m,n) = \{(p,q),(p,q+1),(p+1,q),(p+1,q+1)\}. \tag{5.8}$$

Furthermore, in this case $\mathcal{D}\Big(\mathbf{u}\big(w_i^{-1}(m,n)\big)\Big)$ can be written as

$$\mathcal{D}\Big(\mathbf{u}\big(w_i^{-1}(m,n)\big)\Big) = \frac{1}{4}\Big\{\mathbf{u}(p,q) + \mathbf{u}(p,q+1) + \mathbf{u}(p+1,q) + \mathbf{u}(p+1,q+1)\Big\}.$$

Hence, replacing $\phi_i(t) = \alpha_i t + \beta_i$ yields

$$\begin{aligned}\Psi^{(i)}_{(m,n)} &= \Big\{\mathbf{u} \in \ell^2(\Upsilon) : \Big|\frac{\alpha_i}{4}\big\{\mathbf{u}(p,q) + \mathbf{u}(p,q+1) + \mathbf{u}(p+1,q) \\ &+ \mathbf{u}(p+1,q+1)\big\} + \beta_i - \mathbf{u}(m,n)\Big| \leq \delta_{(i,j)(m,n)}\Big\}.\end{aligned} \tag{5.9}$$

Proposition 5.3.1. *$\Psi^{(i)}_{(m,n)}$ is a closed and convex set in $\ell^2(\Upsilon)$ provided that $\phi_i(t) = \alpha_i t + \beta_i$.*

Proof. To prove the convexity it is enough to show that for any real number $0 \leq \lambda \leq 1$, the convex combination $(\lambda\mathbf{u} + (1-\lambda)\mathbf{v}) \in \Psi^{(i)}_{(m,n)}$, provided that $\mathbf{u} \in \Psi^{(i)}_{(m,n)}$ and $\mathbf{v} \in \Psi^{(i)}_{(m,n)}$. Note that both $\mathcal{D}$ and ϕ_i are linear, hence

$$\begin{aligned}&\Big|\phi_i\Big(\mathcal{D}\Big((\lambda\mathbf{u} + (1-\lambda)\mathbf{v})\big(w_i^{-1}(m,n)\big)\Big)\Big) - (\lambda\mathbf{u} + (1-\lambda)\mathbf{v})(m,n)\Big| \\ =\ &\Big|\lambda\Big[\phi_i\Big(\mathcal{D}\Big(\mathbf{u}\big(w_i^{-1}(m,n)\big)\Big)\Big) - \mathbf{u}(m,n)\Big] + (1-\lambda)\Big[\phi_i\Big(\mathcal{D}\Big(\mathbf{v}\big(w_i^{-1}(m,n)\big)\Big)\Big) - \mathbf{v}(m,n)\Big]\Big| \\ \leq\ &\lambda\Big|\phi_i\Big(\mathcal{D}\Big(\mathbf{u}\big(w_i^{-1}(m,n)\big)\Big)\Big) - \mathbf{u}(m,n)\Big| + (1-\lambda)\Big|\phi_i\Big(\mathcal{D}\Big(\mathbf{v}\big(w_i^{-1}(m,n)\big)\Big)\Big) - \mathbf{v}(m,n)\Big| \\ \leq\ &\lambda\delta^{(i)}_{(m,n)} + (1-\lambda)\delta^{(i)}_{(m,n)} = \delta^{(i)}_{(m,n)}.\end{aligned} \tag{5.10}$$

Hence, $(\lambda\mathbf{u}+(1-\lambda)\mathbf{v})\in\Psi^{(i)}_{(m,n)}$ which shows the convexity of $\Psi^{(i)}_{(m,n)}$.

To prove the closedness of $\Psi^{(i)}_{(m,n)}$ in $\ell^2(\Upsilon)$, assume that the sequence of images $\mathbf{u}^{(k)}\in\Psi^{(i)}_{(m,n)}$ converges to $\mathbf{u}$ in $\ell^2(\Upsilon)$ as $k\to\infty$. The condition $\mathbf{u}^{(k)}\in\Psi^{(i)}_{(m,n)}$ implies that for any $k\in\mathbb{N}$,

$$\Big|\phi_i\Big(\mathcal{D}\Big(\mathbf{u}^{(k)}\big(w_i^{-1}(m,n)\big)\Big)\Big)-\mathbf{u}^{(k)}(m,n)\Big|\leq\delta^{(i)}_{(m,n)}. \tag{5.11}$$

The operators $\mathcal{D}$ and ϕ are both linear and continuous, the absolute value is also a continuous function. Hence, taking the limit of the left-hand side for $k\to\infty$ is meaningful. Furthermore, the limit of the left-hand side can not be larger than $\delta^{(i)}_{(m,n)}$. Therefore, taking the limit $k\to\infty$ yields

$$\Big|\phi_i\Big(\mathcal{D}\Big(\mathbf{u}\big(w_i^{-1}(m,n)\big)\Big)\Big)-\mathbf{u}(m,n)\Big|\leq\delta^{(i)}_{(m,n)}, \tag{5.12}$$

i.e., $\mathbf{u}\in\Psi^{(i)}_{(m,n)}$. Therefore, $\Psi^{(i)}_{(m,n)}$ is closed in $\ell^2(\Upsilon)$. □

Now that the convex and closed pointwise similarity constraints are constructed, so the projection operators on these sets are meaningful. If we find the projection operators on these sets we will be able to set-up our POCS-based framework.

The following proposition provides the explicit form of the projection operator on $\Psi^{(i)}_{(m,n)}$ given an arbitrary $\mathbf{u}\in\ell^2(\Upsilon)$.

Proposition 5.3.2. *Assume an image $\mathbf{u}\in\ell^2(\Upsilon)$ is given. Let $\mathbf{v}$ be the projection of $\mathbf{u}$ on $\Psi^{(i)}_{(m,n)}$. This projection can be computed in the following manner.*

Assume that the four pixels in the set

$$\Lambda=\Big\{(p,q),(p,q+1),(p+1,q),(p+1,q+1)\Big\} \tag{5.13}$$

are the ones mapped from (m,n) under the inverse mapping w_i^{-1}. Take $\delta=\delta^{(i)}_{(m,n)}\geq 0$, $\beta=\beta_i$, $\alpha=\alpha_i$, and

$$r=\frac{\alpha}{4}\Big[\mathbf{u}(p,q)+\mathbf{u}(p,q+1)+\mathbf{u}(p+1,q)+\mathbf{u}(p+1,q+1)\Big]+\beta-\mathbf{u}(m,n). \tag{5.14}$$

In the case that $(m,n) \notin \Lambda$*, the values* $\mathbf{v}(x,y)$ *for* $(x,y) \in \Upsilon$ *can be computed as*

$$\mathbf{u}(x,y) + \begin{cases} \textit{Case } (x,y) \in \Lambda \\ \frac{-\alpha/4}{1+4(\alpha/4)^2}(r-\delta), & r > \delta \\ 0, & |r| \leq \delta \\ \frac{-\alpha/4}{1+4(\alpha/4)^2}(r+\delta), & r < -\delta \\ \\ \textit{Case } (x,y) = (m,n) \\ \frac{1}{1+4(\alpha/4)^2}(r-\delta), & r > \delta \\ 0, & |r| \leq \delta \\ \frac{1}{1+4(\alpha/4)^2}(r+\delta), & r < -\delta \\ \\ \textit{Case } (x,y) \notin \Lambda \cup \{(m,n)\} \\ 0. \end{cases} \tag{5.15}$$

In the case that $(m,n) \in \Lambda$*, the values* $\mathbf{v}(x,y)$ *for* $(x,y) \in \Upsilon$ *can be computed as*

$$\mathbf{u}(x,y) + \begin{cases} \textit{Case } (x,y) \in \Lambda \setminus \{(m,n)\} \\ \frac{-\alpha/4}{(1-\alpha/4)^2+3(\alpha/4)^2}(r-\delta), & r > \delta \\ 0, & |r| \leq \delta \\ \frac{-\alpha/4}{(1-\alpha/4)^2+3(\alpha/4)^2}(r+\delta), & r < -\delta \\ \\ \textit{Case } (x,y) = (m,n) \\ \frac{1-\alpha/4}{(1-\alpha/4)^2+3(\alpha/4)^2}(r-\delta), & r > \delta \\ 0, & |r| \leq \delta \\ \frac{1-\alpha/4}{(1-\alpha/4)^2+3(\alpha/4)^2}(r+\delta), & r < -\delta \\ \\ \textit{Case } (x,y) \notin \Lambda \\ 0. \end{cases} \tag{5.16}$$

Proof. One strategy is to prove the proposition directly by setting a constrained optimization problem and solving using the Karush-Kuhn-Tucker (KKT) conditions [Kar39, KT51] for solving the optimization problems with inequality constraints. Another indirect approach is to make use of a result of a paper [OTS94] on deblurring of a signal in presence of space varying blur. We ignore the difference between the two contexts and simply state and employ the result in the paper. Below we have slightly modified the notations in the result of [OTS94], to match with

ours. In [OTS94], it is stated that given

$$r = \mathbf{g}(m,n) - \sum_x \sum_y \mathbf{u}(m,n)\mathbf{h}(m,n;x,y), \tag{5.17}$$

and the set $\Psi_{m,n} = \{\mathbf{u}(x,y) : |r| \leq \delta\}$, the projection of a given $\mathbf{u}(x,y)$ on $\Psi_{m,n}$ is computed by

$$\mathbf{u}(x,y) + \begin{cases} \frac{\mathbf{h}(m,n;x,y)}{\sum_k \sum_l \mathbf{h}^2(m,n;k,l)}(r-\delta), & r > \delta \\ 0, & |r| \leq \delta \\ \frac{\mathbf{h}(m,n;x,y)}{\sum_k \sum_l \mathbf{h}^2(m,n;k,l)}(r+\delta), & r < -\delta \end{cases} \tag{5.18}$$

Back to our proposition, if we assume $\mathbf{g}(m,n) = \beta$, and $\mathbf{h}(m,n;k,l)$ is constructed appropriately based on the α's then, we may readily obtain the projection based on this result.

In the first case that $(m,n) \notin \Lambda$, we need to define $\mathbf{h}$ in the following fashion, which would produce the expression r in our problem.

$$\mathbf{h}(m,n;k,l) = \begin{cases} 1 & \textit{Case } (k,l) = (m,n) \\ (-\alpha/4) & \textit{Case } (k,l) \in \Lambda \\ 0 & \textit{Case } (k,l) \notin \Lambda \cup \{(m,n)\} \end{cases} \tag{5.19}$$

Now, we consider three cases of (i) $(x,y) \in \Lambda$, (ii) $(x,y) = (m,n)$, and (iii) $(x,y) \notin \Lambda \cup \{(m,n)\}$. Substituting the expressions of Equation 5.19 in Equation 5.18 yields our formula in Equation 5.15.

Similarly, in the second case that $(m,n) \in \Lambda$, we define $\mathbf{h}$ in the following fashion

$$\mathbf{h}(m,n;k,l) = \begin{cases} (1-\alpha/4) & \textit{Case } (k,l) = (m,n) \\ (-\alpha/4) & \textit{Case } (k,l) \in \Lambda \setminus \{(m,n)\} \\ 0 & \textit{Case } (k,l) \notin \Lambda \end{cases} \tag{5.20}$$

This time, considering three possible cases of (i) $(x,y) \in \Lambda \setminus \{(m,n)\}$, (ii) $(x,y) = (m,n)$, and (iii) $(x,y) \notin \Lambda$, and substituting the expressions of Equation 5.20 in Equation 5.18 yields our formula in Equation 5.16. This simple observation completes the proof of the proposition. □

In Figure 5.1, each of the top and bottom images shows a mapping of a domain to a range block. In each image, as an example, we have indicated four pixels in Λ which are mapped to a pixel (m,n). The only difference is that in the image on top the range block and domain block are disjoint while in the bottom image the range block is included in the domain block. In Figure 5.2 , the mapping from the four pixels in Λ to the pixel (m,n) is indicated for two cases where $(m,n) \notin \Lambda$ in the top image and $(m,n) \in \Lambda$ in the bottom image. Figures 5.3 and 5.4 demonstrate an example of the projection defined in Proposition 5.3.2 for the two cases $(m,n) \notin \Lambda$ and $(m,n) \in \Lambda$ respectively.

Now that the constraints $\Psi^{(i)}_{(m,n)}$ and their associated projections are in hand we may apply these constraints in the POCS iterations along with other consistent constraints at the decoding stage.

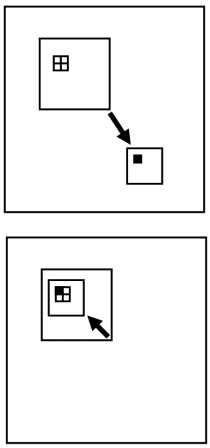


Figure 5.1: Top: Mapping from a domain block to a range block, when these blocks are non-overlapping. Bottom: The chosen domain and range blocks are overlapping. The four by four square and a black square indicates a pointwise correspondence.

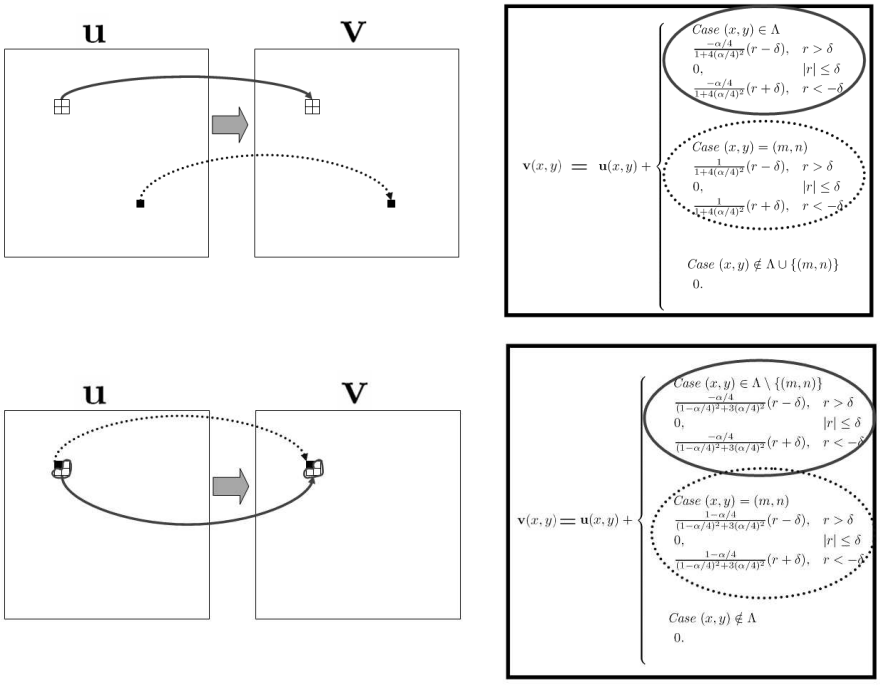


Figure 5.2: Top: Pointwise mapping corresponding to the $(m,n) \notin \Lambda$ case. Bottom: Pointwise mapping corresponding to the $(m,n) \in \Lambda$ case.

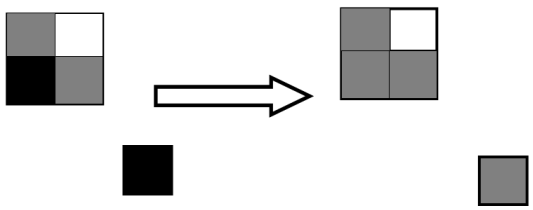


Figure 5.3: Sample projection for a pointwise collage constraint from $\mathbf{u}$ to $\mathbf{v}$ for the case where $(m, n) \notin \Lambda$ case. In this figure, greyscales represent the intensity values of each pixel. The intensity of the rest of the pixels under this projection remains unchanged. As indicated only five pixels are involved.

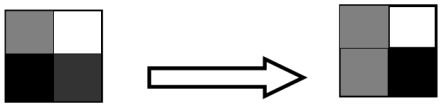


Figure 5.4: Sample projection for a pointwise collage constraint from $\mathbf{u}$ to $\mathbf{v}$ for the case where $(m, n) \in \Lambda$ case. In this figure, greyscales represent the intensity values of each pixel. The intensity of the rest of the pixels under this projection remains unchanged. As indicated only four pixels are involved.

5.4 An Application to Partially Known Fractal Code Problem

In this section, we present a problem of image recovery given a partially known fractal code as a motivating example for the theory described above.

Assume that from a given fractal code of an image, part of the range-domain block assignments along with their corresponding coefficients are missing. In this case, the usual fractal decoding scheme employing an arbitrary "seed" image will collapse since the range blocks of the image for which the fractal code is missing cannot be modified. These blocks will simply remain stationary, identical to the corresponding subblocks of the seed image. To illustrate, assume that the fractal codes corresponding to all range blocks in the bottom half of an image are missing. Figures 5.5(b), 5.6(b), and 5.7(b) show the results of this situation. In the usual fractal decoding scheme, the mapping of domain onto range blocks is performed only for the top-half range blocks. The difference between the results is due to the difference in seed images: purely black images for Figures 5.5(b) and 5.7(b), and random for Figure 5.6(b).

The problem of missing code is, in fact, a problem which is underdetermined, i.e., there are many possible solutions. This gives the opportunity of using POCS over the constraints $\Psi^{(i)}_{(m,n)}$, for $\delta = 0$, along with an additional prior, for example, extra smoothness, that will yield an approximation of the original image.

Figure 5.5(c) shows the result of POCS using the constraints $\Psi^{(i)}_{(m,n)}$ for all available range indices i and all image pixels (m, n), along with one extra smoothness constraint. Here, we assume such constraint defined using the set Ψ_b, the set of all images $\mathbf{u} \in \ell^2(\Upsilon)$ that are bandlimited to a certain bandwidth b, i.e.,

$$\Psi_b = \{\mathbf{u} \in \ell^2(\Upsilon) : |DFT(\mathbf{u})(u, v)| = 0, \text{for any } u, v \in \mathbb{Z}, \text{that } |u| \leq \frac{N}{2},\ |v| \leq \frac{N}{2},\ \sqrt{u^2 + v^2} \geq bN\}. \tag{5.21}$$

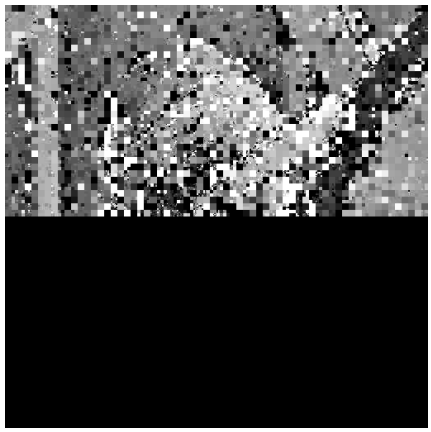
It is well known that the set Ψ_b is closed and convex and the projection onto Ψ_b is the operation of ideal frequency low-pass filtering. This corresponds to some smoothness prior along with the partially known self-similarity constraints. We have used this type of constraint along with the partially known self-similarity constraints to generate Figure 5.5(c).

It turns out that applying other types of smoothing operators, e.g., Gaussian smoothing, also works with the method numerically to produce more desirable results although such smoothing operators are not projection operators in general.

Assuming the same missing fractal code as for Figures 5.6(b) and 5.7(b), Figures 5.6(c) and 5.7(c) are the approximations obtained when the POCS model along with Gaussian smoothing with, respectively, random and black seed images.

The images in Figures 5.6(c) and 5.7(c), obtained by the POCS model along with the extra smoothness constraints are seen to represent a great improvement over Figures 5.6(b) and 5.7(b). More interestingly, the usual fractal decoding scheme of mapping domain to range blocks does not change the missing range blocks at all in the iteration process, i.e., the bottom part of random

(a)

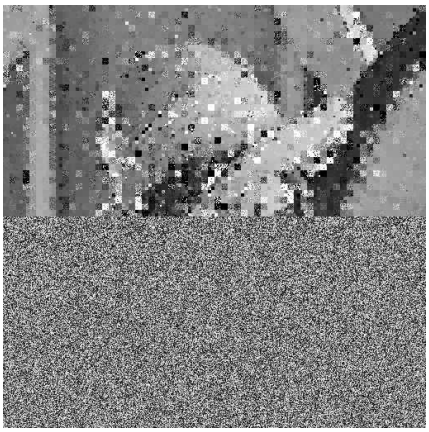


(b)

(c)

Figure 5.5: (a) Lena original. (b) Decoded as bottom half of the fractal code is missing, black seed. (c) Proposed method using POCS with frequency ideal low-pass filtering.

(a)



(b)

(c)

Figure 5.6: (a) Lena original. (b) Decoded as bottom half of the fractal code is missing, random seed. (c) Proposed method using POCS with Gaussian smoothing.

(a)

(b)

(c)

Figure 5.7: (a) Lena attractor 4×4 range block size. (b)Decoded as bottom half of the fractal code is missing, black seed. (c) Proposed method using POCS with Gaussian smoothing.

(a)

(b)

(c)

Figure 5.8: (a) Lena original. (b) Decoded image starting from random seed, a quarter of the fractal code is randomly missing. (c) Proposed method using POCS with Gaussian smoothing.

(a)

(b)

(c)

Figure 5.9: (a) A quarter of the code related to the range blocks in black is missing. (b) Decoded image starting from black seed, a quarter of the fractal code is randomly missing. (c) Proposed method using POCS with Gaussian smoothing.

and black seed images remains unchanged as in Figures 5.6(b) and 5.7(b). But based on the fact that the projections in the POCS model do change the domain and range blocks simultaneously, we get a somewhat reasonably meaningful reconstruction at the domain scale for the bottom-half of these two images. This is possible because of the cross-similarities between the domain blocks in the bottom-half of the image and some range blocks. The more domain blocks that are in the regions of missing range blocks, the better we may approximate the image based on the partial fractal code.

In another experiment we have assumed that part of the fractal code corresponding to a uniformly random distribution of range blocks are missing. These blocks are shown in Figure 5.9(a). Figures 5.8(b) and 5.9(b) show the results of of applying simple fractal decoding with this missing code for the two cases, respectively, (i) random seed image and (ii) black seed image. In Figures 5.8(c) and 5.9(c) we show the results obtained by using the POCS method along with a smoothness prior. Again here, we have used Gaussian smoothing. The improvement in the resulting images is evident.

5.5 Potential Application to Image Denoising

It is well known that subjecting a noisy image to a lossy compression scheme, e.g., JPEG, can produce some denoising of the image. This is also observed in the case of fractal image coding. If a noisy image is fractally coded, with little or no regard for compression, then the attractor produced by the fractal code often represents a significantly denoised version of the original image. (In fact, one can go some steps further and improve this procedure – see [GFV03].)

In such fractal image denoising algorithms, however, one performs the usual iteration procedure after the fractal code is obtained. Once again, regardless of the starting seed image, the procedure converges to the attractor of the fractal transform defined by the code. However, the POCS-based reconstruction algorithm explained earlier in this chapter can provide added flexibility in this procedure. For example, if we choose the input noisy image as the starting point of the iteration, it is possible that the procedure converges to a noise-free image that is closer to the original noisy image.

As discussed earlier, the solution of the POCS-based reconstruction method can also depend on the starting point of the iteration if the solution space is non-unique. A larger solution space for the POCS-based fractal coding method can be produced if the strict similarity constraints are relaxed to inequality constraints. Furthermore, if the original noisy image is chosen to be the starting point of the POCS-based iteration, we are actually using the information in this image in the reconstruction process, even after it has been used in determining the fractal code which, in turn, determines the similarity/inequality constraints. Based on experiments, we believe that by choosing appropriate δ tolerance values in the constraints, one may obtain denoised images that are visually more pleasing because they are "closer" to the original image.

Figure 5.10 presents the result of an experiment that supports this claim. The original and a noisy version of a 256×256 image of Lena are shown in Figures 5.10(a) and 5.10(b). The noisy

Figure 5.10: (a) Lena original. (b) The noisy Image, $\sigma = 15$, PSNR=20.03 dB. (c) The IFS attractor of the noisy image, using 8×8 range blocks, PSNR=24.35 dB. (d) Proposed method using POCS, the starting point is the noisy image, and the same fractal transform with 8×8 range blocks is used, with $\delta = |r|/3$, PSNR=25.28 dB.

image is generated using additive Gaussian white noise with zero mean and standard deviation $\sigma = 15$.

The fractal transform of the noisy image in Figure 5.10(b) corresponding to 8×8 pixel range blocks and 16×16 pixel domain blocks was then computed. Traditional fractal decoding produces the attractor shown in Figure 5.10(c). In spite of its blockiness, this image looks less noisy than the one in Figure 5.10(b).

Using the same fractal code as above, we then applied the following POCS-based reconstruction method. In order to enlarge the solution space we used $\delta = |r|/3$ at every pixel of the image in this experiment, where r is defined in Equation 5.14. Finally, the starting point of the POCS iterations was the original noisy image shown in Figure 5.10(b). The result of this POCS-based procedure is shown in Figure 5.10(d). Although no postprocessing has been employed here, the blockiness plaguing Figure 5.10(c) is not visible as in Figure 5.10(d). As well, it seems that more denoising has been achieved with Figure 5.10(d) than with Figure 5.10(c).

5.6 Concluding Remarks

In this chapter, we have described a reformulation of traditional fractal image decoding in terms of the framework of projections onto convex subsets (POCS). In the case that all the constraints defined by the fractal code are applied to the problem then the solution, defined by the intersection of all constraints, is unique and corresponds to the fixed point of the contractive fractal transform. There is a difference, however, between the POCS method and the fractal coding method regarding the nature of the respective iteration procedures and the convergence toward the solution. In the POCS method, the convergence is accomplished in a set-theoretic framework. The projections associated with the POCS method involve simultaneous alteration of domain-range block pairs, unlike the case of fractal coding in which domain blocks are mapped onto range blocks, which may be viewed as a "greedy process."

The principal advantage of the POCS framework is that it provides the flexibility to incorporate constraints and possibly additional knowledge about the reconstructed image. In this framework, it is also possible to replace the strict equality constraints associated with traditional fractal coding with inequalities that allow similarities to within some designated threshold. The latter represent more feasible and realistic conditions encountered in the real world. And, in this way, the set of feasible solutions is extended.

A POCS-based approach also provides the opportunity of solving underdetermined inverse problems in fractal coding as we have shown for the case of the incomplete fractal code problem.

Finally, recall that if all similarity constraints defined by the fractal code are applied strictly, then the solution is unique, namely, the fixed point of the associated fractal transform operator. If additional constraints are applied in a POCS-based framework, then the entire set of constraints may be inconsistent, i.e., there is no "solution" that satisfies all constraints. In this interesting situation, one may need to employ the more recent POCS formulations for inconsistent feasibility problems as studied by P. Combettes [Com94].

Chapter 6

Regularized Fractal Image Decoding

6.1 Introduction

The goal of this chapter is to present a new recipe for the fractal image decoding process. In this chapter, we explain how fractal-based methods can be combined with regularization schemes, e.g., Tikhonov, Total Variation (TV), in an algebraic fashion. Although the regularization procedure is very common in the context of algebraic image restoration, it has not yet been of directly in the context of fractal-based methods. This implication can improve the quality of the decoded image depending on the regularization function. We develop the theory and apply the standard iterative methods in inverse problems, e.g., steepest descent, to estimate the solutions.

In the previous chapter, we asked the question whether fractal image encoding/decoding can be combined with other extra knowledge about the image, i.e., can we regularize the estimated image? In the framework introduced in this chapter, the problem of fractal image decoding will be modelled as the solution of a linear equation with a sparse coefficient matrix. Due to the structure of the coefficient matrix, the solution of this linear system is unique and coincides with the attractor of the contractive fractal transform T. Traditional fractal decoding finds a solution of this linear system in a greedy fashion. In the proposed technique, we construct a minimization problem for the fractal linear equation along with some regularization function. We present the results of some of experiments that employ regularized fractal decoding to solve the problem of missing fractal codes introduced in the previous chapter.

6.2 Regularization Model of Fractal-Transform Operator

In this section, assume that T is the fractal transform of an observed image $\mathbf{u}$ of size $N \times N$, i.e., $\mathbf{u}$ is approximated by the fixed point $\mathbf{u}^\star$ of T. Also for convenience, assume that the blocks in both domain and range pools are nonoverlapping.

Given T, we can define a corresponding $\underline{T}$ acting on $\underline{\mathbf{x}}$ such that $\underline{T}(\underline{\mathbf{x}})$ equals $T(\mathbf{x})$ represented in the vector format, as we did in Section 3.3.4. By these notations, the result of the traditional fractal image decoding in the vector format is

$$\underline{\mathbf{u}}^\star = \arg\min_{\underline{\mathbf{x}}} \| \underline{\mathbf{x}} - \underline{T}(\underline{\mathbf{x}}) \|^2. \tag{6.1}$$

We also take advantage of writing $\underline{\mathbf{x}}$ in the vector format, so that operator $\underline{T}$ can be written as

$$\underline{T}(\underline{\mathbf{x}}) = \mathbf{M}\underline{\mathbf{x}} + \mathbf{B}. \tag{6.2}$$

As mentioned earlier in Section 3.3.4, a similar type of matrix representation of the fractal transform operator was introduced in [Ham98]. Here, $\mathbf{M}$ is an $N^2 \times N^2$ matrix and both $\underline{\mathbf{x}}$ and $\mathbf{B}$ are vectors of dimension $N^2 \times 1$. Matrix $\mathbf{M}$ carries the α_i information, and $\mathbf{B}$ contains the information of the β_i, all in the appropriate locations. Remember that α_i and β_i are the parameters of the greyscale maps.

By this notation, $\underline{\mathbf{u}}^\star$ is the solution of a linear equation of the form

$$\mathbf{M}\underline{\mathbf{x}} + \mathbf{B} = \underline{\mathbf{x}}, \tag{6.3}$$

or equivalently

$$\mathbf{A}\underline{\mathbf{x}} = \mathbf{B}, \tag{6.4}$$

where $\mathbf{A} = \mathbf{I}_{N^2 \times N^2} - \mathbf{M}$, and $\mathbf{I}_{N^2 \times N^2}$ is the identity matrix of size $N^2 \times N^2$. In the case that the fractal transform is fully known, this linear system has a unique solution $\underline{\mathbf{u}}^\star$, determined by the structure of $\mathbf{A}$. This unique solution is in fact the attractor of the fractal transform T in the vector order, i.e., $\underline{\mathbf{u}}^\star$. It is also true that

$$\mathbf{A}\underline{\mathbf{x}} - \mathbf{B} = \underline{\mathbf{x}} - \underline{T}(\underline{\mathbf{x}}), \tag{6.5}$$

and hence finding the fixed point of the fractal transform operator is equivalent to finding the unique minimizer of the collage error. Therefore, the image can be approximated as the solution of the least squares problem

$$\underline{\mathbf{u}}^\star = \arg\min_{\underline{\mathbf{x}}} \| \mathbf{A}\underline{\mathbf{x}} - \mathbf{B} \|^2. \tag{6.6}$$

Now we consider the new objective function

$$\underline{F_{G,\lambda}}(\underline{\mathbf{x}}) = \| \mathbf{A}\underline{\mathbf{x}} - \mathbf{B} \|^2 + \lambda \underline{G}(\underline{\mathbf{x}}), \tag{6.7}$$

in which $G(\underline{\mathbf{x}})$ is some convex and smooth regularization function and λ is the regularization parameter. The idea is to improve the traditional fractal attractor $\underline{\mathbf{u}}^\star$ by minimizing this new objective function.

This is especially very attractive when the expression $\| \mathbf{A}\underline{\mathbf{x}} - \mathbf{B} \|^2$ does not have a unique minimizer – for example, the missing fractal code problem. However, even when $\| \mathbf{A}\underline{\mathbf{x}} - \mathbf{B} \|^2$ has a unique minimizer one can improve the solution by choosing the appropriate values of λ and the regularizer $\underline{G}(\underline{\mathbf{x}})$. Hence, we define

$$\underline{\mathbf{u}^\star{}_{G,\lambda}} = \arg\min_{\underline{\mathbf{x}}} \underline{F_{G,\lambda}}(\underline{\mathbf{x}}). \tag{6.8}$$

The value $\underline{\mathbf{u}^\star{}_{G,\lambda}}$ can be solved for analytically depending on the regularization function, but the goal is to numerically estimate $\underline{\mathbf{u}^\star{}_{G,\lambda}}$ in general.

In the next section, we present a numerical algorithm, the so-called *regularized fractal decoding algorithm (RFDA)*, to estimate $\underline{\mathbf{u}^\star{}_{G,\lambda}}$.

6.3 Regularized Fractal Decoding Algorithm (RFDA)

Constructing the large sparse matrix $\mathbf{A} = \mathbf{I}_{N^2 \times N^2} - \mathbf{M}$, is computationally inefficient. Matrices $\mathbf{A}$ and $\mathbf{M}$ are not only sparse, but also redundant in the nonzero values. Vector $\mathbf{B}$ is also redundant. The redundancy is based on the fact that the values of α_i and β_i are the same for all of the pixels in some range block. A fixed α_i and β_i can appear many times respectively in $\mathbf{A}$ and $\mathbf{B}$ depending on the size of the range and domain blocks.

In this section, we present an algorithm based on the steepest descent (SD) method, Algorithm 2.6.3, (Chapter 2) to minimize $\underline{F_{G,\lambda}}(\underline{\mathbf{x}})$. Our method is efficient in the sense that it does not explicitly construct the large matrices $\mathbf{A}$ and $\mathbf{B}$.

Because the objective function in Equation 6.7 is smooth and convex, we may use the SD iterations to estimate $\underline{\mathbf{u}^{\star}_{G,\lambda}}$. It turns out that SD is not necessarily the most efficient numerical scheme for this purpose, but it is a simple one.

Using the SD with fixed step-size, Algorithm 2.6.3, the sequence $\{\mathbf{x}_n\}$ defined by

$$\underline{\mathbf{x}}_{n+1} = \underline{\mathbf{x}}_n - \mu \nabla \underline{F_{G,\lambda}}(\underline{\mathbf{x}}), \tag{6.9}$$

converges to an approximation of $\underline{\mathbf{u}^{\star}_{G,\lambda}}$ if μ, the scalar step-size is sufficiently small. The convergence trend could be improved by taking advantage of a variable step-size, i.e., by taking a sequence μ_n, but in this chapter we use only a fixed step-size μ.

Differentiating $\underline{F_{G,\lambda}}(\underline{\mathbf{x}})$, and substituting into 6.9 yields

$$\begin{aligned}
\underline{\mathbf{x}}_{n+1} &= \underline{\mathbf{x}}_n + 2\mu \mathbf{A}^T(\mathbf{B} - \mathbf{A}\underline{\mathbf{x}}_n) - \mu\lambda\nabla \underline{G}(\underline{\mathbf{x}}_n) \\
&= \underline{\mathbf{x}}_n + 2\mu(\mathbf{I}_{N^2\times N^2} - \mathbf{M}^T)(\mathbf{B} - \mathbf{A}\underline{\mathbf{x}}_n) - \mu\lambda\nabla \underline{G}(\underline{\mathbf{x}}_n) \\
&= \underline{\mathbf{x}}_n + 2\mu(\mathbf{I}_{N^2\times N^2} - \mathbf{M}^T)(\underline{T}(\underline{\mathbf{x}}_n) - \underline{\mathbf{x}}_n) - \mu\lambda\nabla \underline{G}(\underline{\mathbf{x}}_n) \\
&= \underline{\mathbf{x}}_n + 2\mu(\underline{T}(\underline{\mathbf{x}}_n) - \underline{\mathbf{x}}_n) - 2\mu\mathbf{M}^T(\underline{T}(\underline{\mathbf{x}}_n) - \underline{\mathbf{x}}_n) - \mu\lambda\nabla \underline{G}(\underline{\mathbf{x}}_n).
\end{aligned} \tag{6.10}$$

Now defining $\underline{\mathbf{y}}_n = \underline{T}(\underline{\mathbf{x}}_n) - \underline{\mathbf{x}}_n$ and taking "over-left-arrow" $\underleftarrow{\cdot\cdot}$, i.e, converting the image expressed in column-stacked vector order to matrix representation gives

$$\mathbf{x}_{n+1} = \mathbf{x}_n + 2\mu(\mathbf{y}_n - \overleftarrow{\mathbf{M}^T\underline{\mathbf{y}}_n}) - \mu\lambda\nabla G(\mathbf{x}_n). \tag{6.11}$$

Finally, substituting $\mathbf{z}_n = \overleftarrow{\mathbf{M}^T\underline{y}_n}$ gives

$$\mathbf{x}_{n+1} = \mathbf{x}_n + 2\mu(\mathbf{y}_n - \mathbf{z}_n) - \mu\lambda\nabla G(\mathbf{x}_n). \tag{6.12}$$

Note that $\mathbf{y}_n$ can be easily computed as $T(\mathbf{x}_n) - \mathbf{x}_n$. The only difficulty of computation is now $\mathbf{z}_n = \overleftarrow{\mathbf{M}^T\underline{\mathbf{y}}_n}$. The following proposition shows that we can compute $\overleftarrow{\mathbf{M}^T\underline{\mathbf{x}}}$, for any given $\underline{\mathbf{x}}$ based on $\mathbf{x} = \overleftarrow{\underline{\mathbf{x}}}$ without dealing with large matrices.

6.3.1 Computing the Action of the Adjoint of the Fractal Transform Operator

Matrix $\mathbf{M}$ is not necessarily symmetric. Hence, in the above iterative methods, it is necessary to compute its adjoint, or the *action of its adjoint*, as derived in the following proposition.

Proposition 6.3.1. Assume T is the fractal image transform of some image with fractal code $\{J(i), \alpha_i, \beta_i\}$, for $i \in \mathfrak{C}$ i.e.,

$$\Big(T(\mathbf{x})\Big)\Big|_{C_i} = \alpha_i \mathcal{D}\Big(\mathbf{x}\Big|_{P_{J(i)}}\Big) + \beta_i, \tag{6.13}$$

for the case of square domain and range blocks, with domain blocks twice the size (in each direction) of the range block size. Also, let $\mathbf{M}$ be defined as in Equation 6.2, and operators $\mathcal{U}$ and

$\mathcal{D}$ be respectively, the upsampling and downsampling operators defined in Section 1.4. Then, for any $\mathbf{x} = \overleftarrow{\underline{\mathbf{x}}}$,

$$\Big(\overleftarrow{\mathbf{M}\underline{\mathbf{x}}}\Big)\Big|_{C_i} = \alpha_i \mathcal{D}\Big(\mathbf{x}\Big|_{P_{J(i)}}\Big) \tag{6.14}$$

and

$$\begin{aligned} \Big(\overleftarrow{\mathbf{M}^T\underline{\mathbf{x}}}\Big)\Big|_{P_j} &= \sum_{i\in\mathfrak{C}_j} \frac{\alpha_i}{4}\mathcal{U}\Big(\mathbf{x}\Big|_{C_i}\Big) \qquad (6.15)\\ \text{where} \qquad \mathfrak{C}_j &= \{i\in\mathfrak{C} \mid J(i)=j\}, \qquad \text{for any } j\in\mathfrak{P}. \end{aligned}$$

Proof. Recall that

$$\underline{T}(\underline{\mathbf{x}}) = \mathbf{M}\underline{\mathbf{x}} + \mathbf{B}. \tag{6.16}$$

Now decompose matrix $\mathbf{M}$ into the weighted sum of $|\mathfrak{C}|$ matrices

$$\mathbf{M} = \sum_{i\in\mathfrak{C}} \alpha_i \mathbf{M}_i, \tag{6.17}$$

where each matrix $\mathbf{M}_i$ is a product (composition) of 3 matrices (operations) as follows

$$\mathbf{M}_i\underline{\mathbf{x}} = \mathbf{R}_i^c \mathbf{D}\mathbf{S}_{J(i)}^p \underline{\mathbf{x}}. \tag{6.18}$$

In this Equation, $\mathbf{S}_{J(i)}^p$ is the matrix operation that corresponds to selecting the range block $J(i)$ of $\underline{\mathbf{x}}$. $\mathbf{D}$ is the matrix operation corresponding to the local averaging followed by down-sampling. Finally, $\mathbf{R}_i^c$ is the matrix operation that replaces the content passed to it at the range location i of the matrix to be returned, and returns zero at all of the other locations. Due to the fact that range blocks are not overlapping, this decomposition is well-defined. To prove the first part of the proposition note that,

$$\mathbf{M}\underline{\mathbf{x}} = (\sum_{i\in\mathfrak{C}} \alpha_i \mathbf{M}_i)\underline{\mathbf{x}}. \tag{6.19}$$

Therefore,

$$\begin{aligned} \Big(\overleftarrow{\mathbf{M}\underline{\mathbf{x}}}\Big)\Big|_{C_i} &= \Big(\overleftarrow{(\sum_{i'\in\mathfrak{C}} \alpha_{i'}\mathbf{M}_{i'})\underline{\mathbf{x}}}\Big)\Big|_{C_i} = \alpha_i \Big(\overleftarrow{\mathbf{M}_i\underline{\mathbf{x}}}\Big)\Big|_{C_i} \qquad (6.20)\\ &= \alpha_i \Big(\overleftarrow{\mathbf{R}_i^c\mathbf{D}\mathbf{S}_{J(i)}^p\underline{\mathbf{x}}}\Big)\Big|_{C_i} = \alpha_i \Big(\overleftarrow{\mathbf{D}\mathbf{S}_{J(i)}^p\underline{\mathbf{x}}}\Big) \qquad (6.21)\\ &= \alpha_i \mathcal{D}\Big(\mathbf{x}\Big|_{P_{J(i)}}\Big). \qquad (6.22) \end{aligned}$$

To prove the second part, remember that we can partition $\mathfrak{C}$ with the sets $\mathfrak{C}_{j'}$,

$$\mathfrak{C} = \bigcup_{j'\in\mathfrak{P}} \mathfrak{C}_{j'}. \tag{6.23}$$

Hence,

$$\Big(\overleftarrow{\mathbf{M}^T\underline{\mathbf{x}}}\Big)\Big|_{P_j} = \Big(\overleftarrow{\sum_{i\in\mathfrak{C}}\alpha_i\mathbf{M}_i^T\underline{\mathbf{x}}}\Big)\Big|_{P_j} = \Big(\overleftarrow{\sum_{j'\in\mathfrak{P}}\sum_{i\in\mathfrak{C}_{j'}}\alpha_i\mathbf{M}_i^T\underline{\mathbf{x}}}\Big)\Big|_{P_j} \tag{6.24}$$

$$= \Big(\overleftarrow{\sum_{i\in\mathfrak{C}_j}\alpha_i\mathbf{M}_i^T\underline{\mathbf{x}}}\Big)\Big|_{P_j} = \sum_{i\in\mathfrak{C}_j}\alpha_i\Big(\overleftarrow{\mathbf{M}_i^T\underline{\mathbf{x}}}\Big)\Big|_{P_j} \tag{6.25}$$

$$= \sum_{i\in\mathfrak{C}_j}\alpha_i\Big(\overleftarrow{(\mathbf{R}_i^c\mathbf{D}\mathbf{S}_{J(i)}^p)^T\underline{\mathbf{x}}}\Big)\Big|_{P_j} = \sum_{i\in\mathfrak{C}_j}\alpha_i\Big(\overleftarrow{(\mathbf{R}_i^c\mathbf{D}\mathbf{S}_j^p)^T\underline{\mathbf{x}}}\Big)\Big|_{P_j} \tag{6.26}$$

$$= \sum_{i\in\mathfrak{C}_j}\alpha_i\Big(\overleftarrow{(\mathbf{S}_j^p)^T(\mathbf{D})^T(\mathbf{R}_i^c)^T\underline{\mathbf{x}}}\Big)\Big|_{P_j} = \sum_{i\in\mathfrak{C}_j}\alpha_i\Big(\overleftarrow{(\mathbf{R}_j^p)(\frac{\mathbf{U}}{4})(\mathbf{S}_i^c)\underline{\mathbf{x}}}\Big)\Big|_{P_j}. \tag{6.27}$$

In the last line of the equation we used the fact that the transpose of a range i replacement matrix, i.e., $(\mathbf{R}_i^c)^T$ equals the range i selection matrix $\mathbf{S}_i^c$. The upsampling matrix $\frac{\mathbf{U}}{4}$ equals $\mathbf{D}^T$, where the division by 4 is to compensate the local averaging operator applied prior to downsampling in $\mathbf{D}$. Also, the transpose of the domain j selection matrix, i.e., $(\mathbf{S}_j^p)^T$ equals the domain j replacement matrix $\mathbf{R}_j^p$. Therefore,

$$\Big(\overleftarrow{\mathbf{M}^T\underline{\mathbf{x}}}\Big)\Big|_{P_j} = \sum_{i\in\mathfrak{C}_j}\frac{\alpha_i}{4}\Big(\overleftarrow{(\mathbf{R}_j^p)(\mathbf{U})(\mathbf{S}_i^c)\underline{\mathbf{x}}}\Big)\Big|_{P_j} = \sum_{i\in\mathfrak{C}_j}\frac{\alpha_i}{4}\Big(\overleftarrow{\mathbf{U}\mathbf{S}_i^c\underline{\mathbf{x}}}\Big) \tag{6.28}$$

$$= \sum_{i\in\mathfrak{C}_j}\frac{\alpha_i}{4}\mathcal{U}\Big(\mathbf{x}\Big|_{C_i}\Big). \tag{6.29}$$

□

Equation 6.15 states that to compute $\overleftarrow{\mathbf{M}^T\underline{\mathbf{x}}}$ on each domain block piece P_j, we consider all range blocks C_i corresponding to this fixed domain block, up-sample each of these range blocks, multiply it by its corresponding α_i value divided by 4, and finally sum over all of these up-sampled and appropriately scaled pieces. This sum yields the required domain block portion P_j of $\overleftarrow{\mathbf{M}^T\underline{\mathbf{x}}}$, i.e., $\Big(\overleftarrow{\mathbf{M}^T\underline{\mathbf{x}}}\Big)\Big|_{P_j}$. Note that the non-overlapping constraint on domain blocks is required for $\overleftarrow{\mathbf{M}^T\underline{\mathbf{x}}}$ to be uniquely determined for this method to be well-defined.

Now we are prepared to summarize our minimization algorithm. Recall from Equation 6.12, that we have to compute $\mathbf{z}_n = \overleftarrow{\mathbf{M}^T\underline{\mathbf{y}}_n}$. We do so by using Equation 6.15 of Proposition 6.3.1 to yield the following.

6.3.2 Regularized Fractal Decoding Algorithm (RFDA)

The sequence $\{\mathbf{x}_n\}$, defined iteratively as follows converges to an approximation of $\mathbf{u}^\star{}_{G,\lambda} = \underline{\overleftarrow{\mathbf{u}^\star{}_{G,\lambda}}}$ for any $\mathbf{x}_0$ (typically a constant zero image) and sufficiently small μ

$$\begin{aligned}
\mathbf{y}_n &= T(\mathbf{x}_n) - \mathbf{x}_n \\
\mathbf{z}_n\Big|_{P_j} &= \sum_{i\in\mathfrak{C}_j}\frac{\alpha_i}{4}\mathcal{U}\Big(\mathbf{y}_n\Big|_{C_i}\Big), \qquad \forall j\in\mathfrak{P}, \\
\text{where}\quad \mathfrak{C}_j &= \{i \mid J(i) = j\}, \\
\mathbf{x}_{n+1} &= \mathbf{x}_n + 2\mu(\mathbf{y}_n - \mathbf{z}_n) - \mu\lambda\nabla G(\mathbf{x}_n).
\end{aligned} \tag{6.30}$$

6.4 Various Regularization Functions

The choice of regularization function G may depend on the application. In this section, we briefly state three different popular regularization functions and explicitly form their RFDA counterparts.

- **Tikhonov Regularization**
  The special case, $\underline{G}(\underline{\mathbf{x}}) = \| \underline{\mathbf{x}} \|^2$ is called Tikhonov regularization, as explained in Chapter 2. It can be shown that the analytic solution in this case is

$$\underline{\mathbf{u}^\star_{G,\lambda}} = (\mathbf{A}^T\mathbf{A} + \lambda \mathbf{I}_{N^2\times N^2})^{-1}\mathbf{A}^T\mathbf{B}. \tag{6.31}$$

  The RFDA counterpart below can be used to estimate $\mathbf{u}^\star{}_{G,\lambda}$.

$$\begin{aligned} \mathbf{y}_n &= T(\mathbf{x}_n) - \mathbf{x}_n \\ \mathbf{z}_n\Big|_{P_j} &= \sum_{i\in\mathfrak{C}_j} \frac{\alpha_i}{4}\mathcal{U}\Big(\mathbf{y}_n\Big|_{C_i}\Big), \qquad \forall j \in \mathfrak{P} \\ \mathbf{x}_{n+1} &= (1-2\mu\lambda)\mathbf{x}_n + 2\mu(\mathbf{y}_n - \mathbf{z}_n). \end{aligned} \tag{6.32}$$

- **Total Variation (TV)**
  A possibility is to choose Total Variation(TV) for the regularization function. In this case, (cf. Equation 2.49)

$$\underline{G}(\underline{\mathbf{x}}) = TV(\underline{\mathbf{x}}) = \int |\nabla \underline{\mathbf{x}}| dxdy \tag{6.33}$$

$$\text{and } \nabla TV(\underline{\mathbf{x}}) = -\nabla\cdot\big(\frac{\nabla\underline{\mathbf{x}}}{|\nabla\underline{\mathbf{x}}|}\big). \tag{6.34}$$

  Hence, the RFDA corresponding to this regularization becomes

$$\begin{aligned} \mathbf{y}_n &= T(\mathbf{x}_n) - \mathbf{x}_n \\ \mathbf{z}_n\Big|_{P_j} &= \sum_{i\in\mathfrak{C}_j} \frac{\alpha_i}{4}\mathcal{U}\Big(\mathbf{y}_n\Big|_{C_i}\Big), \qquad \forall j \in \mathfrak{P} \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + 2\mu(\mathbf{y}_n - \mathbf{z}_n) + \mu\lambda\nabla\cdot\big(\frac{\nabla\mathbf{x}_n}{|\nabla\mathbf{x}_n|}\big). \end{aligned} \tag{6.35}$$

- **Hard Constrained**
  In some certain cases, we may have the prior knowledge that the encoded image belongs to some closed and convex set Ψ_c (typically a subset of the Hilbert space in which the image is defined). For example, Ψ_c can be constructed based on certain smoothness or energy boundedness constraints on the image. Assume that Ψ_c is given and its projection operator $\mathcal{P}_{\Psi_c}$ is easy to compute. In this case, it can be assumed that

$$\underline{G}(\underline{\mathbf{x}}) = 1_{\Psi_c}(\underline{\mathbf{x}}) \times \infty, \tag{6.36}$$

  where 1_{Ψ_c} is the characteristic function on Ψ_c, and $0\times\infty = 0$. Similar to use of SD now applying the *projected Landweber algorithm* 2.6.1 in the RFDA, yields the following

algorithm in the so-called *hard constrained case*.

$$\begin{aligned} \mathbf{y}_n &= T(\mathbf{x}_n) - \mathbf{x}_n \\ \mathbf{z}_n\Big|_{P_j} &= \sum_{i \in \mathfrak{C}_j} \frac{\alpha_i}{4} \mathcal{U}\Big(\mathbf{y}_n\Big|_{C_i}\Big), \qquad \forall j \in \mathfrak{P} \\ \mathbf{x}_{n+1} &= \mathcal{P}_{\Psi_c}\Big(\mathbf{x}_n + 2\mu(\mathbf{y}_n - \mathbf{z}_n)\Big). \end{aligned} \tag{6.37}$$

6.5 Conjugate Gradient RFDA

In the previous sections, we proposed the RFDA algorithm based on the steepest descent iterations. It is also possible to use the conjugate gradient iterations explained in Section 2.6.4, again using Proposition 6.3.1 to avoid dealing with large matrices.

In general, conjugate gradient has superior convergence rates compared to the steepest descent iterations. In the next section, we present results based on both RFDA and Conjugate Gradient RFDA.

6.6 Application to Restoration of an Image with an Incomplete Fractal Code

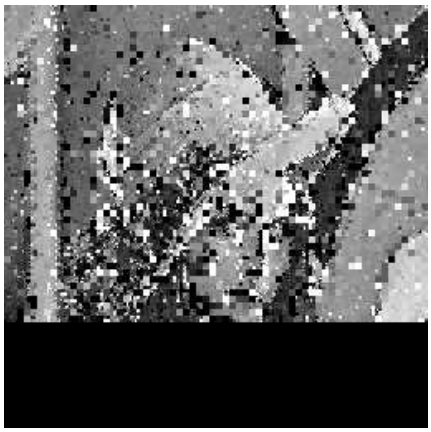
In the previous chapter, we introduced the problem of restoring an image with an incomplete fractal code, and used the method of projection onto convex sets (POCS) to address this problem. Here, we use the regularization approach introduced above.

6.6.1 Applying the RFDA Algorithm

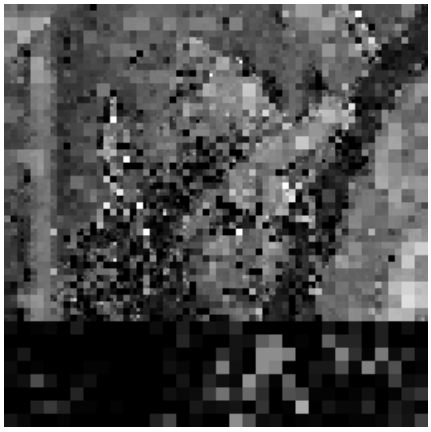
Assume that we are given an incomplete fractal code of an image, i.e., some of the domain-range block assignments and corresponding greyscale map coefficients are missing. Recall that, in such a case, the usual fractal decoding scheme, in which an arbitrary "seed" image is employed, will collapse since the range blocks of the image for which the fractal code is missing cannot be modified. These blocks will simply remain identical to the corresponding subblocks of the seed image.

To illustrate, let us consider the fractal code associated with the *Lena* image (256×256 pixels, 8 bits per pixel) of respectively, 8×8 and 4×4, domain and range block size.

Suppose that the fractal code corresponding to all range blocks in the bottom quarter of the image are missing. Using a black image as the seed, the limiting image produced by the fractal decoding procedure is shown in Figure 6.1(a). Only the domain blocks in the top-three-quarters of the image have been modified – those in the bottom quarter are identical to their counterparts in the seed black image.



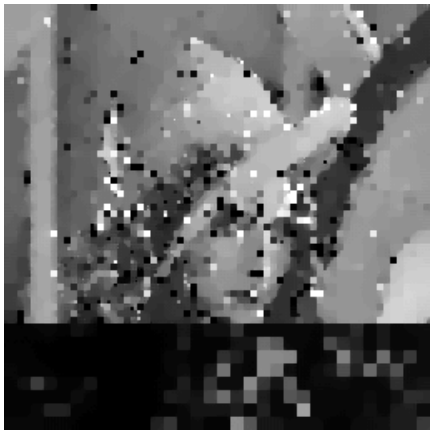
(a)



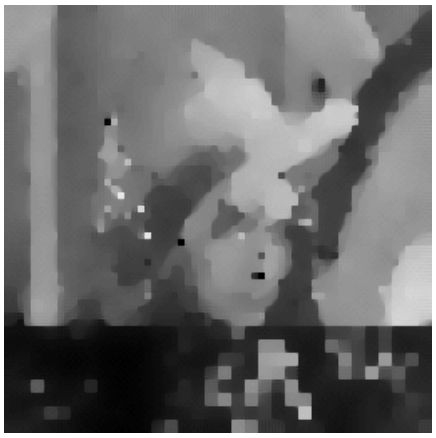
(b)

(c)

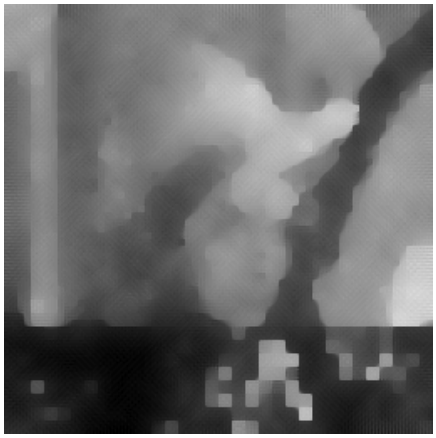
Figure 6.1: (a) Decoded as bottom quarter of the fractal code is missing, black seed. (b) Proposed Minimum norm least square solution using RFDA with $\lambda = 0, \mu = 0.005$. (c) Proposed Hard-constrained RFDA with smoothness constraint, $\mu = 0.005$.



(a)



(b)



(c)

Figure 6.2: (a) Proposed RFDA using using TV regularization with $\lambda = 0.2, \mu = 0.005$. (b) Proposed RFDA using using TV regularization with $\lambda = 1, \mu = 0.005$. (c) Proposed RFDA using using TV regularization with $\lambda = 2, \mu = 0.005$.

We now show how this situation can be improved using the theory presented in this chapter. We consider different types of regularization functions discussed in this chapter, and apply the RFDA with a fixed small step size $\mu = 0.005$ in all cases after mapping the image values to $[0, 1]$. Now we use the RFDA to estimate the solution to this under-determined problem.

First we set $\lambda = 0$ and use RFDA iterating from $\mathbf{x}_0 = 0$. This scheme will find the minimum norm least squares solution to the problem. The result is given in Figure 6.1(b).

It can be seen that the bottom quarter of the image which was not changed at all by the traditional fractal decoding in Figure 6.1(a) has been improved. The RFDA updates $\mathbf{z}_n$ by

$$\mathbf{z}_n\Big|_{P_j} = \sum_{i \in \mathfrak{C}_j} \frac{\alpha_i}{4} \mathcal{U}\Big(\mathbf{y}_n\Big|_{C_i}\Big), \qquad \forall j \in \mathfrak{P}. \tag{6.38}$$

This actually causes an alteration of *domain blocks* P_j that are included in the missing range part, i.e., the bottom quarter of the image. Since some domain blocks used in the fractal coding procedure come from that region, it is possible that portions of the lower part of the image are modified.

In Figure 6.1(c), we show the result obtained using the RFDA in the hard-constrained case, as explained in Equation 6.37. The initial image was black, i.e., $\mathbf{x}_0 = 0$. Ψ_c can be thought of as the set of images with certain smoothness, e.g, low-pass filtered in the frequency domain, or Gaussian smoothed. In this case, $\mathcal{P}_{\Psi_c}(\underline{\mathbf{x}})$ the projection operator on Ψ_c smoothes the input image $\underline{\mathbf{x}}$. Clearly, Figure 6.1(c) is an improvement over the usual fractal coding method of Figure 6.1(a), although some edges are smeared out as a result of the projection $\mathcal{P}_{\Psi_c}$.

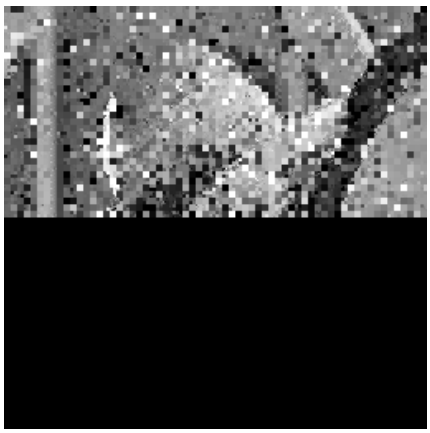
In Figures 6.2(a,b,c) are shown the results obtained by applying RFDA with total variation regularization as in 6.35, with $\lambda = 0.2$, 1 and 2, respectively. It can be seen that the important edges are preserved using the TV model. It is a noteworthy improvement over Figure 6.1(a) to preserve the edges as well as to recover some missing information from the bottom quarter of the image.

6.6.2 Applying the Conjugate Gradient RFDA

It can be observed that the RFDA algorithm does not yield superior results compared to the POCS method in the previous chapter, even though we have a fractal code with more information (three quarters of the code was known as opposed to one half in case of the POCS examples given). We associate this mostly with the fact that we used a fixed step size in the associated gradient descent iterations. Figures 6.3 and 6.4 show improved results based on the proposed conjugate gradient RDFA.

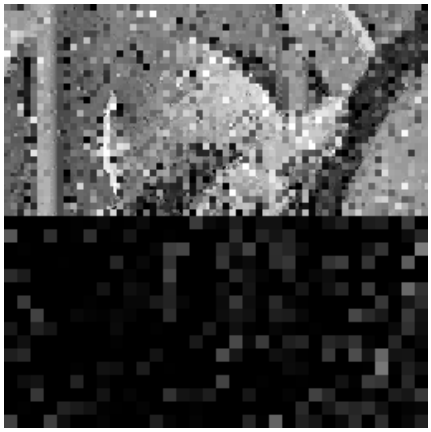
(a)

(b)



(c)

Figure 6.3: (a) Lena original. (b) Lena attractor 4×4 range block size. (c) Decoded as bottom half of the fractal code is missing, black seed.



(a)

(b)

(c)

Figure 6.4: (a) Proposed Minimum norm least square solution using conjugate gradient RFDA with $\lambda = 0$. (b) Proposed conjugate gradient RFDA solution using Laplacian regularization with $\lambda = 1$. (c) Proposed conjugate gradient RFDA solution using TV regularization with $\lambda = 1$.

6.7 Concluding Remarks

In this chapter, we have described a reformulation of traditional fractal image decoding along with regularization. The solution of the traditional fractal-based methods are too restrictive in the sense that no additional knowledge about the image can be combined with self-similarity constraints to improve the decoded image. The principal advantage of our framework is that it provides the flexibility to incorporate constraints and possibly additional knowledge about the reconstructed image.

We have improved this situation by viewing the fractal transform operator as a linear operator, and modelling the decoding procedure as a least-squares problem. In addition, we have applied the steepest descent, projected Landweber, and conjugate gradient algorithms to numerically estimate the regularized solutions. This procedure produced the regularized fractal decoding algorithm (RFDA).

We also introduced an explicit formulation to combine fractal image decoding with a variety of regularization procedures, e.g., Tikhonov, total variation (TV), or hard-constrained. A regularized decoding approach also provides the opportunity of solving under-determined inverse problems in fractal coding, as we have shown for the case of the incomplete fractal code problem.

Chapter 7

Regularization Expressions Involving Self-similarity

7.1 Introduction

In this chapter, we introduce and analyze a set of regularization expressions based on self-similarity properties of images in order to address the classical inverse problem of image denoising and the ill-posed inverse problem of single-frame image zooming.

The regularization expressions introduced are constructed using either the fractal image transform or the newly developed "Nonlocal-means (NL-means) image denoising filter" of Buades *et al.* [BCM05b, BCM05c].

We exploit these regularization terms in a global MAP-based formulation and produce analytical and computational solutions. Analytical comparisons are made with results based on classical methods (e.g., fractal-based denoising and zooming, and NL-means image denoising).

In the sections addressing the denoising problem, images and observations are assumed as elements of $\ell^2(\Upsilon)$ for $\Upsilon = [1, \ldots, N] \times [1, \ldots, N]$. In the sections addressing the zooming problem, the ideal images (objects) are assumed to be in $\ell^2(\Psi)$ for $\Psi = [1, \ldots, Nz] \times [1, \ldots, Nz]$, and the integer $z > 1$.

7.2 Regularization via the Fractal-Transform Operator

7.2.1 Denoising

In this subsection, we consider the following problem of image denoising: Given

$$\underline{\mathbf{u}} = \underline{\mathbf{x}} + \underline{\mathbf{n}}, \tag{7.1}$$

find an approximation of $\underline{\mathbf{x}}$. As mentioned earlier, it was observed [GFV03, GFV06, Ale05] that fractal-based methods have denoising capabilities. If we assume that $\underline{T}$ is a contractive fractal transform of an image $\underline{\mathbf{u}}$, then its unique fixed point $\underline{\mathbf{u}^\star}$ is an approximation of $\underline{\mathbf{x}}$.

We seek to improve this approximation by emphasizing the influence of the observed data $\underline{\mathbf{u}}$ on the reconstruction process. First of all, note that we can write $\underline{\mathbf{u}^\star}$, the unique fixed point of the contractive fractal transform $\underline{T}$ as

$$\underline{\mathbf{u}^\star} = \arg\min_{\underline{\mathbf{x}}} \| \underline{\mathbf{x}} - \underline{T}(\underline{\mathbf{x}}) \|^2. \tag{7.2}$$

Now $\underline{T}$ can be represented as follows,

$$\underline{T}(\underline{\mathbf{x}}) = \mathbf{M}\underline{\mathbf{x}} + \mathbf{B}, \tag{7.3}$$

where the matrices $\mathbf{M}$ and $\mathbf{B}$ contain the "greyscale map" parameters, α_i and β_i, respectively, as well as the domain-range assignments of the transform $\underline{T}$.

We consider two constructions of the regularization expression:

1. Squared collage error,
$$\underline{G_c}(\underline{\mathbf{x}}) = \| \underline{\mathbf{x}} - \underline{T}(\underline{\mathbf{x}}) \|^2. \tag{7.4}$$

2. Squared fixed-point approximation error,
$$\underline{G_f}(\underline{\mathbf{x}}) = \| \underline{\mathbf{x}} - \underline{\mathbf{u}^\star} \|^2. \tag{7.5}$$

In the first case, where $\underline{G_c}$ is the collage error, we define $\mathbf{A} = \mathbf{I}_{N^2\times N^2} - \mathbf{M}$, in which $\mathbf{I}_{N^2\times N^2}$ is the identity matrix, so that

$$\underline{G_c}(\underline{\mathbf{x}}) = \| \underline{\mathbf{x}} - \underline{T}(\underline{\mathbf{x}}) \|^2 = \| \mathbf{A}\underline{\mathbf{x}} - \mathbf{B} \|^2. \tag{7.6}$$

Then

$$\begin{aligned} \underline{\tilde{\mathbf{x}}_{G_c,\lambda}} &= \arg\min_{\underline{\mathbf{x}}} \left[\|\underline{\mathbf{x}} - \underline{\mathbf{u}}\|^2 + \lambda \| \underline{\mathbf{x}} - \underline{T\mathbf{x}} \|^2 \right] \\ &= \arg\min_{\underline{\mathbf{x}}} \left[\|\underline{\mathbf{x}} - \underline{\mathbf{u}}\|^2 + \lambda \| \mathbf{A}\underline{\mathbf{x}} - \mathbf{B} \|^2 \right] \\ &= \arg\min_{\underline{\mathbf{x}}} \left[\|\underline{\mathbf{x}} - \underline{\mathbf{u}}\|^2 + \lambda \| \mathbf{A}(\underline{\mathbf{x}} - \underline{\mathbf{u}^\star}) \|^2 \right]. \end{aligned} \tag{7.7}$$

It follows that
$$\underline{\tilde{\mathbf{x}}_{G_c,\lambda}} = (\mathbf{I}_{N^2\times N^2} + \lambda \mathbf{A}^T\mathbf{A})^{-1}(\underline{\mathbf{u}} + \lambda \mathbf{A}^T\mathbf{B}). \tag{7.8}$$
The matrix $\mathbf{I}_{N^2\times N^2} + \lambda\mathbf{A}^T\mathbf{A}$ is positive definite for any $\lambda \geq 0$ and hence invertible.

In the second case, where $\underline{G_f}$ is the fixed point error, we define

$$\underline{\tilde{\mathbf{x}}_{G_f,\lambda}} = \arg\min_{\underline{\mathbf{x}}} \left[\|\underline{\mathbf{x}} - \underline{\mathbf{u}}\|^2 + \lambda \| \underline{\mathbf{x}} - \underline{\mathbf{u}^\star} \|^2 \right], \tag{7.9}$$

and it easily follows that
$$\underline{\tilde{\mathbf{x}}_{G_f,\lambda}} = \frac{1}{1+\lambda}\underline{\mathbf{u}} + \frac{\lambda}{1+\lambda}\underline{\mathbf{u}^\star}. \tag{7.10}$$
This is simply a weighted average of the measurement $\underline{\mathbf{u}}$ and the attractor $\underline{\mathbf{u}^\star}$ of the fractal transform $\underline{T}$. The following is an asymptotic result on the difference of the minimizers of the above two cases.

Proposition 7.2.1.

$$\underline{\tilde{\mathbf{x}}_{G_f,\lambda}} - \underline{\tilde{\mathbf{x}}_{G_c,\lambda}} = \lambda(\mathbf{I}_{N^2\times N^2} - \mathbf{A}^T\mathbf{A})(\underline{\mathbf{u}^\star} - \underline{\mathbf{u}}) + O(\lambda^2) \qquad as\ \lambda \to 0. \tag{7.11}$$

Proof. Note that, as $\lambda \to 0$,

$$\underline{\tilde{\mathbf{x}}_{G_c,\lambda}} = (\mathbf{I}_{N^2\times N^2} + \lambda \mathbf{A}^T\mathbf{A})^{-1}(\underline{\mathbf{u}} + \lambda \mathbf{A}^T\mathbf{B}) = [\mathbf{I}_{N^2\times N^2} - \lambda\mathbf{A}^T\mathbf{A} + O(\lambda^2)](\underline{\mathbf{u}} + \lambda \mathbf{A}^T\mathbf{B}). \tag{7.12}$$

Therefore,

$$\begin{aligned}
\underline{\tilde{\mathbf{x}}_{G_f,\lambda}} - \underline{\tilde{\mathbf{x}}_{G_c,\lambda}} &= \frac{1}{1+\lambda}\underline{\mathbf{u}} + \frac{\lambda}{1+\lambda}\underline{\mathbf{u}^\star} - \underline{\tilde{\mathbf{x}}_{G_c,\lambda}} \\
&= (1-\lambda)\underline{\mathbf{u}} + \lambda\underline{\mathbf{u}^\star} + O(\lambda^2) - \underline{\tilde{\mathbf{x}}_{G_c,\lambda}} \\
&= (1-\lambda)\underline{\mathbf{u}} + \lambda\underline{\mathbf{u}^\star} + O(\lambda^2) - [\mathbf{I}_{N^2\times N^2} - \lambda\mathbf{A}^T\mathbf{A} + O(\lambda^2)](\underline{\mathbf{u}} + \lambda\mathbf{A}^T\mathbf{B}) \\
&= \underline{\mathbf{u}} - \lambda\underline{\mathbf{u}} + \lambda\underline{\mathbf{u}^\star} - \underline{\mathbf{u}} + \lambda\mathbf{A}^T\mathbf{A}\underline{\mathbf{u}} - \lambda\mathbf{A}^T\mathbf{B} + O(\lambda^2) \\
&= \lambda(\underline{\mathbf{u}^\star} - \underline{\mathbf{u}} + \mathbf{A}^T(\mathbf{A}\underline{\mathbf{u}} - \mathbf{B})) + O(\lambda^2) \\
&= \lambda(\underline{\mathbf{u}^\star} - \underline{\mathbf{u}} + \mathbf{A}^T(\mathbf{A}\underline{\mathbf{u}} - \mathbf{B} - (\mathbf{A}\underline{\mathbf{u}^\star} - \mathbf{B}))) + O(\lambda^2) \\
&= \lambda(\underline{\mathbf{u}^\star} - \underline{\mathbf{u}} - \mathbf{A}^T\mathbf{A}(\underline{\mathbf{u}^\star} - \underline{\mathbf{u}})) + O(\lambda^2) \\
&= \lambda(\mathbf{I}_{N^2\times N^2} - \mathbf{A}^T\mathbf{A})(\underline{\mathbf{u}^\star} - \underline{\mathbf{u}}) + O(\lambda^2). \qquad (7.13)
\end{aligned}$$

□

7.2.2 Zooming

In this subsection, we consider the following problem of image zooming: Given the observed image data $\mathbf{y}$, where

$$\mathbf{y} = \mathcal{D}_z(\mathbf{x}) + \mathbf{n}, \qquad (7.14)$$

in which $\mathcal{D}_z$ is the downsampling by a factor of z, find an approximation of $\underline{\mathbf{x}}$.

The precise definition of the operator $\mathcal{D}_z$ is given in Section 1.4. Assuming the corresponding vector notation of the equation, we can write

$$\underline{\mathbf{y}} = \mathbf{D}_z\underline{\mathbf{x}} + \underline{\mathbf{n}}. \qquad (7.15)$$

In the fractal coding literature, the normal procedure of zooming an image by a factor of z is to find a fractal transform $\underline{T}$ for $\underline{\mathbf{y}}$, once again by minimizing $\| \underline{T}(\underline{\mathbf{y}}) - \underline{\mathbf{y}} \|$. One then "zooms" this attractor by applying the fractal transform $\underline{T}$ (actually the operator $\underline{T}_z$ induced by $\underline{T}$) to an image that is z-times larger in each direction to produce the attractor $\underline{\mathbf{y}_z^\star}$.

Here, we shall use the zoomed fixed-point error to define the regularization expression, i.e., $\underline{G_z}(\underline{\mathbf{x}}) = \|\underline{\mathbf{x}} - \underline{\mathbf{y}_z^\star}\|^2$ so that the minimization problem becomes

$$\underline{\tilde{\mathbf{x}}_{G_z,\lambda}} = \arg\min_{\mathbf{x}} \left[\|\mathbf{D}_z\underline{\mathbf{x}} - \underline{\mathbf{y}}\|^2 + \lambda\|\underline{\mathbf{x}} - \underline{\mathbf{y}_z^\star}\|^2. \right] \qquad (7.16)$$

Notice that the above functional balances the consistency with original data, via the term $\|\mathbf{D}_z\underline{\mathbf{x}} - \underline{\mathbf{y}}\|^2$, along with the self-similarity constraints encoded in $\underline{\mathbf{y}_z^\star}$. It follows that the solution of this minimization problem is

$$\underline{\tilde{\mathbf{x}}_{G_z,\lambda}} = (\mathbf{D}_z^T\mathbf{D}_z + \lambda\mathbf{I}_{z^2N^2\times z^2N^2})^{-1}(\mathbf{D}_z^T\underline{\mathbf{y}} + \lambda\underline{\mathbf{y}_z^\star}). \qquad (7.17)$$

Computational results on this will be presented in the following section.

7.3 Computational Considerations and Results

In this section, we present some computational results for the fractal-based regularizations introduced in the previous section, as applied to the denoising and zooming problems.

In Figure 7.1, are shown some results obtained by the image denoising method (2) of Subsection 7.2.1 in which the regularization function G_f is the squared fixed-point approximation error, $\| \underline{\mathbf{x}} - \underline{\mathbf{u}^\star} \|^2$, cf. Equation 7.9.

The "input" or observed noisy image was the 256×256-pixel (8bpp) image $\underline{\mathbf{u}}$, at the upper left, with zero-mean Gaussian white noise with standard deviation $\sigma = 15$. This noisy image was fractally coded using domain and range blocks of size 16×16 and 8×8 pixels, respectively. The attractor $\underline{\mathbf{u}^\star}$ of the fractal transform is shown at the bottom right. Recall from Equation 7.10 that the solution $\underline{\tilde{\mathbf{x}}_{G_f,\lambda}}$ to the minimization problem is given by a weighted average of $\underline{\mathbf{u}}$ and $\underline{\mathbf{u}^\star}$. Various weighted averages are shown in the figure, from $\lambda = 0$, corresponding to the original noisy image $\underline{\mathbf{u}}$, to the limit $\lambda \to \infty$, corresponding to the fractal attractor $\underline{\mathbf{u}^\star}$. The plot of PSNR is also given in the figure.

We have observed that, for various test images employed in our study, the denoising method (1) using squared collage-error regularization function, results $\underline{\tilde{\mathbf{x}}_{G_c,\lambda}}$ (cf. Equation 7.8) do not differ visually from $\underline{\tilde{\mathbf{x}}_{G_f,\lambda}}$. However, computing $\underline{\tilde{\mathbf{x}}_{G_c,\lambda}}$ is much more computationally intensive.

In Figure 7.2 are presented some results obtained by the image zooming method of Subsection 7.2.2.

The goal was to zoom an original 128×128-pixel input image $\mathbf{y}$ by a factor of 2. The first column of this Figure shows the result obtained by simple pixel-replication of $\mathbf{y}$, for which there is a great deal of visible blockiness. The input image $\mathbf{y}$ was then fractally coded using 8×8-pixel domain blocks and 4×4-pixel range blocks. The resulting fractal transform T was then applied to an initial seed image of size 256×256 to obtain the 256×256-pixel attractor $\mathbf{y}_z^\star$ in the middle column. In the third column, $\tilde{\mathbf{x}}_{G_z,\lambda}$ is shown for the fixed value of $\lambda = 1$. We have used the conjugate gradient iterations , explained in Subsection 2.6.4, to approximate $\tilde{\mathbf{x}}_{G_z,\lambda}$ in this image. Note that the expression $(\mathbf{D}_z^T \mathbf{D}_z + \lambda \mathbf{I}_{z^2N^2 \times z^2N^2})$ is symmetric positive definite for $\lambda > 0$ and hence the conditions to use the conjugate gradient iterations are satisfied.

$\mathbf{u} = \mathbf{x} + \mathbf{n}$, $\lambda = 0$, PSNR=20.03 dB

$\tilde{\mathbf{x}}_{G_f,\lambda}$, $\lambda = 0.25$, PSNR=21.72 dB

$\tilde{\mathbf{x}}_{G_f,\lambda}$, $\lambda = 0.5$, PSNR=22.89 dB

$\tilde{\mathbf{x}}_{G_f,\lambda}$, $\lambda = 1$, PSNR=24.25 dB

$\tilde{\mathbf{x}}_{G_f,\lambda}$, $\lambda = 2$, PSNR=25.30 dB

$\mathbf{u}^\star$, $\lambda \to \infty$, PSNR=24.35 dB

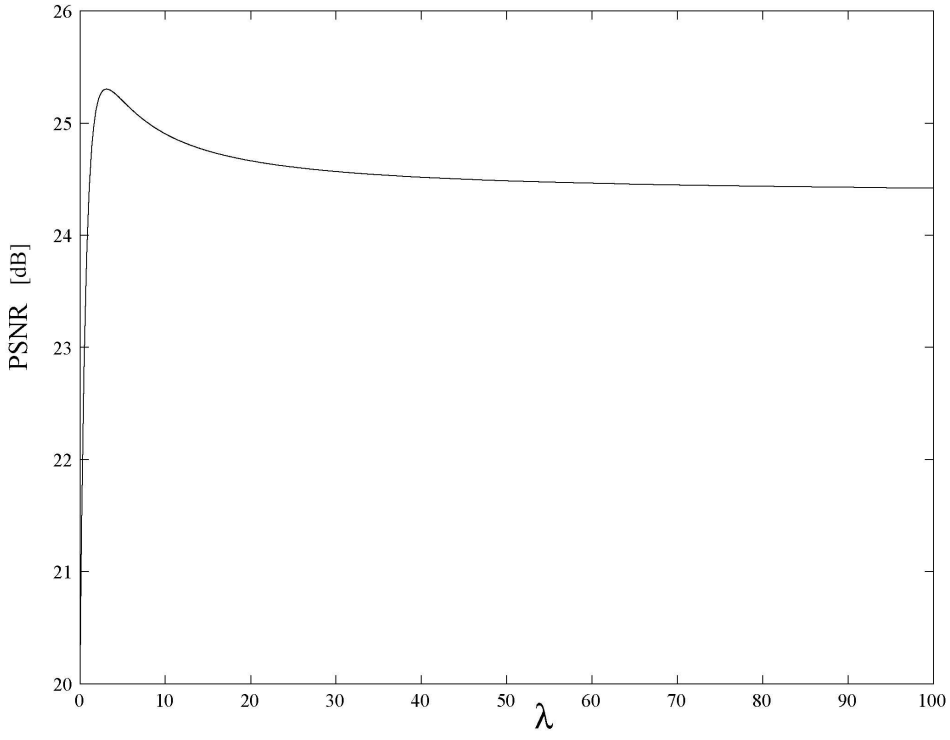
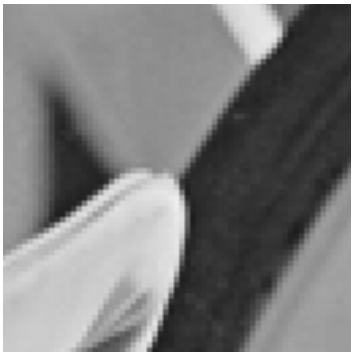


Figure 7.1: Image denoising with fractal-based regularization.

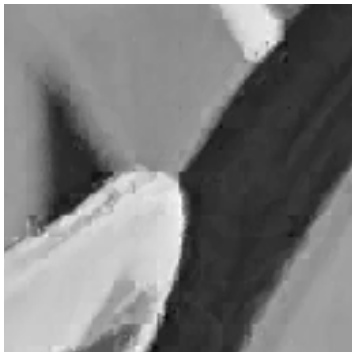
Pixel replication of $\mathbf{y}$

$\mathbf{y}_z^\star$

$\tilde{\mathbf{x}}_{G_z,\lambda}$, $\lambda = 1$



Pixel replication of $\mathbf{y}$



$\mathbf{y}_z^\star$

$\tilde{\mathbf{x}}_{G_z,\lambda}$, $\lambda = 1$

Pixel replication of $\mathbf{y}$

$\mathbf{y}_z^\star$

$\tilde{\mathbf{x}}_{G_z,\lambda}$, $\lambda = 1$

Figure 7.2: Image zooming with fractal-based regularization.

7.4 A Regularization Scheme Based on NL-means Denoising

In this section, we return to the denoising problem

$$\underline{\mathbf{u}} = \underline{\mathbf{x}} + \underline{\mathbf{n}}, \tag{7.18}$$

In Section 3.6.1, the NL-means algorithm was briefly described, including the determination of a weight matrix $\mathbf{W}$, for some smoothness parameter h, so that $\mathbf{W}\underline{\mathbf{u}}$ is the denoised copy of $\underline{\mathbf{u}}$. In what follows, we analyze and make use of this matrix to improve the results.

Once again, we consider two forms for the regularization term,

1. The collage distance error
$$\underline{G_d}(\underline{\mathbf{x}}) = \| \underline{\mathbf{x}} - \mathbf{W}\underline{\mathbf{x}} \|^2. \tag{7.19}$$

2. The explicit approximation error
$$\underline{G_e}(\underline{\mathbf{x}}) = \| \underline{\mathbf{x}} - \mathbf{W}\underline{\mathbf{u}} \|^2. \tag{7.20}$$

In the first case, define

$$\underline{\tilde{\mathbf{x}}_{G_d,\lambda}} = \arg\min_{\underline{\mathbf{x}}} \Big[\| \underline{\mathbf{x}} - \underline{\mathbf{u}} \|^2 + \lambda \| \underline{\mathbf{x}} - \mathbf{W}\underline{\mathbf{x}} \|^2 \Big], \tag{7.21}$$

which yields the solution

$$\underline{\tilde{\mathbf{x}}_{G_d,\lambda}} = [\mathbf{I}_{N^2\times N^2} + \lambda(\mathbf{W} - \mathbf{I}_{N^2\times N^2})^T(\mathbf{W} - \mathbf{I}_{N^2\times N^2})]^{-1}(\underline{\mathbf{u}}). \tag{7.22}$$

Note that the expression $[I + \lambda(\mathbf{W} - \mathbf{I}_{N^2\times N^2})^T(\mathbf{W} - \mathbf{I}_{N^2\times N^2})]$ is positive definite and invertible if $\lambda \geq 0$.

In the second case, define

$$\underline{\tilde{\mathbf{x}}_{G_e,\lambda}} = \arg\min_{\underline{\mathbf{x}}} \Big[\| \underline{\mathbf{x}} - \underline{\mathbf{u}} \|^2 + \lambda \| \underline{\mathbf{x}} - \mathbf{W}\underline{\mathbf{u}} \|^2 \Big]. \tag{7.23}$$

The solution is simply given by

$$\underline{\tilde{\mathbf{x}}_{G_e,\lambda}} = \frac{1}{1+\lambda}\underline{\mathbf{u}} + \frac{\lambda}{1+\lambda}\mathbf{W}\underline{\mathbf{u}}. \tag{7.24}$$

The minimizer $\underline{\tilde{\mathbf{x}}_{G_e,\lambda}}$ obtained in the second case is exactly the one proposed in [BCM05b, BCM05c] as a way of achieving superior results by taking a weighted average of the output of the NL-means denoising and the original image. Below is an asymptotic result on the difference of the solutions of the above two cases.

Proposition 7.4.1.

$$\underline{\tilde{\mathbf{x}}_{G_e,\lambda}} - \underline{\tilde{\mathbf{x}}_{G_d,\lambda}} = \lambda\mathbf{W}^T(\mathbf{W} - \mathbf{I}_{N^2\times N^2})\underline{\mathbf{u}} + O(\lambda^2) \qquad \textit{as } \lambda \to 0. \tag{7.25}$$

Proof. Note that as $\lambda \to 0$, $\underline{\tilde{\mathbf{x}}_{G_e,\lambda}} = \lambda \mathbf{W}\underline{\mathbf{u}} + (1-\lambda)\underline{\mathbf{u}} + O(\lambda^2)$. Also,

$$\begin{aligned} \underline{\tilde{\mathbf{x}}_{G_d,\lambda}} &= [\mathbf{I}_{N^2\times N^2} + \lambda(\mathbf{W} - \mathbf{I}_{N^2\times N^2})^T(\mathbf{W} - \mathbf{I}_{N^2\times N^2})]^{-1}\underline{\mathbf{u}} \\ &= [\mathbf{I}_{N^2\times N^2} - \lambda(\mathbf{W} - \mathbf{I}_{N^2\times N^2})^T(\mathbf{W} - \mathbf{I}_{N^2\times N^2})]\underline{\mathbf{u}} + O(\lambda^2). \end{aligned} \tag{7.26}$$

Hence,

$$\begin{aligned} \underline{\tilde{\mathbf{x}}_{G_e,\lambda}} - \underline{\tilde{\mathbf{x}}_{G_d,\lambda}} &= \lambda \mathbf{W}\underline{\mathbf{u}} + (1-\lambda)\underline{\mathbf{u}} - [\mathbf{I}_{N^2\times N^2} - \lambda(\mathbf{W} - \mathbf{I}_{N^2\times N^2})^T(\mathbf{W} - \mathbf{I}_{N^2\times N^2})]\underline{\mathbf{u}} + O(\lambda^2) \\ &= \lambda \mathbf{W}\underline{\mathbf{u}} + \underline{\mathbf{u}} - \lambda\underline{\mathbf{u}} - \underline{\mathbf{u}} + \lambda\mathbf{W}^T\mathbf{W}\underline{\mathbf{u}} - \lambda\mathbf{W}\underline{\mathbf{u}} - \lambda\mathbf{W}^T\underline{\mathbf{u}} + \lambda\underline{\mathbf{u}} + O(\lambda^2) \\ &= \lambda\mathbf{W}^T\mathbf{W}\underline{\mathbf{u}} - \lambda\mathbf{W}^T\underline{\mathbf{u}} + O(\lambda^2) \\ &= \lambda\mathbf{W}^T(\mathbf{W} - \mathbf{I}_{N^2\times N^2})\underline{\mathbf{u}} + O(\lambda^2). \end{aligned} \tag{7.27}$$

□

7.5 A Contractive Operator Associated with NL-means Denoising

It is interesting to investigate the consequence of iterating the NL-means denoising operator $\mathbf{W}\underline{\mathbf{x}}$ on images. In what follows, we show that $\mathbf{W}$ is not a projection operator, i.e, $\mathbf{W}^2\underline{\mathbf{x}}$ is not necessarily equal to $\mathbf{W}\underline{\mathbf{x}}$ in general.

Proposition 7.5.1. *In general,* $\mathbf{W}$ *is not a projection operator, in which case* $\lim_{k\to\infty} \mathbf{W}^k\underline{\mathbf{x}}$ *is a constant-valued image.*

Proof. Note that $\mathbf{W}$ is a square matrix whose rows consists of nonnegative real numbers, with each row summing to 1, i.e., $\mathbf{W}$ is a right stochastic matrix. The Perron-Frobenius theorem for the right stochastic matrix $\mathbf{W}$ ensures that there exists a stationary probability vector $\underline{\mathbf{p}}$ such that

$$\underline{\mathbf{p}}\mathbf{W} = \underline{\mathbf{p}}. \tag{7.28}$$

The j-th element of the vector $\underline{\mathbf{p}}$ may be computed by taking the limit

$$\lim_{k\to\infty} (\mathbf{W}^k)_{(i,j)} = \underline{\mathbf{p}}_j. \tag{7.29}$$

Hence, independent of i,

$$\begin{aligned} \forall i \quad (\lim_{k\to\infty} \mathbf{W}^k\underline{\mathbf{x}})_i &= \lim_{k\to\infty} \sum_j (\mathbf{W}^k)_{(i,j)}\underline{\mathbf{x}}_j \\ &= \sum_j \lim_{k\to\infty} (\mathbf{W}^k)_{(i,j)}\underline{\mathbf{x}}_j \\ &= \sum_j \underline{\mathbf{p}}_j\underline{\mathbf{x}}_j \\ &= \underline{\mathbf{p}}\underline{\mathbf{x}} = c. \end{aligned} \tag{7.30}$$

Therefore,

$$\lim_{k\to\infty} \mathbf{W}^k \underline{\mathbf{x}} = (c, c, \ldots, c)^T \tag{7.31}$$

is a constant-valued image. If $\mathbf{W}$ is a projection operator then

$$\mathbf{W}\underline{\mathbf{x}} = \mathbf{W}^n \underline{\mathbf{x}} = \lim_{k\to\infty} \mathbf{W}^k \underline{\mathbf{x}} = (c, c, \ldots, c)^T, \text{ for any } n \geq 1, \tag{7.32}$$

which does not hold in general. Hence, $\mathbf{W}$ is not a projection operator. □

We now define a contractive denoising operator, $\underline{S}$, which acts on $\underline{\mathbf{x}}$ by producing a linear combination of the original image, $\underline{\mathbf{u}}$, and $\mathbf{W}\underline{\mathbf{x}}$, which is a denoised copy of $\underline{\mathbf{x}}$ under $\mathbf{W}$:

$$\underline{S}(\underline{\mathbf{x}}) = (1-\eta)\underline{\mathbf{u}} + \eta\mathbf{W}\underline{\mathbf{x}}. \tag{7.33}$$

The following proposition suggests that under certain conditions $\underline{S}$ is a contraction, implying the existence of a unique fixed point $\underline{\mathbf{u}}_{\underline{S}}^\star = \underline{S}(\underline{\mathbf{u}}_{\underline{S}}^\star)$.

Proposition 7.5.2. *S is a contraction on $\ell^2(\Psi)$ if $|\eta|\|\mathbf{W}\|_{2-norm} < 1$, in which case its unique fixed point is given by*

$$\underline{\mathbf{u}}_{\underline{S}}^\star = (1-\eta)(\mathbf{I}_{N^2\times N^2} - \eta\mathbf{W})^{-1}\underline{\mathbf{u}}. \tag{7.34}$$

Proof.

$$\|\underline{S}(\underline{\mathbf{x}}) - \underline{S}(\underline{\mathbf{y}})\| = |\eta|\|\mathbf{W}(\underline{\mathbf{x}} - \underline{\mathbf{y}})\| \leq |\eta|\|\mathbf{W}\|_{2-norm} \cdot \|(\underline{\mathbf{x}} - \underline{\mathbf{y}})\|. \tag{7.35}$$

It follows that $\underline{S}$ is a contraction if $|\eta|\|\mathbf{W}\|_{2-norm} < 1$. Its fixed point $\underline{\mathbf{u}}_{\underline{S}}^\star = \underline{S}(\underline{\mathbf{u}}_{\underline{S}}^\star)$ must satisfy the equation

$$(1-\eta)\underline{\mathbf{u}} + \eta\mathbf{W}\underline{\mathbf{x}}_s^\star = \underline{\mathbf{x}}_s^\star. \tag{7.36}$$

Solving for $\underline{\mathbf{u}}_{\underline{S}}^\star$ yields the result

$$\underline{\mathbf{u}}_{\underline{S}}^\star = (1-\eta)(\mathbf{I}_{N^2\times N^2} - \eta\mathbf{W})^{-1}\underline{\mathbf{u}}. \tag{7.37}$$

□

It is now interesting to consider the difference between $\underline{\mathbf{u}}_{\underline{S}}^\star$ and

$$\underline{S}(\underline{\mathbf{u}}) = \eta\mathbf{W}\underline{\mathbf{u}} + (1-\eta)\underline{\mathbf{u}}. \tag{7.38}$$

Note that $\underline{S}(\underline{\mathbf{u}})$ is the same as $\underline{\tilde{\mathbf{x}}_{G_e,\lambda}}$ defined previously in Equation 7.24 if $\eta = \frac{\lambda}{1+\lambda}$.

Proposition 7.5.3.

$$\underline{S}(\underline{\mathbf{u}}) - \underline{\mathbf{u}}_{\underline{S}}^\star = \eta\Big(\sum_{n=1}^{\infty} \eta^n \mathbf{W}^n\Big)(\mathbf{I}_{N^2\times N^2} - \mathbf{W})\underline{\mathbf{u}}. \tag{7.39}$$

Proof.

$$\begin{aligned}
\underline{S}(\underline{\mathbf{u}}) - \underline{\mathbf{u}}^{\star}_{S} &= \Big(\eta \mathbf{W}\underline{\mathbf{u}} + (1-\eta)\underline{\mathbf{u}}\Big) - (1-\eta)(\mathbf{I}_{N^2\times N^2} - \eta\mathbf{W})^{-1}\underline{\mathbf{u}} \\
&= \eta\mathbf{W}\underline{\mathbf{u}} + (1-\eta)\underline{\mathbf{u}} - (1-\eta)\underline{\mathbf{u}} - (1-\eta)\Big(\sum_{n=1}^{\infty}\eta^n\mathbf{W}^n\Big)\underline{\mathbf{u}} \\
&= \eta\mathbf{W}\underline{\mathbf{u}} - \eta\mathbf{W}\underline{\mathbf{u}} - \sum_{n=2}^{\infty}\eta^n\mathbf{W}^n\underline{\mathbf{u}} + \sum_{n=1}^{\infty}\eta^{n+1}\mathbf{W}^n\underline{\mathbf{u}} \\
&= -\sum_{n=2}^{\infty}\eta^n\mathbf{W}^n\underline{\mathbf{u}} + \sum_{n=1}^{\infty}\eta^{n+1}\mathbf{W}^n\underline{\mathbf{u}} \\
&= \Big(\sum_{n=1}^{\infty}\eta^{n+1}(-\mathbf{W}^{n+1} + \mathbf{W}^n)\Big)\underline{\mathbf{u}} \\
&= \eta\Big(\sum_{n=1}^{\infty}\eta^n\mathbf{W}^n\Big)(\mathbf{I}_{N^2\times N^2} - \mathbf{W})\underline{\mathbf{u}}. \qquad (7.40)
\end{aligned}$$

□

Due to the fact that matrix $\mathbf{W}$ in the NL-means regularization schemes outlined in Sections 7.4 and 7.5 are non-sparse and very large, computation of $\tilde{\mathbf{x}}_{G_d,\lambda}$ and $\mathbf{u}^{\star}_{S}$ seems infeasible.

7.6 Concluding Remarks

In this chapter, we have introduced a set of regularization functions based on the self-similarity of images for the purpose of image denoising and image zooming. These regularization functions are based on existing schemes that exploit the self-similarity of images, namely fractal image coding and nonlocal-means image denoising. Some analytic asymptotic results have also been presented. Furthermore, some computational results of image zooming and denoising using the fractal-based regularization schemes have also been presented.

By no means do we believe that the schemes introduced in this chapter will replace existing efficient methods of denoising and zooming. Our method, however, introduces the possibility of using self-similarity-based priors in some imaging problems that may well be combined with other existing prior information about the image, possibly producing improved results. We also believe that the ideas introduced here could be further investigated for the case of more general degradation operators.

Chapter 8

Role of Scale in the Context of the Non-Local-Means Filter

8.1 Introduction

Natural images exhibit various *regularity properties* such as the degree of smoothness, total variation, decay as well as sparsity of the transform domain coefficients, which have been exploited in various image processing tasks, including compression [Mal98, MZ92]. Many of these regularity properties are *local*, in the sense that the greyscale value at a pixel is correlated with values in its neighbourhood.

Self-similarity is an example of a *nonlocal* regularity property, in the sense that local neighbourhoods of an image can be highly correlated (i.e., affinely similar) to other neighbourhoods throughout the image. Self-similarity is the basis of fractal image coding [Bar88, Lu97, Fis95] which, historically, concentrated on image compression. Fractal coding has also been shown to be effective in other image processing tasks, such as denoising [GFV03, Ale05]. More recently, the translational self-similarity of images has been exploited for the purpose of denoising [BCM05b, BCM05c].

The essence of the fractal transform is to approximate smaller *range* subblocks of an image with modified copies of larger *domain* subblocks [Bar88, Lu97, Fis95]. The underlying basis for the success of these methods is that of *approximability*: "How well can an image be approximated using self-similarity constraints?" Equivalently, this may be viewed in terms of *regularity*: "How regular or invariant is an image under the imposition of self-similarity constraints?" As mentioned earlier, some recent investigations [Ale05, AVT08] have shown that images generally possess a great deal of local (affine) self-similarity.

The non-local-means (NL-means) filter [BCM05b, BCM05c] performs denoising using the *same-scale* translational redundancy of information in an image. This is done by a weighted averaging process, where the weights are determined by the similarities of neighbourhoods, or "patches." The efficacy of the NL-means as a denoising method is again an issue of *approximability and regularity*.

In this chapter, we examine the self-similarity properties of an image by formulating a simple *cross-scale* variant of the NL-means and comparing its effectiveness to that of the standard *same-scale* NL-means method. The former will use neighbourhoods/patches that are taken from the same image, or even from another image, *at a different scale*. Such a role of scale for the NL-means filter, has not yet been investigated.

In Section 8.2, we introduce the background material required for this chapter. We also introduce the cross-scale variant of the NL-means which will be used in our experiments. A rather simple statistical examination of patch-similarity at two-scales for various images is given in Section 8.3. In Sections 8.4 and 8.5, we examine the role of scale for a couple of images in the context of the NL-means filter. In Section 8.4, the patches are taken from the observed image, i.e., "self-examples", whereas in Section 8.5, they are taken from another image. Some concluding remarks are presented in Section 8.6.

8.2 Some New Variants of the NL-means Filter

8.2.1 An Example-based Variant of the NL-means Filter

In this section, we introduce an extension of the NL-means method which employs neighbourhoods taken from an *example image*, $\mathbf{v}$.

Example image: The notion of *example image* is denoted by $\mathbf{v} \in \ell^2(\Phi)$, where Φ is the $K \times L$ pixel lattice defined by

$$\Phi = [1, \ldots, K] \times [1, \ldots, L].$$

Algorithm 8.2.1. ***An Example-based Variant of the NL-means Filter***
Consider the denoising problem $\mathbf{u} = \mathbf{x} + \mathbf{n}$. *The approximation of* $\mathbf{x}$, *denoted by* $\widetilde{\mathbf{x}}$, *given the observation* $\mathbf{u}$, *using* example image $\mathbf{v}$, *is computed in the following way. For any* $x \in \Omega$,

$$\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v},h,d,a)}(x) = \frac{1}{C(x)} \sum_{y \in \Phi} w(x,y)\mathbf{v}(y), \tag{8.1}$$

where

$$w(x,y) = \exp\Big(- \frac{\| \mathbf{u}(\mathcal{N}^d\{x\}) - \mathbf{v}(\mathcal{N}^d\{y\}) \|_{2,a}^2}{h^2} \Big)$$

and

$$C(x) = \sum_{y \in \Phi} w(x,y). \tag{8.2}$$

It is clear that NL-means image denoising, introduced in Section 3.6.1, is equivalent to Algorithm 8.2.1 in the case that $\mathbf{v} = \mathbf{u}$, so that the approximation is $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{u},h,d,a)}(x)$.

In a manner similar to the restricted search-window algorithm, introduced in Subsection 3.6.1, we may speed up this algorithm by restricting y. Due to the fact that the size of the example image $\mathbf{v}$ is $K \times L$ and not necessarily identical to the size of $\mathbf{u}$, $M \times N$, we consider a corresponding pixel to x in $\mathbf{v}$ called x_{map} defined as $x_{map} = (\lfloor Ki/M \rfloor, \lfloor Lj/N \rfloor])$, in which $x = (i,j)$, and $\lfloor \ldots \rfloor$ denotes the floor function. Therefore, we define $y \in \Phi \cap \mathcal{N}^r\{x_{map}\}$, i.e., in a neighbourhood of x_{map} rather than the whole field of Φ in the corresponding equations above.

8.2.2 Projection Operator to a Coarser Scale Image

We now consider the approximation of images yielded by the NL-means filter at varying scales. To do so, we first need to remember the precise definition of a projection to a coarser scale defined in Section 1.4. Such projection, will be denoted as $\mathcal{D}_z$. Let us define $\mathcal{D}_z : \ell^2(\Omega) \to \ell^2(\Omega)$ to be $\mathcal{D}_z = \mathcal{S}_z \circ \mathcal{A}_z$, where $\mathcal{A}_z : \ell^2(\Omega) \to \ell^2(\Omega)$ is the local averaging operator of length z: For any $(i,j) \in \Omega$,

$$(\mathcal{A}_z\mathbf{x})(i,j) = \frac{1}{z^2} \sum_{0 \le i' < z, 0 \le j' < z} \mathbf{x}(i+i', j+j'). \tag{8.3}$$

(Note that boundary conditions on $\mathbf{x}$ may be required so that $\mathcal{A}_z$ is well-defined.) Also, the down-sampling operator $\mathcal{S}_z : \ell^2(\Omega) \to \ell^2(\Omega)$ is defined for any image $\mathbf{x} \in \ell^2(\Omega)$ such that for any $(i, j) \in \Omega$,

$$(\mathcal{S}_z \mathbf{x})(i, j) = \mathbf{x}\Big((i-1)z + 1, (j-1)z + 1\Big). \tag{8.4}$$

In the experiments presented throughout the rest of the chapter, we shall be using $\mathcal{D}_2$, i.e., the case that $z = 2$.

8.3 Distribution of Neighbourhood Distances at Various Scales

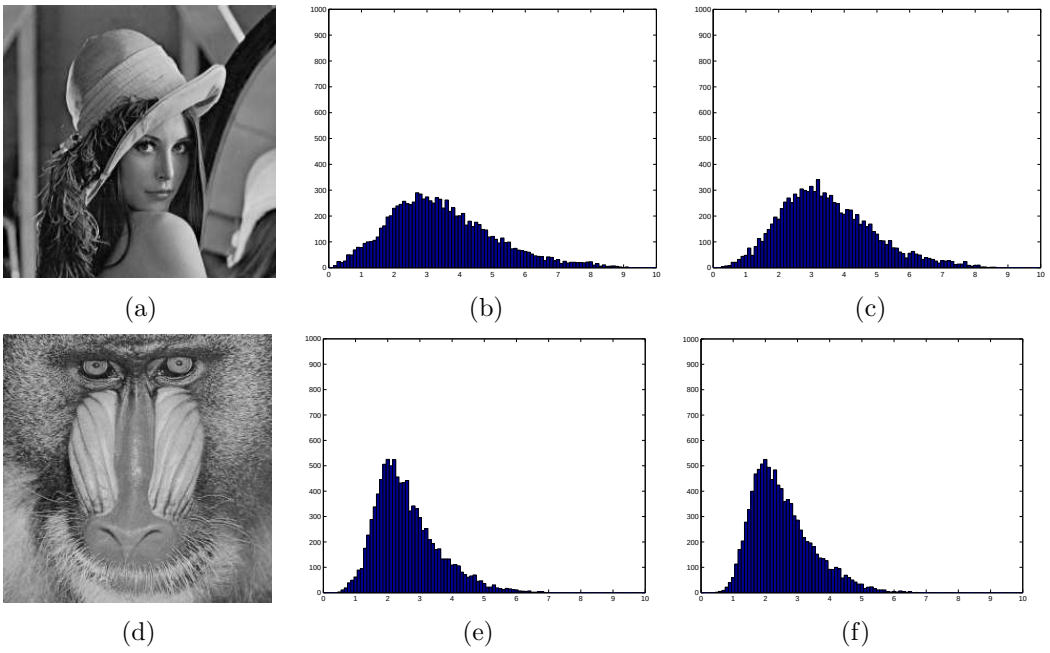


Figure 8.1: (a) and (d) are the 256×256 test images. (b) and (e) are plots of the distributions of $\| \mathbf{u}(\mathcal{N}^6\{x\}) - \mathbf{u}(\mathcal{N}^6\{y\}) \|$ corresponding to the images in (a) and (d), respectively. The plots in (c) and (f) are the distributions of $\| \mathbf{u}(\mathcal{N}^6\{x\}) - (\mathcal{D}_2\mathbf{u})(\mathcal{N}^6\{y\}) \|$ for the images in (a) and (d), respectively.

As mentioned earlier, some recent investigations [Ale05, AVT08] in the context of fractal-image coding have shown that images generally possess a great deal of local (affine) self-similarity: Given a subimage $\mathbf{u}|_{C_i}$ there are often a good number of domain blocks P_j whose subimages $\mathbf{u}|_{P_j}$ approximate it roughly as well as the "best" domain block. This feature, which never seems to have been quantified previously, accounts for the rather small degradations that are experienced when the size of the *domain pools* – the domain blocks P_j to be examined – is decreased.

In a parallel fashion, the NL-means algorithm relies not only on the intensity values of pixels but also on the neighbourhood (or patch) similarities existing in an image. Below, we present

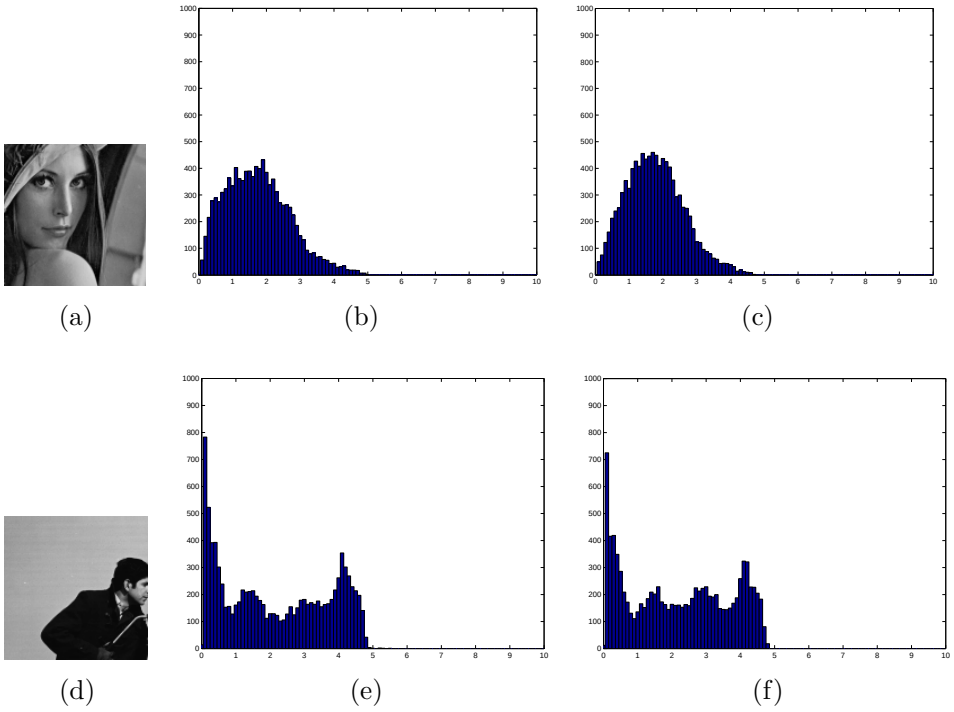


Figure 8.2: (a) and (d) are the 128×128 test images. (b) and (e) are plots of the distributions of $\| \mathbf{u}(\mathcal{N}^3\{x\}) - \mathbf{u}(\mathcal{N}^3\{y\}) \|$ corresponding to the images in (a) and (d), respectively. The plots in (c) and (f) are the distributions of $\| \mathbf{u}(\mathcal{N}^3\{x\}) - (\mathcal{D}_2\mathbf{u})(\mathcal{N}^3\{y\}) \|$ for the images in (a) and (d), respectively.

experiments to evaluate same-scale and cross-scale similarities of an image simply under the ℓ^2 distance.

In Figures 8.1(a) and 8.1(d) are shown the test images used here: 256×256 pixel, 8 bits per pixel, *Lena* and *Mandrill* images for which the intensity values are rescaled to the interval $[0, 1]$. We have chosen $10,000$ randomly selected pairs of 13×13 pixel-blocks from each image and have plotted the distribution of the ℓ^2 distances of these blocks on the left hand-side. More precisely, Figures 8.1(b) and 8.1(e) show plots of the distributions of $\| \mathbf{u}(\mathcal{N}^6\{x\}) - \mathbf{u}(\mathcal{N}^6\{y\}) \|$ for the images in Figures 8.1(a) and 8.1(d), respectively. Here, we have performed the experiment for $10,000$ uniformly-randomly selected pairs of (x, y). The plots in Figures 8.1(c) and 8.1(f) are the distributions of $\| \mathbf{u}(\mathcal{N}^6\{x\}) - (\mathcal{D}_2\mathbf{u})(\mathcal{N}^6\{y\}) \|$ for the images in Figures 8.1(a) and 8.1(d), respectively. As a result, the histograms of Figures 8.1(b) and 8.1(e) correspond to *same-scale* patch comparisons, while those in Figures 8.1(c) and 8.1(f) correspond to *cross-scale* patch comparisons.

Similarly, Figures 8.2(a) and 8.2(d) show 8 bit-per-pixel 128×128 pixel test images, *Lena* and *Cameraman* for which the intensity values are rescaled to the interval $[0, 1]$. From these images, we have chosen $10,000$ randomly-selected pairs of 7×7 pixel-blocks. Figures 8.2(b) and 8.2(e) show plots of the distributions of $\| \mathbf{u}(\mathcal{N}^3\{x\}) - \mathbf{u}(\mathcal{N}^3\{y\}) \|$ corresponding to the images in 8.2(a) and 8.2(d), respectively. Similarly, the plots in Figures 8.2(c) and 8.2(f) are the distributions of the values $\| \mathbf{u}(\mathcal{N}^3\{x\}) - (\mathcal{D}_2\mathbf{u})(\mathcal{N}^3\{y\}) \|$ corresponding to the images in 8.2(a) and 8.2(d), respectively.

For each test image, the similarity between the histograms yielded by the two experiments, i.e., same-scale and cross-scale, is striking. This indicates that the self-similarity properties of these images at both same-scale and cross-scale are remarkably similar. This is a hint that an appropriately defined cross-scale version of the NL-means filter should behave in a similar fashion to the traditional same-scale NL-means filter. In the next section we compare these two methods.

8.4 Same-scale vs. Cross-scale Approximations

In this section, we compare the results of two approximations: the traditional, same-scale NL-means method which yields the approximation $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{u},h,d,a)}(x)$ and its cross-scale NL-means counterpart, with approximation $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathcal{D}_2(\mathbf{u}),h,d,a)}(x)$. Given an observed image $\mathbf{u}$ and using Algorithm 8.2.1, we choose the example image $\mathbf{v} = \mathbf{u}$ to obtain results of the NL-means filter. Choosing the example image $\mathbf{v} = \mathcal{D}_2(\mathbf{u})$, translates to cross-scale neighbourhood-comparison and averaging. Furthermore, we examine how these two schemes differ from each other as the smoothness parameter h is varied. In the experiments presented below, we have used Algorithm 8.2.1 with $d = 3$ and a search-window of size 21×21, i.e., $r = 10$. Furthermore, we also have chosen $G_a = 1_{\{(0,0)\}}$, i.e., simply ignoring the effect of Gaussian weighting.

Figures 8.3(a) and 8.5(a) display the two test images $\mathbf{x}$ used, while the observations $\mathbf{u}$ are shown in Figures 8.3(b) and 8.5(b). Here, the standard deviation of noise was $\sigma = 25$. Two experiments using NL-means are shown, first by taking the example image $\mathbf{v}_1 = \mathbf{u}$, i.e., the observed image itself, shown in 8.3(c) and 8.5(c), and second by taking the example image $\mathbf{v}_2 = \mathcal{D}_2(\mathbf{u})$, i.e., the

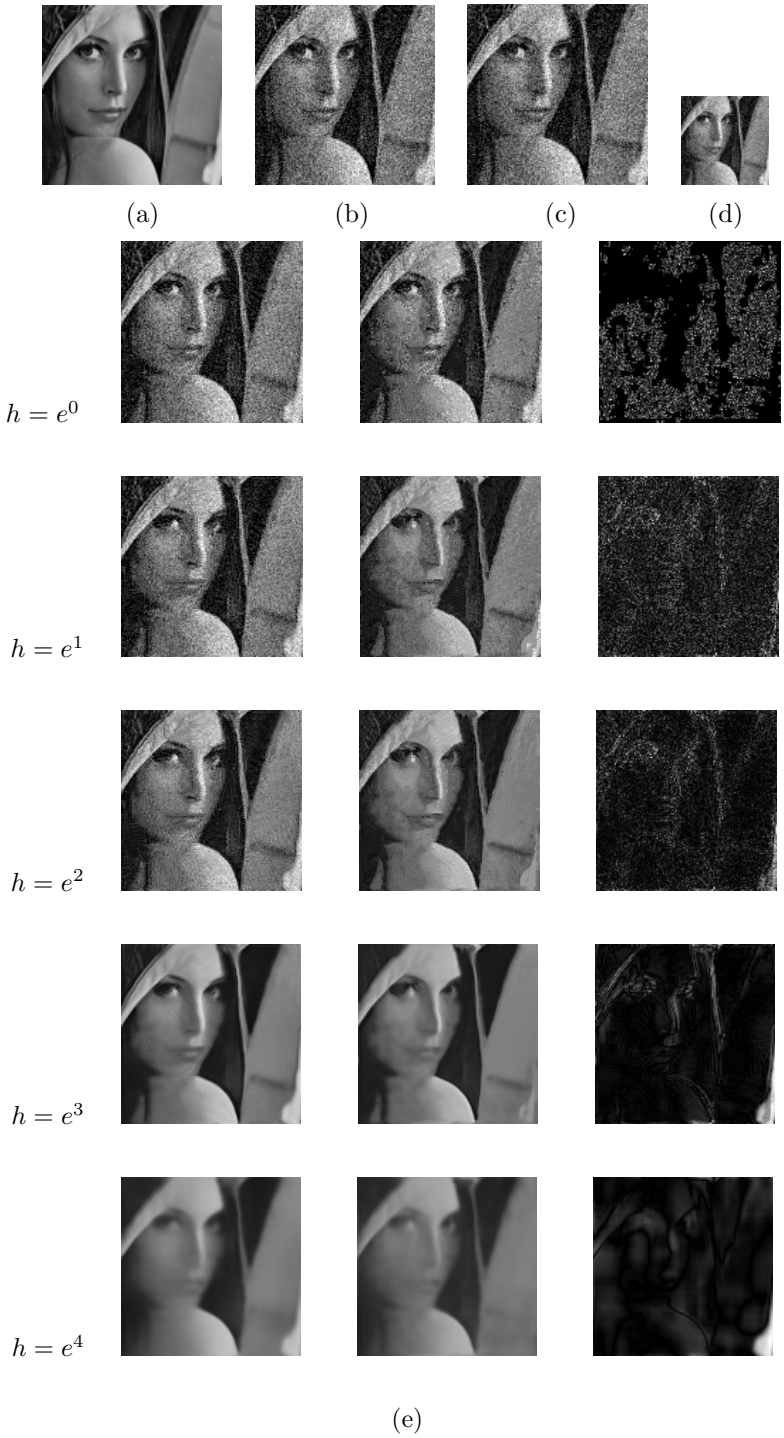


Figure 8.3: (a) shows the original image $\mathbf{x}$, (b) the observation $\mathbf{u}$, (c) $\mathbf{v}_1 = \mathbf{u}$, (d) $\mathbf{v}_2 = \mathcal{D}_2\mathbf{u}$. In (e): For each value of h are shown: $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$ (left), $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_2,h,d,a)}(x)$ (middle) and $|\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$-$\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_2,h,d,a)}(x)|$ (right).

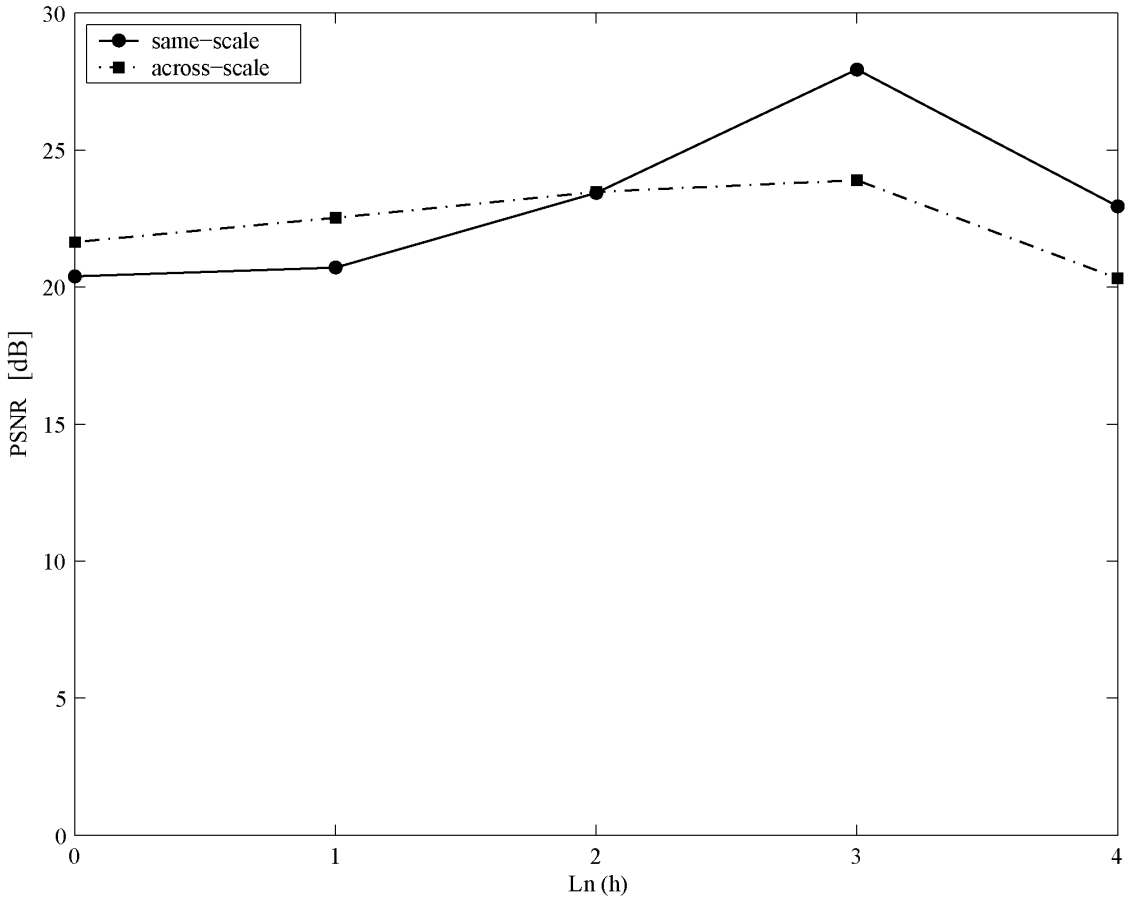


Figure 8.4: PSNR vs $\ln(h)$ for images plotted on left and in middle of Figure 8.3(e).

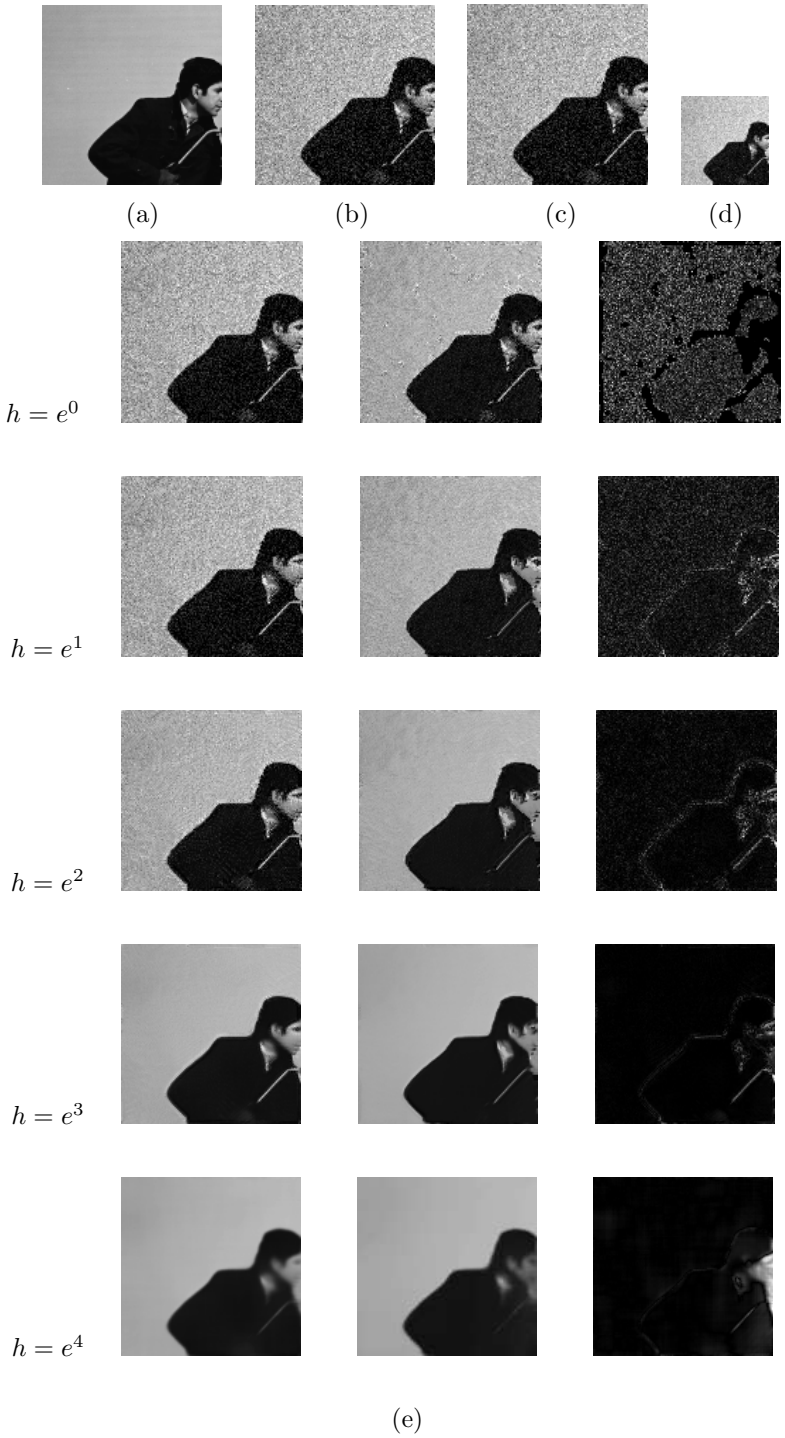


Figure 8.5: (a) shows the original image $\mathbf{x}$, (b) the observation $\mathbf{u}$, (c) $\mathbf{v}_1 = \mathbf{u}$, (d) $\mathbf{v}_2 = \mathcal{D}_2\mathbf{u}$. In (e): For each value of h are shown: $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$ (left), $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_2,h,d,a)}(x)$ (middle) and $|\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$-$\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_2,h,d,a)}(x)|$ (right).

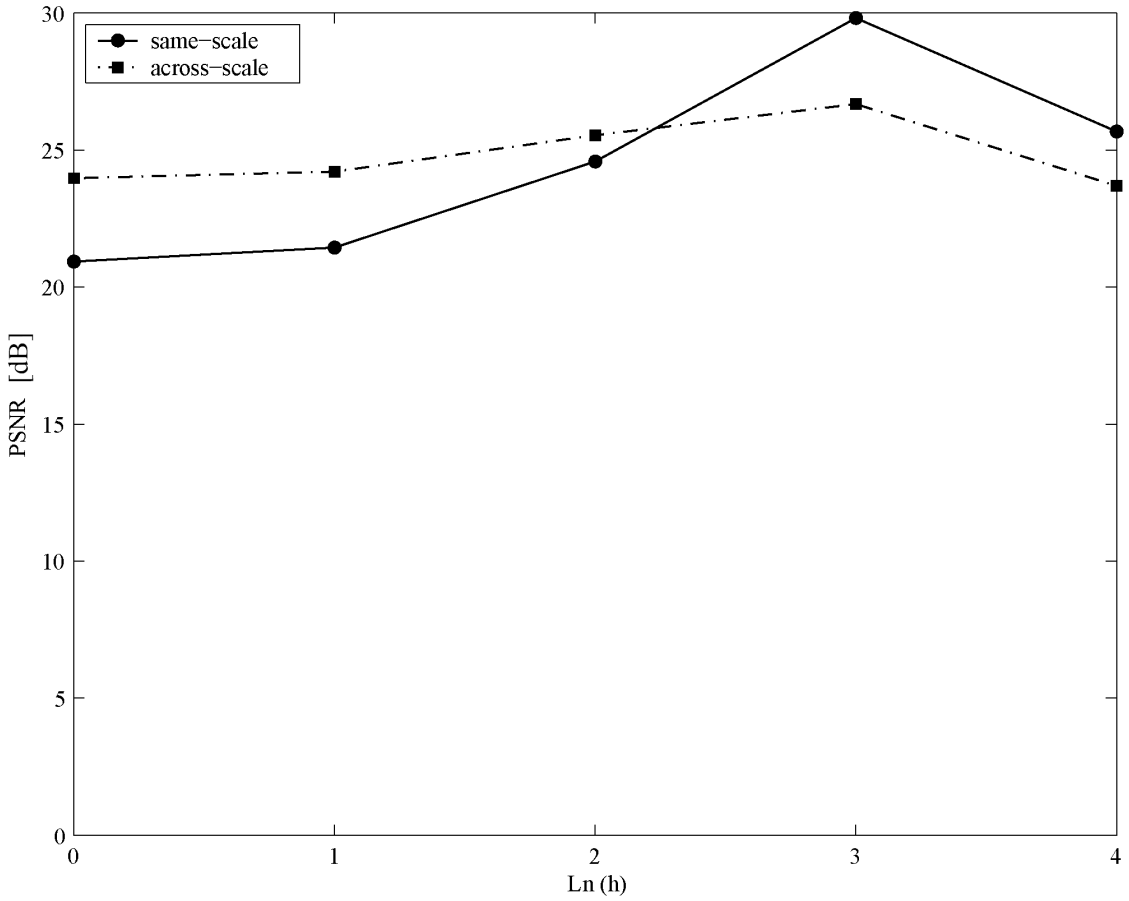


Figure 8.6: PSNR vs $\ln(h)$ for images plotted on left and in middle of Figure 8.5(e).

coarser scale approximation of the observed image, with factor 2, shown in 8.3(d) and 8.5(d). In 8.3(e) and 8.5(e), we have chosen various values of the smoothness parameter h. For each value of h, we have plotted $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{u},h,d,a)}(x)$ on the left and $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathcal{D}_2(\mathbf{u}),h,d,a)}(x)$ in the middle. On the right, for each h-value, is plotted the absolute value of the difference of these two images, $|\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{u},h,d,a)}(x)$- $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathcal{D}_2(\mathbf{u}),h,d,a)}(x)|$. Finally in Figures 8.4 and 8.6 plots of PSNR vs. $\ln(h)$ for the images plotted on the left and middle of 8.3(e) and 8.5(e) are presented. It can be observed that as the smoothness parameter h is increased from zero, the PSNR first increases for both same-scale and cross-scale experiments, and eventually decreases as h approaches infinity. We have used a logarithmic scale for h so that the decay of PSNR becomes more visible for high-values of h on the graph.

These experiments suggests that good denoising is accomplished with the use of either $\mathbf{u}$ and $\mathcal{D}_2(\mathbf{u})$ as examples, i.e, same-scale or cross-scale neighbourhoods, in the NL-means filter.

8.5 Filters Based on Irrelevant Examples and Results

We now examine the performance of Algorithm 8.2.1 in the case of irrelevant examples, i.e., examples taken from another image, both in the same-scale and cross-scale case. In what follows, we use the *Cameraman* and *Lena* images. Once again, we have used Algorithm 8.2.1 with $d = 3$ and a search-window of size 21×21, i.e., $r = 10$. Furthermore, we also have chosen $G_a = 1_{\{(0,0)\}}$, i.e., simply ignoring the effect of Gaussian weighting.

Figures 8.7(a) and 8.9(a), show the images $\mathbf{x}$ while the respective observations $\mathbf{u}$ are shown in 8.7(b) and 8.9(b). Here, once again, the standard deviation of the added noise was $\sigma = 25$. The results of two experiments using NL-means are shown. In the first experiment, we used the example image $\mathbf{v}_1$ shown in (c). In the second experiment, we used a factor-2 coarser scale version of the example image, $\mathbf{v}_2 = \mathcal{D}_2(\mathbf{v}_1)$, shown in (d). In (e), we present results for various values of h the smoothness parameter. On the left are presented the results $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$. In the middle are shown $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathcal{D}_2(\mathbf{v}_1),h,d,a)}(x)$. And on the right are shown the absolute differences of these two images $|\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$- $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathcal{D}_2(\mathbf{v}_1),h,d,a)}(x)|$. Finally in Figures 8.8 and 8.10, plots of PSNR vs. $\ln(h)$ are shown.

It can be observed that in both cases, PSNR decreases with h, due to the use of the irrelevant examples in the approximation step.

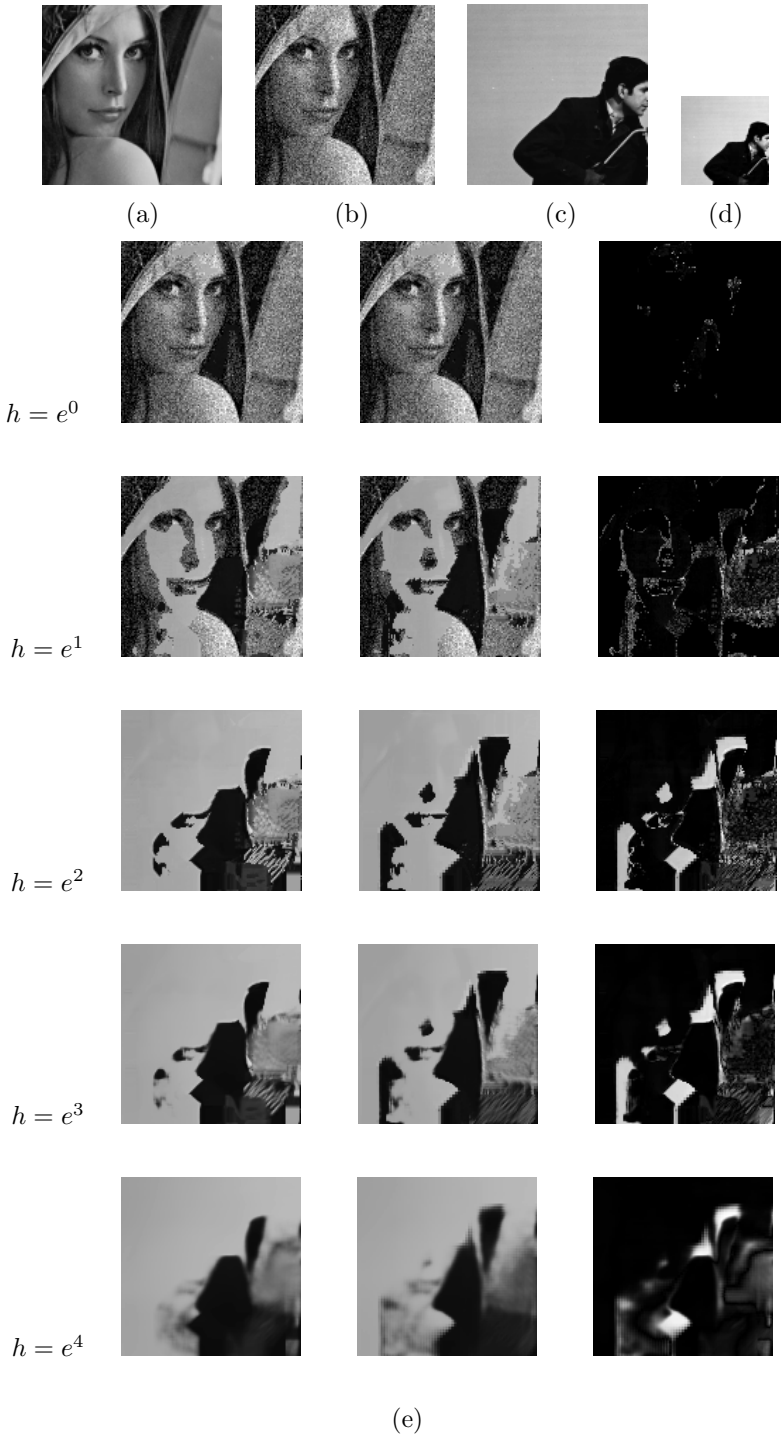


Figure 8.7: (a) contains the image $\mathbf{x}$, (b) the observation $\mathbf{u}$, (c) $\mathbf{v}_1$, (d) $\mathbf{v}_2 = \mathcal{D}_2(\mathbf{v}_1)$, (e) For each value of h, Left hand-side: $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$, middle: $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_2,h,d,a)}(x)$, right hand-side: $|\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$- $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_2,h,d,a)}(x)|$,

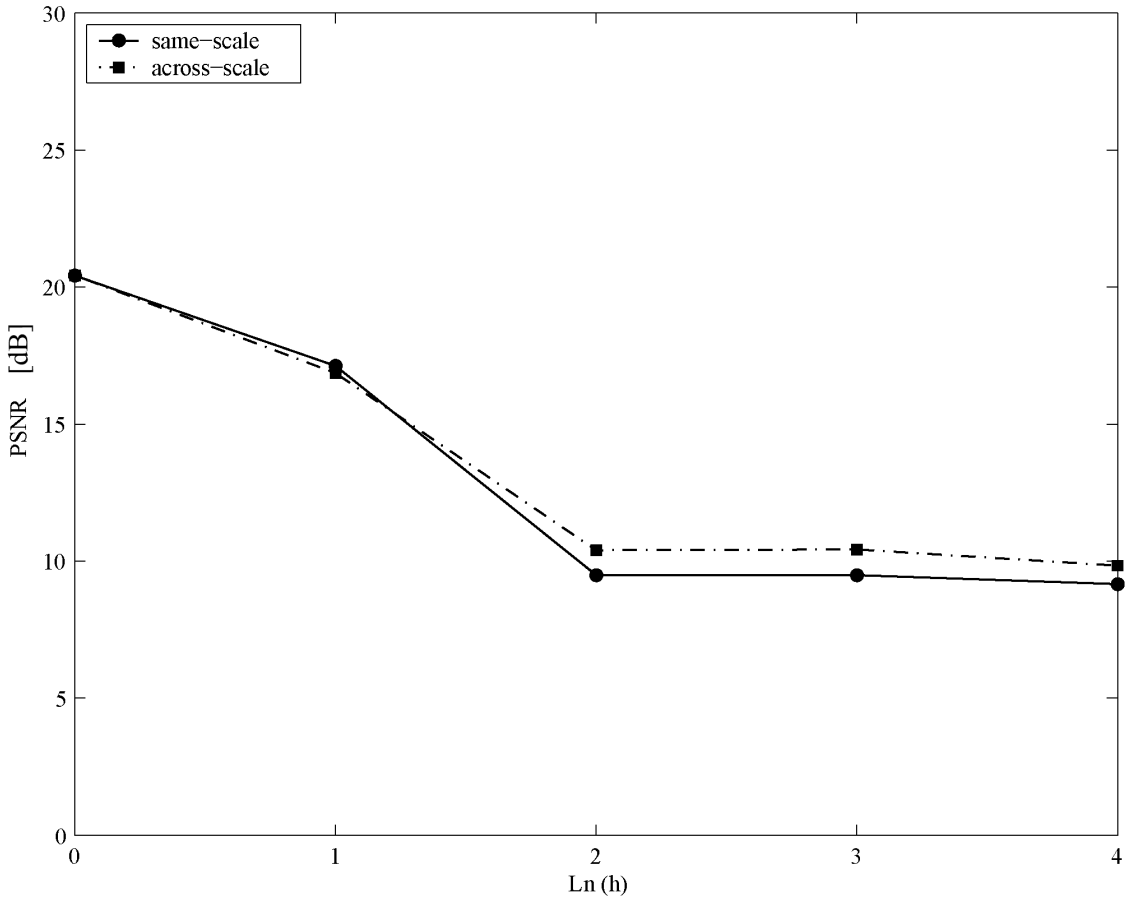


Figure 8.8: PSNR vs $\ln(h)$ for images plotted on left and in middle of Figure 8.7(e).

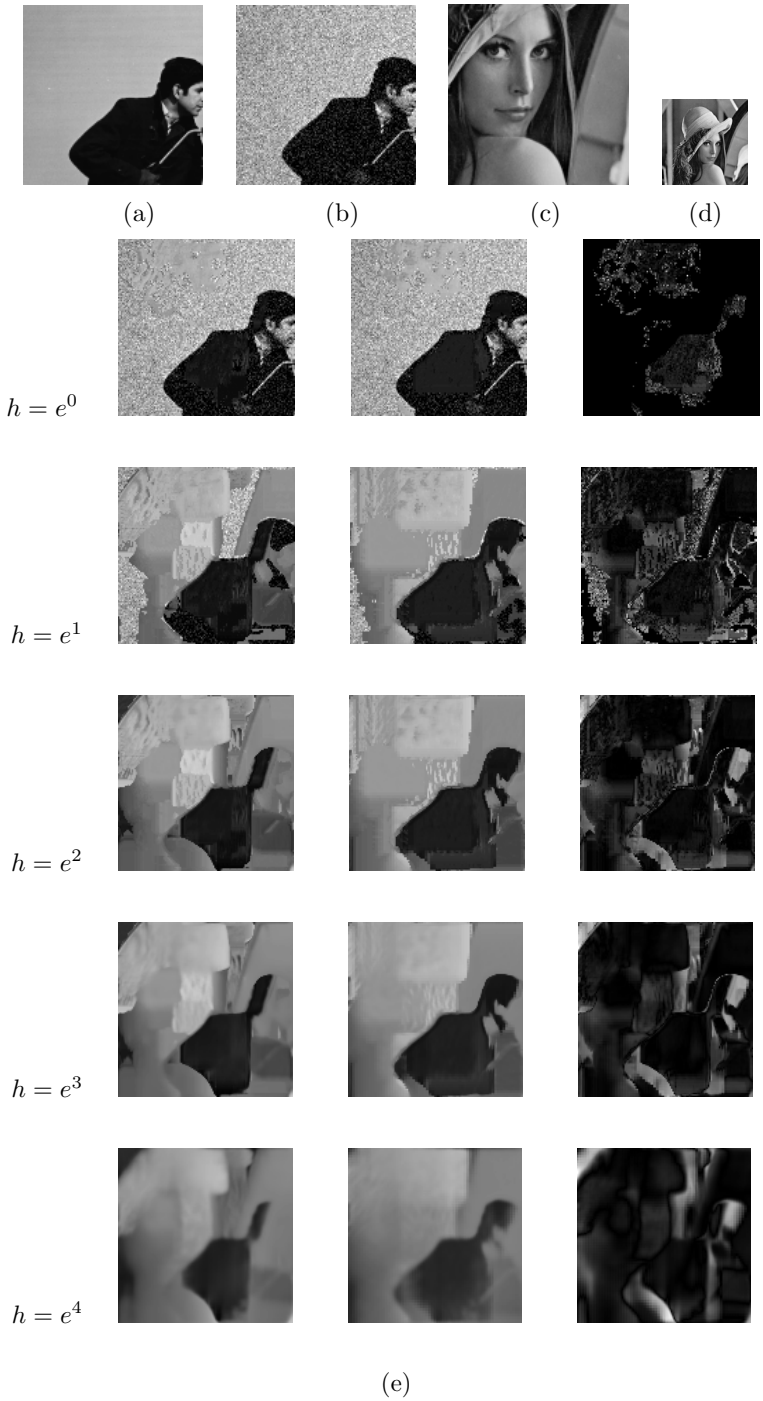


Figure 8.9: (a) shows the original image $\mathbf{x}$, (b) the observation $\mathbf{u}$, (c) $\mathbf{v}_1$, (d) $\mathbf{v}_2 = \mathcal{D}_2(\mathbf{v}_1)$, (e) For each value of h, Left hand-side: $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$, middle: $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_2,h,d,a)}(x)$, right hand-side: $|\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$- $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v}_2,h,d,a)}(x)|$.

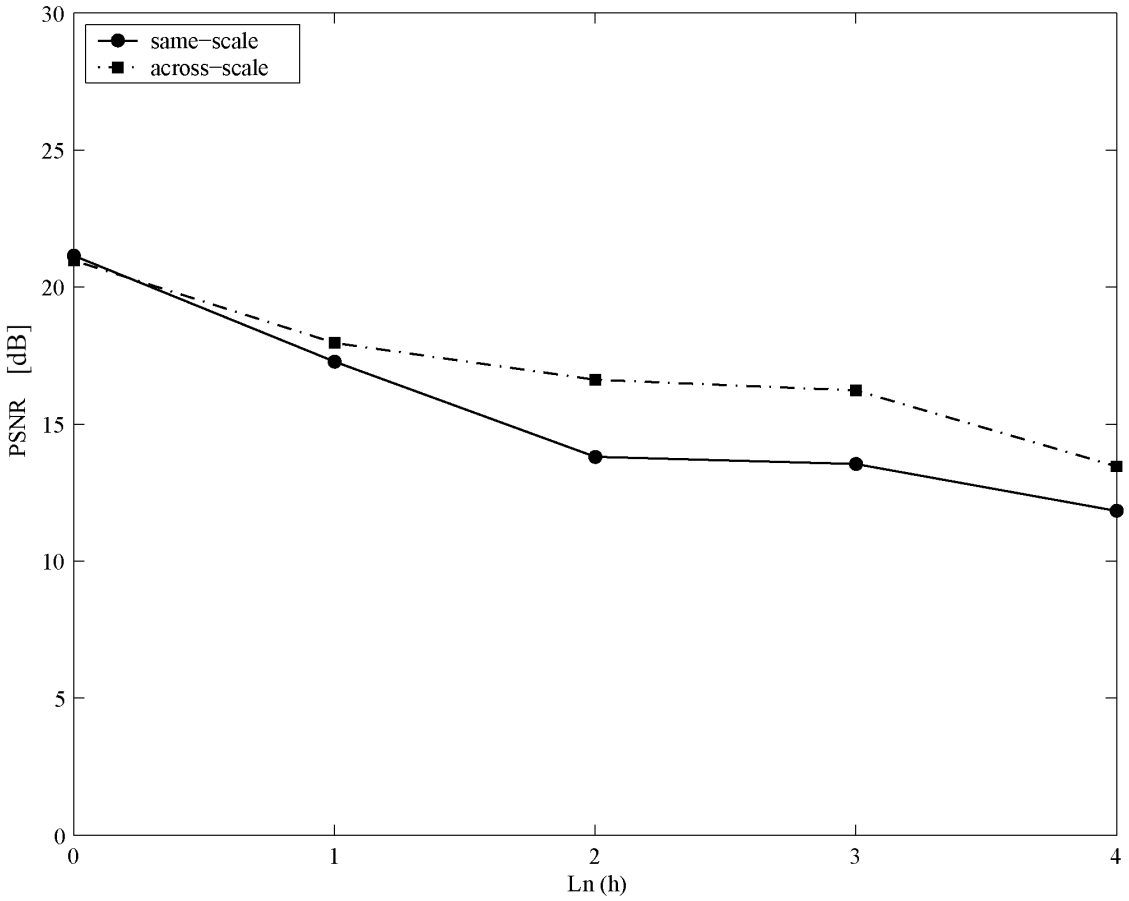


Figure 8.10: PSNR vs ln(h) for images plotted on left and in middle of Figure 8.9(e).

8.6 Concluding Remarks

In this chapter, we have examined the role of scale in the context of the NL-means filter by experimenting over various images and various values of smoothness parameter h. A new example-based variant of the NL-means is introduced and we have also examined the issue of relevant examples. It turns out that cross-scale patches are also "relevant" in approximating the image, as opposed to some "irrelevant" image, as intuitively expected. This work in some way provides a deeper insight to the question, that will be examined in Chapter 9, of why examples from cross-scales of an image can be used as a regularizer to solve the problem of single-frame image zooming using self-examples.

Chapter 9

Single-frame Image Zooming using "Self-examples"

9.1 Introduction

In this chapter, we present a novel single-frame image zooming technique based on so-called "self-examples". Our method combines the ideas of fractal-based image zooming, example-based zooming, and nonlocal-means image denoising in a consistent and improved framework. In Bayesian terms, this example-based zooming technique targets the MMSE estimate by learning the posterior directly from examples taken from the image itself at a different scale, similarly to fractal-based techniques. The examples are weighted similarly to the NL-means image denoising algorithm.

Our algorithm compares the neighbourhood information of each pixel to the neighbourhoods of the same size in the coarser scale over the entire image. Due to the reconstruction scheme, our results do not suffer from the blockiness generally inherent in fractal zooming. Moreover, the geometry of objects is well preserved for the test images presented.

In Section 9.2, we introduce the problem of image zooming, and present a brief introduction to various techniques used. In Section 9.3, we review the role of self-similarity priors in various inverse problems, namely fractal-based methods and example-based approaches. We explain how our example-based approach takes advantage of self-similarity across scales similar to fractal-based methods. In Section 9.4, we introduce our method, and discuss some computational issues in Section 9.5. Finally, some concluding remarks are presented in Section 9.6.

9.2 Some Background on the Inverse Problem of Image Zooming

9.2.1 A Word on Resolution

In the characterization of an image, the term "resolution" can be confusing since it involves a rather large number of competing terms and definitions. In its simplest form *image resolution* is defined as the smallest measurable detail in a visual presentation [Cha01].

Researchers in digital image processing and computer vision use the term resolution in three different ways, as introduced in [Cha01].

- Spatial resolution refers to the spacing of pixels in an image and is measured in pixels per inch (ppi). The higher the spatial resolution, the greater the number of pixels in the image.

  Many imaging devices today, such as charged-coupled device (CCD) cameras, consist of arrays of light detectors. A detector determines pixel intensity values depending upon the amount of light detected from its assigned area in the scene [Hol96] (see Figure 9.1). The spatial resolution of images produced by these types of devices is proportional to the density of the detector array.

- Brightness resolution refers to the number of brightness levels that can be recorded at any given pixel. For example, the brightness resolution for monochrome images is usually 256

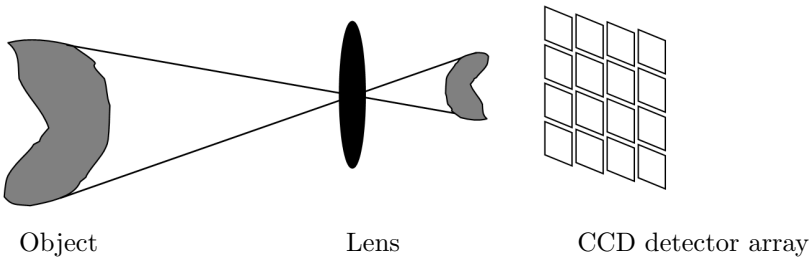


Figure 9.1: A simple CCD model.

implying that the greyscale value at a pixel is represented by 8 bits. A more appropriate term for the brightness resolution is the number of quantization levels (see Figure 9.2).

- Temporal resolution refers to the number of frames captured per second in video applications and is also commonly known as the frame rate. It is related to the amount of perceptible motion between the frames. Higher frame rates result in less smearing due to movements in the scene. The typical frame rate suitable for a pleasing view is about 25 frames per second or above.

(a) (b) (c)

Figure 9.2: (a) Original, Grey-levels: 8 bits, Size: 512 by 512, (b) Low brightness resolution, Grey-levels: 3 bits, Size: 512 by 512, (c) Low spatial resolution, Grey-levels: 8 bits, Size: 64 by 64 (replicated for larger view).

In this thesis, we only refer to the *spatial resolution* of images.

9.2.2 The Inverse Problem of Image Zooming

In many applications, the imaging sensors have poor resolution output. The process of producing a high-resolution image from a single lower-resolution and distorted (e.g., blurred, noisy) image is called *(single-frame) image zooming*.

More formally, we consider the following degradation model [Cha01, ED07, EF97]

$$\mathbf{u} = \mathcal{H}\mathbf{x} + \mathbf{n}, \tag{9.1}$$

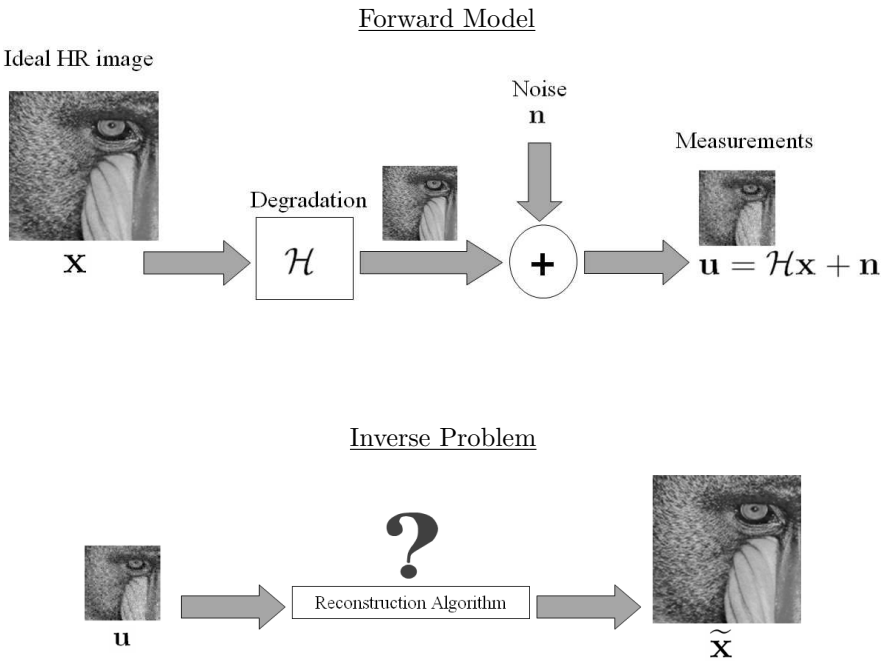


Figure 9.3: The inverse problem of image zooming.

in which $\mathbf{u} \in \ell^2(\Omega)$ is the low-resolution $M \times N$-pixel observation, i.e.,

$$\Omega = [1, \ldots, M] \times [1, \ldots, N],$$

and $\mathbf{n} \in \ell^2(\Omega)$ is additive white, independent Gaussian noise with zero-mean and variance σ^2. $\mathbf{x} \in \ell^2(\Psi)$ is the high-resolution image to be recovered such that

$$\Psi = [1, \ldots, Mz] \times [1, \ldots, Nz],$$

where z is a positive integer. The operator $\mathcal{H} = \mathcal{S}_z \mathcal{A}_z$ is the composition of a blurring operator $\mathcal{A}_z$ followed by a downsampling operator $\mathcal{S}_z$ of factor z in each direction. More technically, we remind readers that earlier, in Section 1.4, we defined the blurring operator $\mathcal{A}_z : \ell^2(\Psi) \to \ell^2(\Psi)$ as a local averaging operator of length z, i.e., for any $(i, j) \in \Psi$,

$$(\mathcal{A}_z \mathbf{x})(i, j) = \frac{1}{z^2} \sum_{0 \le i' < z, 0 \le j' < z} \mathbf{x}(i + i', j + j'). \tag{9.2}$$

Note that boundary conditions on $\mathbf{x}$ may be required so that $\mathcal{A}_z$ is well-defined. Also, the downsampling operator $\mathcal{S}_z : \ell^2(\Psi) \to \ell^2(\Omega)$ is defined for any image $\mathbf{x} \in \ell^2(\Psi)$ such that for any $(i, j) \in \Omega$,

$$(\mathcal{S}_z \mathbf{x})(i, j) = \mathbf{x}\Big((i-1)z + 1, (j-1)z + 1\Big). \tag{9.3}$$

The **problem of image zooming** can be viewed as the recovery of an estimate of $\mathbf{x}$, denoted by $\widetilde{\mathbf{x}}$, given the observation $\mathbf{u}$ and scaling factor z (see Figure 9.3).

9.2.3 A Brief Introduction to Various Techniques

Image Interpolation:

The easiest and not necessarily most appealing approach to estimate $\mathbf{x}$ is to ignore the effect of the noise $\mathbf{n}$ and the blurring operator $\mathcal{A}_z$, thereby assuming that $\mathbf{u}$ is provided by downsampling of $\mathbf{x}$, i.e., $\mathbf{u} = \mathcal{D}_z\mathbf{x}$. In this case, one could simply apply traditional interpolation algorithms on $\mathbf{u}$ to estimate $\mathbf{x}$, including *nearest neighbour or pixel replication*, *bilinear*, *bicubic*, *spline*, or *sinc interpolation*. In general, the resizing of an image does not translate into an increase in its resolution. Resizing should be accompanied by approximations to frequencies higher than those present in the observation $\mathbf{u}$ as well as at a higher signal-to-noise ratio. However some of these algorithms provide overly smoothed results around edges and fine details. The application of sharpening operators on these oversmoothed results does not usually recover the image details [Cha01].

Frequency Extrapolation Techniques:

It is well known that interpolation in the spatial domain is equivalent to extrapolation in the frequency domain. The original work on frequency extrapolation for a bandlimited image is due to Papoulis [Pap75] and Gerchberg [Ger74]. While Gerchberg proposed a method to perform signal reconstruction given the diffraction limit of the signal and a part of the spectrum, the motivation for Papoulis work was extrapolation of a bandlimited signal from the information of only a part of the original signal. The signal extrapolation is carried out by the method of alternate projections [Jai01], iterating alternately between spatial and spectral domains, which is essentially a POC-based algorithm. Any zooming technique may also be evaluated by considering the effect of the method on the Fourier coefficients.

Regularization and Classical Inverse Theory Techniques:

Single image zooming cannot recover high-frequency components that are lost during the low-resolution sampling process unless we include additional *a priori* information about the image. This can normally be done by using regularization as explained earlier in Chapter 2. According to [ED07]: "Much of the progress made in the past two decades on inverse problems in image processing can be attributed to the advances in forming or choosing the way to practise the regularization. *The effectiveness of single-image zooming techniques is always limited to the prior used in the reconstruction process.*"

Example-based Methods:

A very attractive approach in solving inverse imaging problems is to exploit examples in defining the PDF of the image instead of intuitively defining a regularization term. There are various ways to apply examples in inverse problems as comprehensively described in Chapter 2. The method

described in this chapter corresponds to making use of examples *directly* in the reconstruction procedure.

9.3 The Role of Self-similarity in Various Inverse Problems

9.3.1 Self-similarity in Various Example-based Approaches

In the work of [CPT04, EL99, WL00] on texture synthesis, and inpainting (filling in holes), examples are taken from the image *itself*.

As mentioned earlier, another important example-based approach, nonlocal-means (NL-means) image denoising [BCM05b, BCM05c] addresses the denoising problem using examples from the noisy image *itself* at the *same scale* with a Gaussian-type weighting scheme. We shall generalize this algorithm to address the zooming problem.

9.3.2 From Same-scale Toward Across-scale Self-examples

The NL-means algorithm mentioned above [BCM05b, BCM05c] along with the work reported in [EL99] use self-examples, i.e., they take examples from the image itself. However, in all cases, *the examples are taken at the same scale*. This clearly represents a major difference between these example-based methods and fractal-based methods.

In the zooming algorithms proposed below, we take advantage of the richness of the NL-means algorithm (and thereby overcome the issue of blockiness) yet remain consistent with fractal-based methods by *comparing patches across scales*.

9.4 Image Zooming Algorithm using Self-examples

In this section, we introduce our notation, and then formulate a natural extension of the NL-means denoising method, to be denoted as Algorithm 9.4.1 below. Algorithm 9.4.2 (image zooming algorithm using self-examples) will be a special case of Algorithm 9.4.1, in which the example image is precisely the input image. This formulation will detect across-scale similarities.

Example image: The notion of *example image* is denoted by $\mathbf{v} \in \ell^2(\Phi)$, where Φ is the $K \times L$ pixel lattice defined by

$$\Phi = [1, \ldots, K] \times [1, \ldots, L].$$

Lattice covers: Recall that

$$\Psi = [1, \ldots, Mz] \times [1, \ldots, Nz]$$

is the lattice on which the high-resolution image $\mathbf{x}$ was defined. Let $\mathfrak{R}$ be the partition of the lattice Ψ comprised of non-overlapping square blocks of size $z \times z$. Also, define $\mathfrak{S}$ to be the set of all $z \times z$ square blocks on the lattice Φ. We allow the elements of $\mathfrak{S}$ to overlap, implying that $\mathfrak{S}$

does not necessarily form a partition of Φ.

Lattice mapping: Let the mapping $\mathcal{M} : \mathfrak{R} \to \Omega$ be defined in the following way. For any $z \times z$ block $x \in \mathfrak{R}$ where

$$x = \Big\{(x_1, x_2) \in \mathbb{N}^2 \mid \lceil \frac{x_1}{z} \rceil = i, \lceil \frac{x_2}{z} \rceil = j, \text{ for some fixed } (i,j) \in \Omega \Big\},$$

define $\mathcal{S}_x = (i, j)$.

Extended z-neighbourhoods: For any block y in example-image-cover $y \in \mathfrak{S}$ define

$$\mathcal{N}_z^d\{y\} = \Big\{(y_1 + i_1, y_2 + i_2) \mid (y_1, y_2) \in y, (i_1, i_2) \in \mathbb{Z}^2, \max\{|i_1|, |i_2|\} \le dz \Big\}. \tag{9.4}$$

The following is a natural extension of the NL-means algorithm. In Bayesian terms the algorithm is an MMSE estimation by learning the posterior directly from the examples taken from an *example-image* $\mathbf{v}$.

Algorithm 9.4.1. ***An Example-based Zooming Algorithm*** *(see Figure 9.4)*
The approximation of $\mathbf{x}$*, denoted by* $\widetilde{\mathbf{x}}$*, given the observation* $\mathbf{u}$*, using* example image $\mathbf{v}$*, is computed in the following way. For any* $x \in \mathfrak{R}$*,*

$$\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{v},z,h,d,a)}(x) = \frac{1}{C(x)} \sum_{y \in \mathfrak{S}} w(x,y)\mathbf{v}(y), \tag{9.5}$$

where

$$w(x,y) = \exp\Big(- \frac{\| \mathbf{u}(\mathcal{N}^d\{\mathcal{M}x\}) - \mathcal{H}\big(\mathbf{v}(\mathcal{N}_z^d\{y\})\big) \|_{2,a}^2}{h^2} \Big)$$

and

$$C(x) = \sum_{y \in \mathfrak{S}} w(x,y). \tag{9.6}$$

Proposition 9.4.1. *NL-means image denoising defined in 3.6.1 is equivalent to Algorithm 9.4.1 for the case that* $z = 1$ *and* $\mathbf{v} = \mathbf{u}$*, i.e.,*

$$\widetilde{\mathbf{x}_{NL}}(x) = \widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{u},1,h,d,a)}(x). \tag{9.7}$$

Proof. If $z = 1$ and $\mathbf{v} = \mathbf{u}$ then $\Omega = \Psi = \Phi = \mathfrak{R} = \mathfrak{S}$, $\mathcal{M}x = x$, and $\mathcal{H}$ becomes the identity operator. Furthermore, in this case $\mathcal{N}_1^d\{y\} = \mathcal{N}^d\{y\}$. Substituting the corresponding expressions in Algorithm 9.4.1 leads to the equivalent expression of the NL-means algorithm. □

The benefit of our notation and the introduction of Algorithm 9.4.1 above is that a simple substitution, namely, $\mathbf{v} = \mathbf{u}$, will lead to our proposed technique of "Image zooming using self-examples".

Algorithm 9.4.2. ***Image Zooming using Self-examples*** *(see Figure 9.5)*
For a given observation $\mathbf{u}$ *and a zooming factor* z *apply Algorithm 9.4.1, using the observation in place of the example image as well, i.e.,* $\mathbf{v} = \mathbf{u}$*, and estimate* $\widetilde{\mathbf{x}}_{(\mathbf{u},\mathbf{u},z,h,d,a)}$*.*

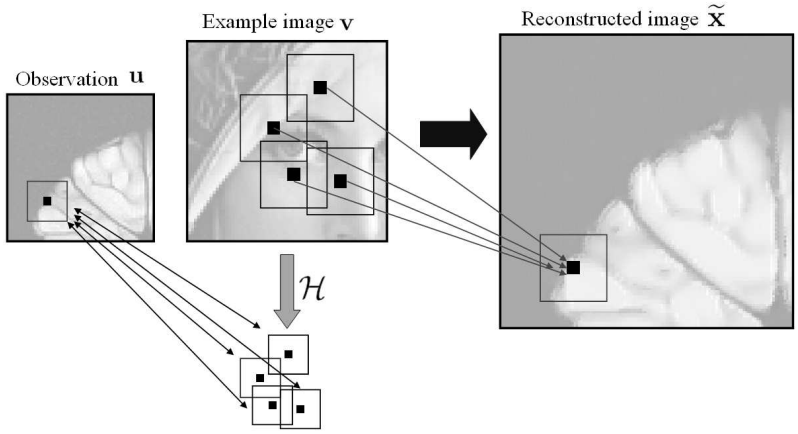


Figure 9.4: The example-based zooming Algorithm 9.4.1. First, the weights are determined based on the similarity of the patches, second, weighted averages of the $z \times z$ central blocks are replaced in the corresponding location.

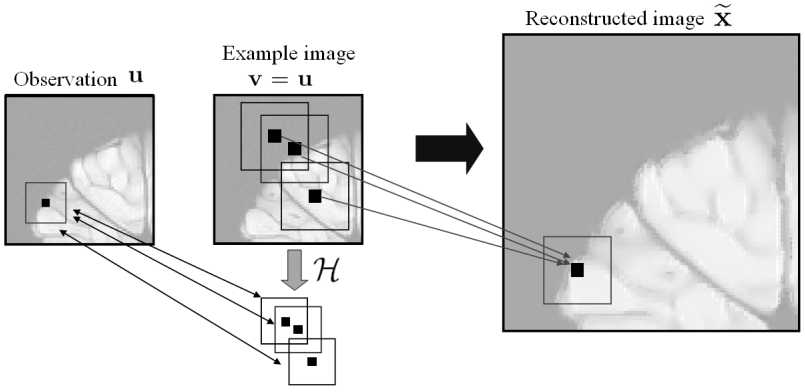


Figure 9.5: Image zooming using self-examples Algorithm 9.4.2. For a given observation, apply Algorithm 9.4.1, using the observation in place of the example image, i.e., take $\mathbf{v} = \mathbf{u}$. This translates to making use of patches taken from the observation at a different scale.

Again, it can be observed that Algorithm 9.4.2 is a special case of Algorithm 9.4.1, naturally by the constraint $\mathbf{v} = \mathbf{u}$, yet still an extension of the NL-means denoising if $z = 1$. Similarly to NL-means denoising, Algorithm 9.4.2, from a Bayesian viewpoint, targets the MMSE estimate by learning the posterior directly from the examples taken from the image itself at a different scale. As mentioned above, this is in accordance with fractal-based approaches.

9.5 Numerical Considerations and Results

9.5.1 Various Numerical Considerations

Several aspects should be carefully considered in the implementation of our algorithm. It may be necessary to apply appropriate boundary conditions, for example, when encountering neighbourhoods that fall out of the image domain or applying the operator $\mathcal{H}$ on patches. In all of our experiments we have applied symmetric boundary conditions. Furthermore, the adjustment of the parameter h varies the degree of filtering in a manner quite similar to that of NL-means denoising. A large h-value will blur the output result; a very small h-value will amplify any image noise in the output.

It should also be noticed that if finite precision is employed in our computations, then for sufficiently small h, all weighting terms in Equation 9.6 will be computed as zero. In other words, it is possible that for some $x \in \mathfrak{R}$ and for some small pre-chosen parameter $h > 0$ the weights are computed as $w(x, y) = 0$ for all values of $y \in \mathfrak{S}$. Computationally, this is not an issue in the NL-means image denoising because there is at least one patch – the neighbourhood of x itself – that makes the exponent equal to 1 regardless of the value of $h > 0$. We can bypass this problem by defining $w(x, y) = 1_{\mathfrak{S}_\mathfrak{m}}(y)$ at the problematic points $x \in \mathfrak{R}$, where $1_{\mathfrak{S}_\mathfrak{m}}(y)$ denotes the indicator function of y over the set $\mathfrak{S}_\mathfrak{m}$ defined as follows,

$$\mathfrak{S}_\mathfrak{m} = \left\{ y_s \in \mathfrak{S} \mid y_s = \arg\min_y \left(\| \mathbf{u}(\mathcal{N}^d\{\mathcal{M}x\}) - \mathcal{H}\big(\mathbf{v}(\mathcal{N}_z^d\{y\})\big) \|_{2,a} \right) \right\}.$$

This is equivalent to averaging over all $\mathbf{v}(y)$ for which y is a minimizer of the similarity distance.

9.5.2 Computational Results

We now present some results of our computational experiments. In all cases we have chosen a zoom factor $z = 2$, and 3×3 neighbourhoods as well as 6×6 extended-neighbourhoods, i.e., $d = 1$. We also have chosen $G_a = 1_{\{(0,0)\}}$, i.e., simply ignoring the effect of Gaussian weighting. As well, we have varied the parameter h with respect to each image individually.

Figure 9.6 indicates the result of an experiment using Algorithm 9.4.1 to show that using irrelevant examples (middle column) may lead to "poor" outputs (right column). In Figure 9.7, we present the outputs of Algorithm 9.4.2 for a noiseless (top left) and noisy (bottom left) image. It can be seen that the algorithm performs denoising in parallel with zooming. For purposes of comparison, the results of pixel replication and bilinear interpolation are also presented.

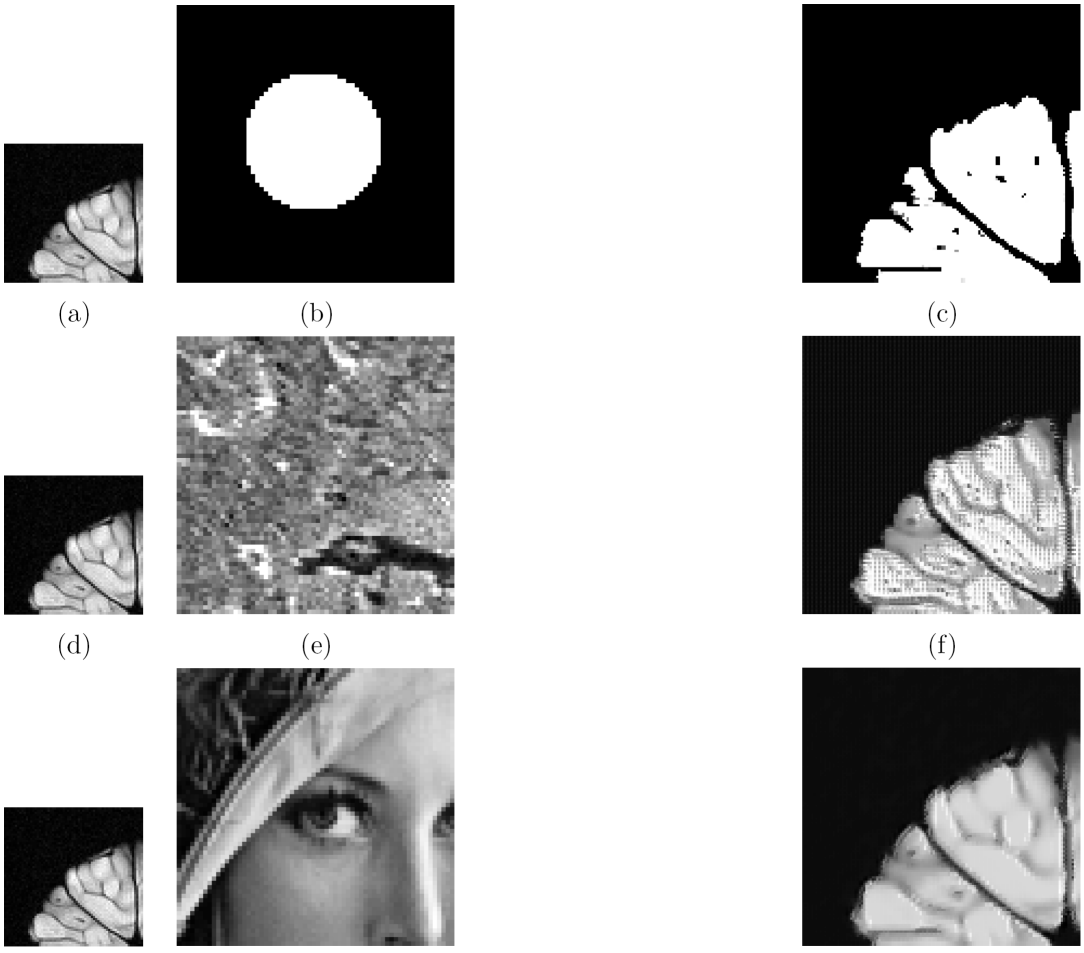


Figure 9.6: Respectively, (a,d,g) Original image. (b,e,h) Example image. (c,f,i) The image reconstructed using Algorithm 9.4.1, $h = 25$ in all three cases.

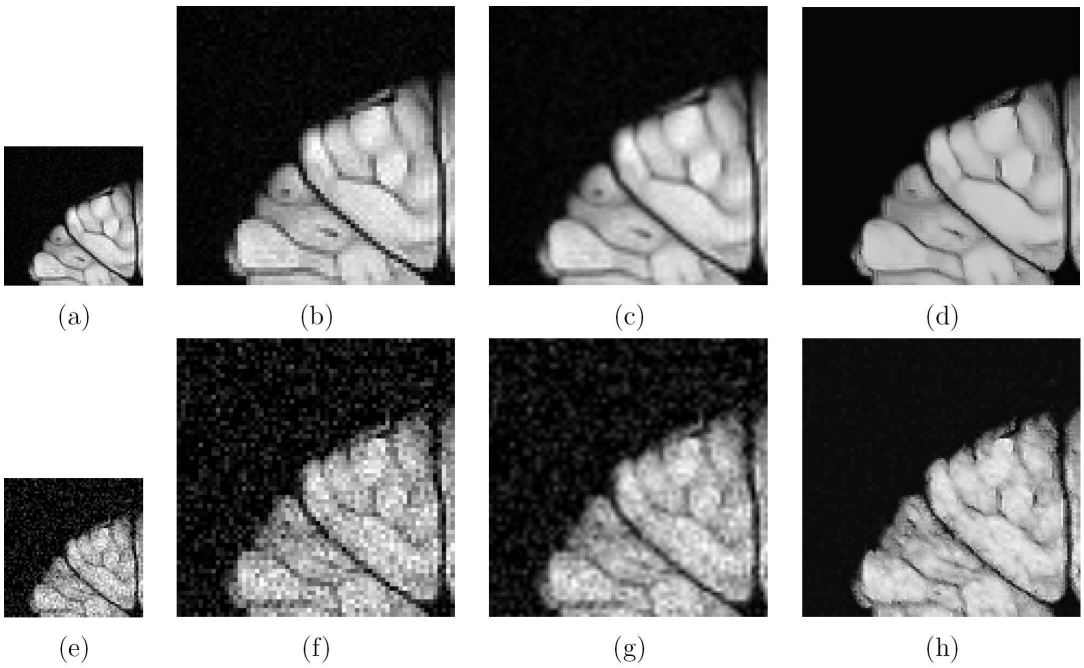


Figure 9.7: Respectively, (a,e) Original image. (b,f) Pixel replication. (c,g) Bilinear interpolation. (d,h) Self-examples. For both rows $h = 37.5$ is applied. In the second row the standard deviation of noise is $\sigma = 25$.

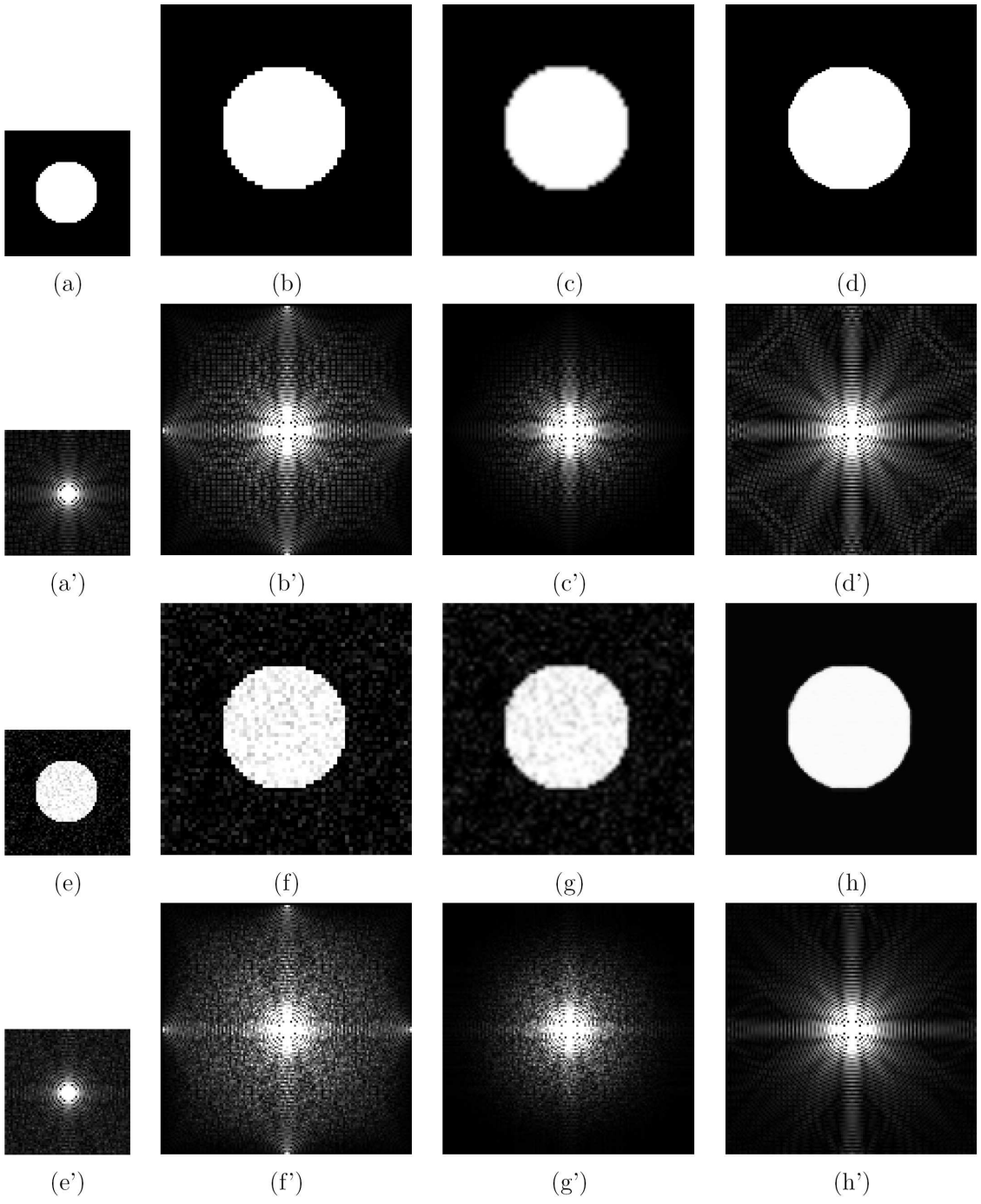


Figure 9.8: A comparison of the image zooming using self-examples, Algorithm 9.4.2, with other methods of zooming. Two input images are considered: Circular region (row 1), noisy circular region (row 3) with the noise of standard deviation $\sigma = 25$. The Fourier spectra of all images are also shown (rows 2 and 4). Starting at left: input image, zooming with pixel replication, zooming with bilinear interpolation, zooming with Algorithm 9.4.2. In rows 1 and 3 the value of h in the experiment are, respectively 2.5, and 125.

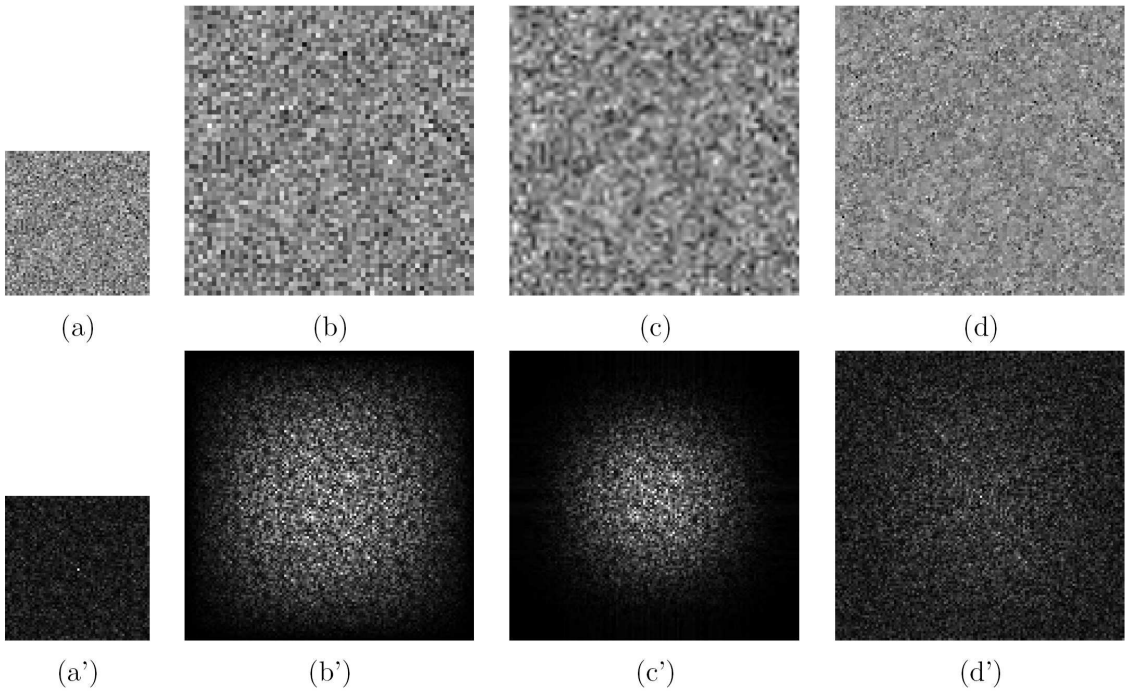
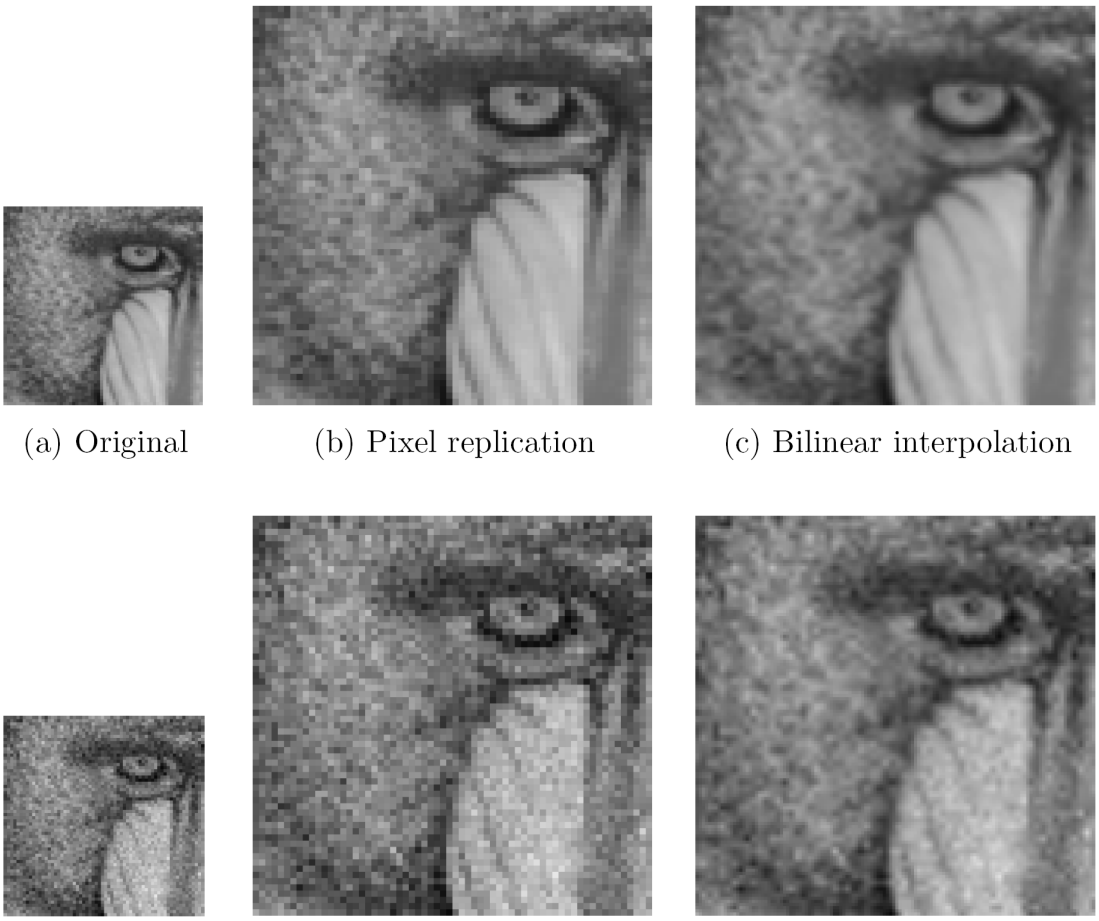


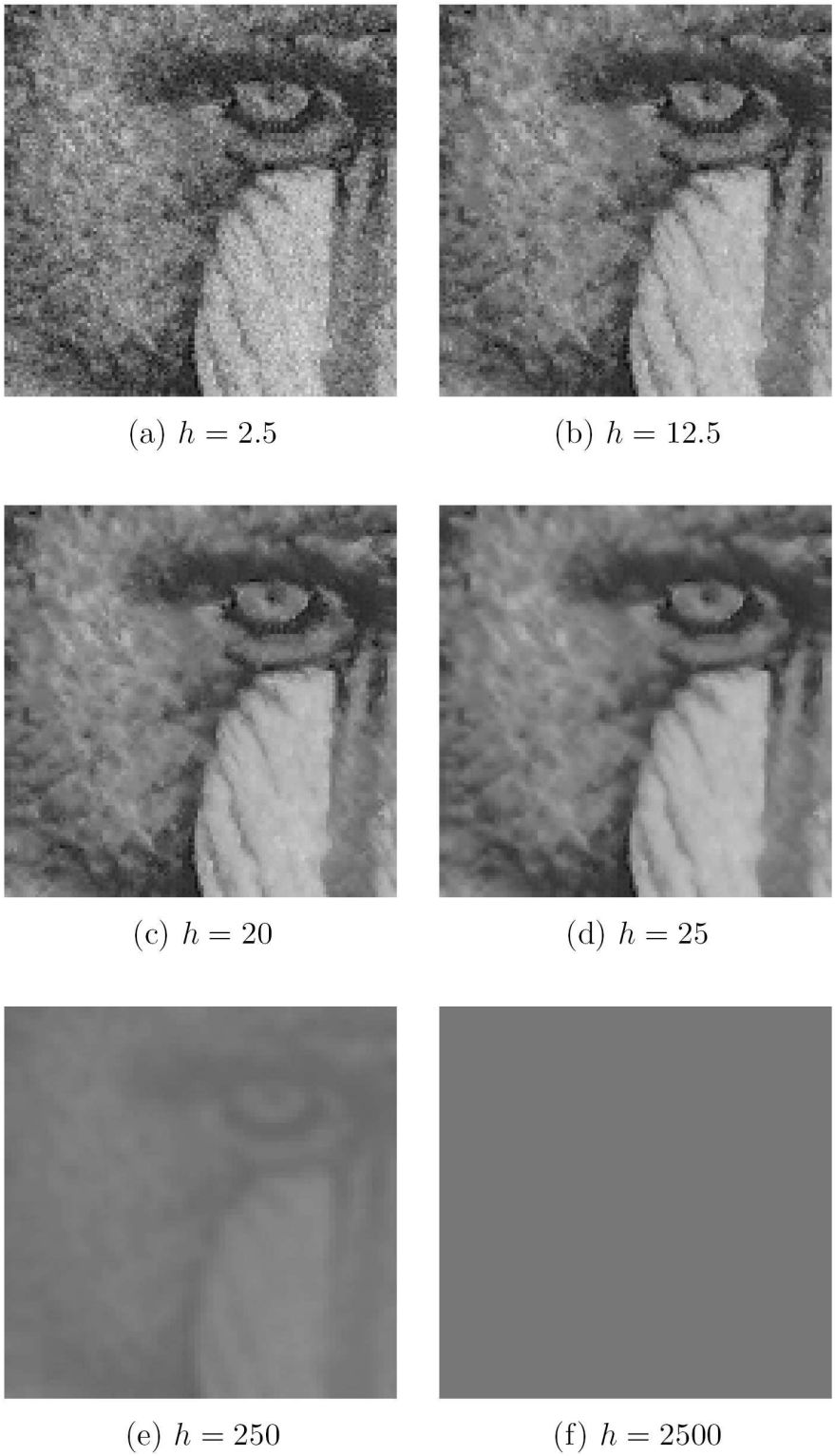
Figure 9.9: A comparison of the image zooming using self-examples, Algorithm 9.4.2, with other methods of zooming with pure Gaussian noise input image with $\sigma = 25$. The Fourier spectra of all images are also shown. Starting at left: input image, zooming with pixel replication, zooming with bilinear interpolation, zooming with Algorithm 9.4.2. The value of h in the experiment is 15.



(a) Original (b) Pixel replication (c) Bilinear interpolation

(d) Noisy input (e) Pixel replication (f) Bilinear interpolation

Figure 9.10: The data used in Figure 9.11.



(a) $h = 2.5$ (b) $h = 12.5$

(c) $h = 20$ (d) $h = 25$

(e) $h = 250$ (f) $h = 2500$

Figure 9.11: A comparison of the results obtained by image zooming using self-examples for different values of the filter parameter h. A noisy input image is considered for which the standard deviation of noise is $\sigma = 12.5$.

In rows 1 and 3 of Figure 9.8, we present the result of Algorithm 9.4.2 when applied to a simple circular shape in both noiseless and noisy situations, respectively. Below each of these rows are displayed the Fourier spectra of each image. In Figure 9.9, we show the effect of Algorithm 9.4.2 on a Gaussian white noise sample, along with corresponding Fourier spectra. The value of parameter h was set in a way that the variance of the noise was three-quarters of the variance of the input noise. However, the output is still noise-like, as opposed to the results of bilinear and pixel replication. It can be seen that the Fourier spectrum of the output is also distributed rather uniformly over the frequency domain, as opposed to the pixel replication and bilinear interpolation cases.

In Figure 9.11, we present the output of Algorithm 9.4.2 on a noisy input image (shown in Figure 9.10) choosing different values of the filter parameter h. The original image in this figure is given only for comparison. The input image in the second row is the algorithm's input. It can be observed that choosing small values of h will produce noisy outputs and as h grows the output will be smoothed out converging to a constant value image.

9.6 Concluding Remarks

In this chapter, we have presented a novel single-frame image zooming method of self-examples, explaining how it combines the ideas of fractal-based image zooming, example-based zooming and nonlocal-means image denoising. Our framework implicitly defines a regularization scheme which exploits the examples taken from the image itself at a different scale in order to achieve image zooming. The method essentially extends the NL-means image denoising technique to the image zooming problem. Various computational issues and results were also presented, showing that frequency domain extrapolation is in fact possible with this method.

Chapter 10

Multi-frame Super-resolution without Explicit Motion Estimation

10.1 Introduction

Naturally, there is always a demand for higher quality and higher resolution images. The level of image detail is crucial for the performance of many computer vision algorithms.

Current imaging devices typically consist of arrays of light detectors. A detector determines pixel intensity values depending upon the amount of light detected from its assigned area in the scene. The spatial resolution of images produced is proportional to the density of the detector array: the greater the number of pixels in the image, the higher the spatial resolution [Ngu00]. In many applications, however, the imaging sensors have poor resolution output. When resolution can not be improved by replacing sensors, either because of cost or hardware physical limits, one can resort to resolution enhancement algorithms. Even when superior equipment is available, such algorithms provide an inexpensive alternative.

The process of producing a high-resolution (HR) image given a single low-resolution (LR) image, known as *single-frame image zooming*, was addressed in the previous chapter. Another possibility is to take advantage of information from several observations rather than from a single image. The problem of recovering a high quality HR image from a set of distorted (e.g., warped, blurred, noisy) and LR images is known as *super-resolution* [EF97, HBA97, IP93, SS96, FREM04a, Bor04, Ngu00]. Fusion of the information from the observations is a fundamental challenge in the recovery process.

Super-resolution can not perform miracles. Extracting sub-pixel information from a single-frame or a sequence of identical frames is possible only if we have additional information about the scene. Hence, in order to obtain super-resolution, there must be available nonredundant information among the frames. Each LR frame should provide a different look at the same scene. With just one imaging device and under the same lighting conditions, we require some relative motions from frame to frame. Motion and nonredundant information are what make super-resolution possible [Ngu00].

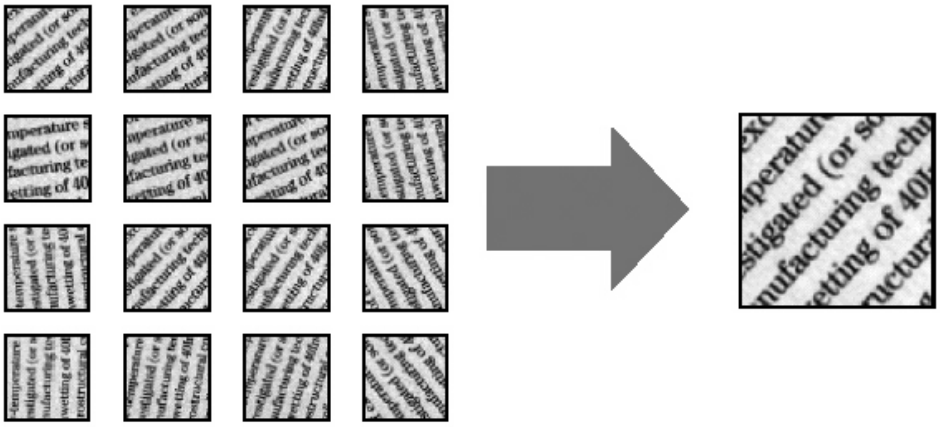


Figure 10.1: Super-resolution example: Fusion of measurements into a high-resolution image [Ela].

10.2 The Inverse Problem of Multi-frame Super-resolution

Formally, we review a general model for super-resolution widely used in the literature [Ngu00, FREM04a, FREM04b, Cha01, EF97]. Assume that the LR grid is

$$\Omega = [1, \ldots, M] \times [1, \ldots, N], \tag{10.1}$$

and given the positive integer z the HR grid is defined as

$$\Psi = [1, \ldots, Mz] \times [1, \ldots, Nz]. \tag{10.2}$$

Forward Model. We consider a forward degradation model that converts an ideal HR image $\mathbf{x}$ to degraded LR frames $\mathbf{u}_i$,

$$\mathbf{u}_i = \mathcal{H}_i \mathbf{x} + \mathbf{n}_i, \qquad 1 \leq i \leq k. \tag{10.3}$$

Here, the operator $\mathcal{H}_i = \mathcal{S}_z \mathcal{B} \mathcal{W}_i$ is the composition of a warping operator $\mathcal{W}_i : \ell^2(\Psi) \to \ell^2(\Psi)$ which maps the HR grid coordinate to the LR grid, a blurring operator $\mathcal{B} : \ell^2(\Psi) \to \ell^2(\Psi)$, and a downsampling operator $\mathcal{S}_z : \ell^2(\Psi) \to \ell^2(\Omega)$ of factor z in each direction. Also, $\mathbf{n}_i \in \ell^2(\Omega)$ denotes additive white independent Gaussian noise with zero-mean and variance σ^2. The inverse problem of multi-frame super-resolution can be stated as follows,

Inverse Problem: Given a set of k LR observed frames $\{\mathbf{u}_i\}_{i=1,\ldots,k} \in \ell^2(\Omega)$ of size $M \times N$, reconstruct the HR image $\mathbf{x} \in \ell^2(\Psi)$ of size $Mz \times Nz$. Figure 10.2 illustrates the problem setup.

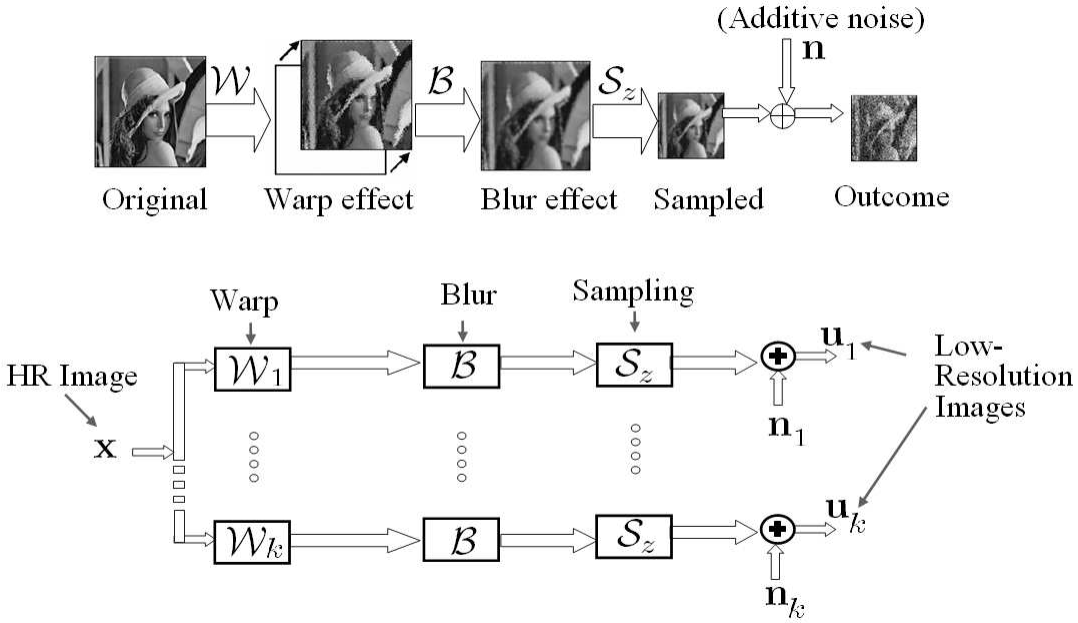


Figure 10.2: Forward Model of super-resolution.

Ignoring the blur, Figure 10.3 illustrates an example of the warping followed by down-sampling. The figure shows three 4×4 pixel LR frames on an 8×8 HR grid. Each set of symbols (square,

circle, triangle) indicates the sampling points of a frame with respect to the HR grid. We pick an arbitrary frame as a reference frame; in this case, the frame marked by the circular symbols. The sampling grid for the triangular frame is just a simple translation of the reference frame grid. The motion between the sampling grid for the square frame and the reference frame grid include translational, rotational, and magnification (zoom) components. The goal of super-resolution is to restore values at the HR grid points [Ngu00].

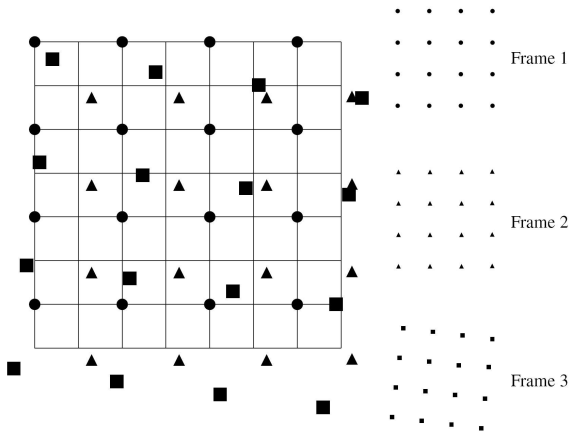


Figure 10.3: Illustration of warping and down-sampling. Low-resolution data on a high-resolution grid, taken from [Ngu00].

10.3 Short Survey of Some Super-resolution Techniques

A rather recent and comprehensive survey of super-resolution techniques is given in [FREM04a].

Irani and Peleg [IP93] proposed an iterative back-projection method to address the super-resolution problem. A large class of iterative multi-frame super-resolution methods have this simple simulate-and-correct approach to restoration.

Sauer and Allebach [SA87] modelled super-resolution as an interpolation problem with nonuniformly sampled data and used a projection onto convex sets (POCS) algorithm to reconstruct the image. Tekalp, Ozkan and Sezan [TOS92] also proposed a POCS-based approach to super-resolution. These approaches rely on the previous POCS restoration models of Stark and Oskoui [SO89]. In the context of super-resolution, given a low-resolution frame $\mathbf{u}_i$, a set-based constraint can be constructed using

$$\Psi_i = \{\mathbf{x} : \|\mathbf{u}_i - \mathcal{H}_i\mathbf{x}\| \leq \delta^i\}. \tag{10.4}$$

In forming this constraint, the parameter δ^i may be characterized by the statistics of the noise $\mathbf{n}_i$, which is present in the measured frame $\mathbf{u}_i$.

Ur and Gross [UG92] considered Papoulis' generalized multichannel sampling theorem [Pap97] for interpolating values on a higher resolution grid. Shekarforoush and Chellappa [SC99] extended

Papoulis' theorem for merging the nonuniform samples of multiple channels into HR data. Aizawa et al. [AKS91] also modelled super-resolution as an interpolation problem with nonuniform sampling and used a formula related to Shannon's sampling theorem [Sha49, Sha98, Whi29] to estimate values on a HR grid.

Tsai and Huang [TH84] were one of the first to superresolve a single HR image from several sampled LR frames (without blur). They considered interpolation from k LR frames $\{\mathbf{u}_i\}_{i=1,\ldots,k}$, each shifted from a reference frame by some shift δ_i. Frame $\mathbf{u}_i$ can be considered as samples from a continuous signal $\mathbf{x}(x+\delta_i)$, where $\mathbf{x}(x)$ is the ideal continuous image. In the frequency domain, by combining the Fourier shift theorem for the continuous Fourier transforms and the relationship between the continuous and discrete Fourier transform, they were able to calculate higher values of the frequency spectrum resulting in an improvement of resolution.

Other super-resolution models have been presented in the literature. A wavelet-based interpolation restoration method for super-resolution was proposed by Nguyen and Milanfar [Ngu00]. Hardie et al. [HBA97] proposed a joint MAP registration and restoration algorithm using a *Gibbs image prior*. Schultz and Stevenson [SS96] used a Markov random field model with Gibbs prior to better represent image discontinuities, such as transitions across sharp edges.

More recently Farsiu et al. [FREM04b] proposed an alternative data fidelity, or regularization term based on the ℓ^1 norm, which has been shown to be very robust to data outliers. They proposed a novel regularization term called Bilateral-TV, which provides robust performance while preserving the edge content common to real image sequences.

10.4 Super-resolution with no Explicit Motion Estimation

10.4.1 A Word on Motion Estimation

Accurate motion estimation has been a very important aspect of super-resolution schemes. In the forward process, described in Equation 10.3, the motion parameters are represented by the warping operators $\mathcal{W}_i$. In many existing super-resolution approaches, the motion is computed directly from the LR frames, while many other super-resolution algorithms unrealistically assume that motion parameters are precisely known. In general, however, accurate motion estimation of subpixel accuracy remains a fundamental challenge in super-resolution reconstruction algorithms.

In our opinion, however, it seems reasonable to assume that the motion can be relaxed from a strict grid mapping to a multi-pixel-pair intensity relation. In this view, pixel-pairs in different frames may be relevant to each other with some measured probability of confidence. In the method we propose below, instead of estimating the motion vectors explicitly, a framework is provided in which such confidence measures are evaluated and employed in the HR image reconstruction.

10.4.2 Changing the Order of Blur and Warp to Isolate Blur

It is well known that the order of $\mathcal{B}$ and $\mathcal{W}_i$ may be changed in the case that blur is linear spatially invariant. Hence, the equations may be written as

$$\mathbf{u}_i = \mathcal{S}_z \mathcal{W}_i \mathcal{B} \mathbf{x} + \mathbf{n}_i, \qquad 1 \leq i \leq k. \tag{10.5}$$

If we define $\mathbf{y} = \mathcal{B}\mathbf{x}$, then the measurements become

$$\mathbf{u}_i = \mathcal{S}_z \mathcal{W}_i \mathbf{y} + \mathbf{n}_i, \qquad 1 \leq i \leq k. \tag{10.6}$$

In our algorithm, we simply focus on solving $\mathbf{y}$, noting that any existing deblurring algorithm can be applied to reconstruct $\mathbf{x}$ after we obtain a solution for $\mathbf{y}$. In order to reduce notation, we define $\mathbf{y}_i = \mathcal{W}_i \mathbf{y}$, so that the above equations become

$$\mathbf{u}_i = \mathcal{S}_z \mathbf{y}_i + \mathbf{n}_i, \qquad 1 \leq i \leq k. \tag{10.7}$$

Furthermore, in many multi-frame super-resolution algorithms it is customary to employ one of the LR frames as a reference frame. The algorithm proposed in the next section provides an estimation of $\mathbf{y}_i$ denoted by $SR(\mathbf{y}_i)$ for any $1 \leq i \leq k$.

10.4.3 HR Reconstruction of the i-th Frame Given the j-th Frame

In what follows, we let $\overbrace{\mathbf{u}_i}$ denote the interpolation $\mathbf{u}_i$ from Ω to Ψ yielded by some standard technique, e.g., bilinear interpolation. Therefore, $\overbrace{\ldots}$ is a mapping from $\ell^2(\Omega)$ to $\ell^2(\Psi)$. Note that, the interpolation $\overbrace{\mathbf{u}_i}$ provides an approximation of $\mathbf{y}_i$, i.e., $\mathbf{y}_i \approx \overbrace{\mathbf{u}_i}$ for $1 \leq i \leq k$. Here, however, we seek a superior approximation to $\mathbf{y}_i$, which will be denoted as $SR(\mathbf{y}_i)$.

The following scheme is inspired by the work on image and image-sequence denoising in [BCM05b, BCM05c, BCM05a]. For any $x \in \Psi$, we evaluate the conditional expectation of $SR(\mathbf{y}_i)(x)$ given the observed LR image $\mathbf{u}_j$ by

$$E[SR(\mathbf{y}_i)(x) \mid \mathbf{u}_j] = \frac{1}{W(x,i,j)} \sum_{y \in \Psi} w(x,y,i,j) \overbrace{\mathbf{u}_j}(y), \text{ such that} \tag{10.8}$$

$$w(x,y,i,j) = \left[\exp\Big(- \frac{\| \overbrace{\mathbf{u}_i}(\mathcal{N}^d\{x\}) - \overbrace{\mathbf{u}_j}(\mathcal{N}^d\{y\}) \|^2}{h^2} \Big) \right], \text{ and}$$

$$W(x,i,j) = \sum_{y \in \Psi} w(x,y,i,j), \tag{10.9}$$

where $\mathcal{N}^d\{x\}$ denotes a square neighbourhood of length $(2d+1) \times (2d+1)$ centred at x. The confidence measures mentioned earlier are expressed in terms of the $w(x,y,i,j)$'s, where W is a normalization factor. Note that in evaluation of $w(x,y,i,j)$ we have used the interpolated copies of $\mathbf{u}_i$ and $\mathbf{u}_j$, respectively denoted by $\overbrace{\mathbf{u}_i}$ and $\overbrace{\mathbf{u}_j}$. Employing such a notion automatically takes into account translations of $\mathbf{u}_i$ and $\mathbf{u}_j$ by sub-pixel accuracy. Also, h is a regularization parameter that can be adjusted to control the smoothness of the output.

10.4.4 HR Reconstruction of the i-th Frame Given the Whole Image Sequence

Eventually, we evaluate the conditional expectation of $SR(\mathbf{y}_i)(x)$ for any $x \in \Psi$, given the information of all of the frames $\{\mathbf{u}_j\}$ for $1 \leq j \leq k$:

$$E[SR(\mathbf{y}_i)(x) \mid \{\mathbf{u}_j\}_{1\leq j\leq k}] = \frac{1}{C(i)} \sum_{1\leq j\leq k} c(|i-j|) \times E[SR(\mathbf{y}_i)(x) \mid \mathbf{u}_j], \tag{10.10}$$

$$C(i) = \sum_{1\leq j\leq k} c(|i-j|), \tag{10.11}$$

where c is a decaying function of $|i-j|$, and C is a normalization factor. The expression $c(|i-j|)$ in this equation represents the temporal confidence on the expectations computed for each of the various frames, j, which has been taken into account in reconstructing the HR image $SR(\mathbf{y}_i)$. In the experiments reported below, we have assumed that each of the frames in hand is equally likely useful in producing the HR details of the i-th frame. Hence, we have taken c to be a box-function of large enough support which yields $c(|i-j|) = 1$. As a result,

$$E[SR(\mathbf{y}_i)(x) \mid \{\mathbf{u}_j\}_{1\leq j\leq k}] = \frac{1}{k} \sum_{1\leq j\leq k} E[SR(\mathbf{y}_i)(x) \mid \mathbf{u}_j]. \tag{10.12}$$

10.5 Computational Experiments

As in the case of the NL-means denoising algorithm, the algorithm described above is computationally intensive. The major computational burden exists in the complexity of computing the weights $w(x, y, i, j)$. A primary scheme to overcome this complexity, introduced in [BCM05b, BCM05c, BCM05a], is to restrict the search window by restricting $y \in \Psi \cap \mathcal{N}^r\{x\}$ i.e., y lies in a square neighbourhood of x with size $(2r+1) \times (2r+1)$, as opposed to the entire field of Ω.

Figures 10.4, 10.5, and 10.6 show the result of evaluating $E[SR(\mathbf{y}_i)(x) \mid \{\mathbf{u}_j\}_{1\leq j\leq k}]$, on an image sequence taken from the data-set library of MDSP at U. California Santa Cruz (`http://www.soe.ucsc.edu/ milanfar/software/sr-datasets.html`), originally obtained from the Adyoron Intelligent Systems Ltd., Tel Aviv, Israel. We have taken the first 20 frames, i.e., $k = 20$, of size 32×32 from this sequence, i.e., $M = N = 32$, and have added independent additive white Gaussian noise of standard deviation $\sigma = 12.5$ to the data set. Figure 10.4 shows the result of applying nearest neighbourhood interpolation on the second frame of this sequence, i.e., when $i = 2$ for a zooming factor of $z = 3$. In Figure 10.5, the result of bilinear interpolation on the same frame is shown. We have plotted the result of $E[SR(\mathbf{y}_2)(x) \mid \{\mathbf{u}_j\}_{1\leq j\leq 20}]$ in Figure 10.6. In this experiment, we have taken $d = 4$ (corresponding to a neighbourhood of size 9×9), a restricted search window of radius $r = 13$, a zooming factor of $z = 3$, and a smoothness parameter $h = 20$. We have used bilinear interpolation of $\mathbf{u}$ wherever the interpolated image notation $\widehat{\mathbf{u}}$ has appeared in the computations.

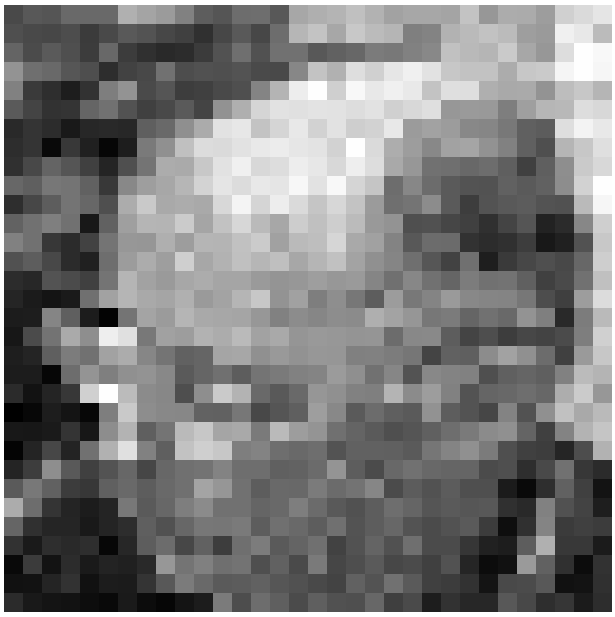


Figure 10.4: Nearest neighbour interpolation.

Figure 10.5: Bilinear interpolation.

Figure 10.6: Proposed multi-frame super-resolution with the following parameters: The original image sequence is of size 32×32, i.e., $M = N = 32$, of a $k = 20$ frames sequence. The HR counterpart of second frame, i.e., $i = 2$ is desired. Additive white Gaussian noise of $\sigma = 12.5$ is added. Neighbourhood of radius $d = 4$, search window radius $r = 13$, zooming parameter $z = 3$, and smoothness parameter $h = 20$.

10.6 Concluding Remarks

In this section, we have introduced a novel multi-frame super-resolution technique which does not require explicit motion estimation. Our algorithm was inspired by the non-local-means denoising algorithms introduced in [BCM05b, BCM05c, BCM05a]. The computational burden of the scheme is a formidable challenge, which precludes any iteration scheme to improve the results. Since there are many parameters in our algorithms, it seems that a fair comparison with other super-resolution algorithms cannot be made. As a result, no comparisons were presented here. That being said, experiments with a number of sets of parameters suggest that our algorithm yields results which are quite comparable if not superior to some of the algorithms in [FREM04a, FREM04b] especially when the image sequence is of very low signal-to-noise ratio.

Chapter 11

Conclusions, Contributions and Future Perspectives

11.1 Conclusions

In this thesis, we have examined the concept of image self-similarity and provided solutions to various associated inverse problems such as resolution enhancement and missing fractal code.

11.1.1 Introduction and Background Material

In the first chapter, we started with introductory material, notation, preparation, and organization of concepts. (Chapter 1)

It was first necessary to review some basic principles of inverse problems and their solutions, including the important concept of regularization. (Chapter 2)

We then investigated the concept of self-similarity of images. (Chapter 3)

11.1.2 Extending the Contractivity and Convergence Results of the Fractal Transform Operator

We revisited the concept of fractal-based methods and addressed various open theoretical problems in the area. This includes formulating a necessary and sufficient condition for the contractivity of the block fractal transform operator. Furthermore, novel analytical results on the convergence of the fractal decoding were derived. (Chapter 4)

11.1.3 Including *A priori* Information in the Fractal Code

Fractal imaging methods have relied almost exclusively on Banach's contraction mapping theorem at the decoding stage. This can be viewed as a major drawback for the traditional fractal-based methods in that the output is quite restricted – no prior knowledge or additional regularization can be combined with these methods. One may ask whether it is absolutely necessary to employ Banach's contraction mapping principle, i.e., to iterate the fractal transform T to reach its fixed point.

Generalized Fractal Image Coding as Projections onto Convex Sets

In this thesis, we examined various possibilities of including *a priori* information with the fractal code. It was shown that fractal image coding can be viewed and generalized in terms of the method of projections onto convex sets (POCS). In the approach, the fractal code is considered as a set of spatial domain similarity constraints. As a result, POCS provides an opportunity to apply the fractal code of an image along with additional constraints at the decoding stage. (Chapter 5)

Regularized Fractal Image Decoding

Furthermore, we presented an algebraic formulation, in which prior information may be incorporated in terms of penalty functions. To illustrate some of these ideas, we considered the problem of incomplete fractal code and showed how the theory presented can be applied. (Chapter 6)

11.1.4 Regularization Expressions Involving Self-similarity

We introduced and analyzed a set of regularization expressions based on self-similarity properties of images in order to address the classical inverse problem of image denoising and the ill-posed inverse problem of single-frame image zooming. (Chapter 7)

11.1.5 Same-scale vs. Cross-scale Approximations

We discussed that a fundamental difference between NL-means and fractal coding is same-scale vs. cross-scale similarity exploitation. The computational experiments performed in this thesis hint that an appropriately defined cross-scale version of the NL-means filter should behave in a similar fashion to the traditional same-scale NL-means filter. (Chapter 8)

11.1.6 Non-local Approaches to Image and Video Resolution Enhancement

We considered the process of producing a high-resolution (HR) image given a single low-resolution (LR) image known as *single-frame image zooming*. Furthermore, we considered the problem of recovering a high-resolution image from a set of distorted (e.g., warped, blurred, noisy) and lower resolution images is known as *super-resolution*.

Single-frame Image Zooming

We presented a single-frame image zooming technique based on so-called "self-examples". The method combined the ideas of fractal-based image zooming and nonlocal-means image denoising in a consistent framework. (Chapter 9)

Multi-frame Super-resolution

A novel super-resolution approach was introduced which was closely associated with the NL-means denoising. In the proposed technique, no explicit motion estimation was performed, unlike the case in many other methods. (Chapter 10)

11.1.7 Implementation of the Algorithms

The algorithms in this thesis were written in MATLAB Version 7.1.246 (R14) Service Pack 3, and were executed on an Intel Pentium 4 machine of 3 GHz CPU and 504 MB RAM under Microsoft Windows XP operating system. Table 11.1 indicates approximate execution times of the main algorithms proposed in this thesis corresponding to each resulting figure.

Figure(s)	**Execution Time (Minutes)**
5.5(c)-5.9(c)	10-15
6.1(b,c), 6.2(a,b,c)	2-5
6.4(a,b,c)	Less than 1
9.7(d,h), 9.8(d,h)	5-10
9.9(d), 9.11(a-f)	5-10
10.6	30-45

Table 11.1: Execution times of the algorithms

11.2 Contributions

In the preparation process of this thesis, the following manuscripts have been published or are in press. For each contribution, the chapters of this thesis that share the same material are also mentioned.

- [EV06a] M. Ebrahimi and E. R. Vrscay, Fractal Image coding as Projections onto Convex Sets. In Image Analysis and Recognition, volume 4141, pages 493-506, Berlin/Heidelberg, Springer, 2006. (Chapter 5)
- [EV06b] M. Ebrahimi and E. R. Vrscay, Regularized Fractal Image Decoding. In Proceedings of CCECE 06, pages 1933-1938, Ottawa, Canada, May 2006. (Chapter 6)
- [EV07a] M. Ebrahimi and E. R. Vrscay, Regularization Schemes Involving Self-similarity in Imaging Inverse Problems. In Proceedings of Applied Inverse Problems (AIP) 2007, University of British Columbia, Vancouver, Canada, June 2007. (Chapter 7)
- [EV07b] M. Ebrahimi and E. R. Vrscay, Solving the Inverse Problem of Image Zooming Using "Self-examples. In Image Analysis and Recognition, volume 4633, pages 117-130, Berlin/Heidelberg, Springer, 2007. (Chapter 9)
- [EV08a] M. Ebrahimi and E. R. Vrscay, Examining the Role of Scale in the Context of the Non-Local-Means Filter. In Proceedings of The International Conference on Image Analysis and Recognition, volume 5112, pages 170-181 Povoa de Varzim, Portugal, June 2008. (Chapter 8)

- [EV08b] M. Ebrahimi and E. R. Vrscay, Multi-frame Super-resolution with no Explicit Motion Estimation. In Proceedings of The International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV 2008, volume 2, pages 455-459, Las Vegas, Nevada, USA, July 2008. (Chapter 10)

- [EV08c] M. Ebrahimi and E. R. Vrscay, Self-similarity in Imaging, 20 Years After "Fractals Everywhere", In the Proceedings of the International Workshop on Local and Non-Local Approximation in Image Processing (LNLA), Lausanne, Switzerland, 2008. (Chapter 3)

- [OEW08] J. Orchard, M. Ebrahimi and A. Wong, Efficient Non-Local-Means Denoising using the SVD, In the Proceedings of ICIP 2008.

The following talks have been delivered at various scientific events. Presented talks with published proceedings are not listed below to avoid duplications with the above list.

- M. Ebrahimi and E. R. Vrscay, Nonlocal Approaches to Image and Video Resolution Enhancement, Presented at SIAM Conference on Imaging Science, San Diego, California, USA, July 7-9, 2008.

- M. Ebrahimi and E. R. Vrscay, Nonlocal-means Single-frame Image Zooming, Contributed talk presented at the 6th International Congress on Industrial and Applied Mathematics (ICIAM), Zurich, Switzerland, July 16-20, 2007.

- M. Ebrahimi and E. R. Vrscay, Image Super-resolution using Self-examples, Contributed poster presented at BICV 2007.

- M. Ebrahimi and E. R. Vrscay, Generalized Fractal Image Coding using Projections onto Convex Sets, Contributed talk presented at SIAM Conference on Imaging Science, Radisson University Hotel, Minneapolis, Minnesota, USA, May 15-17, 2006.

The following manuscripts have been submitted for publication or are in progress.

- D. La Torre, and E. R. Vrscay, M. Ebrahimi and M. Barnsley, A Complete Metric Space of Measure-valued Images and Associated Fractal Transforms, Submitted to the SIAM Journal on Imaging Sciences, Submission year: 2007, In revision.

- M. Ebrahimi and E. R. Vrscay, A Necessary and Sufficient Contractivity Condition for the Fractal Transform Operator, In progress.

- M. Ebrahimi and E. R. Vrscay, Addressing the Problem of Incomplete Fractal Codes, In progress.

11.3 Future Perspectives

11.3.1 Extending the Contractivity and Convergence Results of the Fractal Transform Operator

Following the generalized collage theorem presented in the thesis, it may seem useful to find bounds on the corresponding expression explicitly in terms of the coefficient of the affine transforms in the fractal code. The theory presented for the necessary and sufficient condition gives such an explicit bound in the case that T is a contraction. Furthermore, although theoretical contractivity conditions were developed for codes involving same-size domain-range blocks, no experiments were performed to evaluate the performance of such codes.

The interconnection and mixing of domain-range assignments is another important aspect of fractal coding. This idea was implicitly in mind when we developed various theoretical results in fractal coding. The author believes that employing other existing tools, e.g., graph theory techniques, may lead to more efficient coding schemes in this context.

11.3.2 Generalized Fractal Image Coding as Projections onto Convex Sets

A number of new questions arise following the introduction of POCS-based self-similarity constraints. The constraint sets and the associated projections in this thesis were computed based on *point-wise* collage distances. It seems natural to extend the theory to handle domain-range *blocks* instead of *points*. This may significantly improve the speed of the corresponding algorithms.

Also, the issue of inconsistency of constraints was not addressed although theoretical tools for this extension are readily available. Furthermore, the theory may be easily applied to the multiparent fractal coding schemes.

11.3.3 Regularized Fractal Image Decoding

The material on regularized fractal image decoding was viewed as an algebraic extension of the POCS self-similarity formulation. Two important questions to ask here are the following. 1-How to design fractal codes which are least sensitive to the loss of its components? 2-What regularizers are most suitable to be used along with self-similarity constraints?

11.3.4 Regularization Expressions Involving Self-similarity

The author has made primitive efforts to define regularization expressions based on self-similarity properties of images. In practice, the readily available models were employed including the fractal-transform and the NL-means denoising operator, in a MAP-based formulation. The author believes that the pursuit of more efficient priors (regularization expressions) based on self-similarity properties of images is a quite challenging yet extremely interesting and worthwhile goal.

11.3.5 Same-scale vs. Cross-scale Approximations

More rigorous mathematical or statistical analysis may be required to analyze the performance of same-scale and cross-scale approximations. In addition, the effect of using affine transformed gray-scale values in various non-local approaches remains untouched.

11.3.6 Single-frame and Multi-frame Resolution Enhancement Techniques

There are a number of possible extensions of the resolution enhancement algorithms proposed in this thesis. This includes extending the algorithms to handle non-integer zooming factors and experimenting in the space of color images.

It turns out that the multi-frame algorithm proposed is rather sensitive to the initial interpolation used, which may be enhanced in a corresponding improved model. Another challenge is the issue of examining different motion models in the reconstruction process. Automatic parameter selection is another challenge of the proposed algorithms. As was pointed out earlier, many zooming algorithms proposed are computationally expensive. Developing computational approaches to speed-up various non-local methods will be another challenge ahead of the imaging community in the following years.

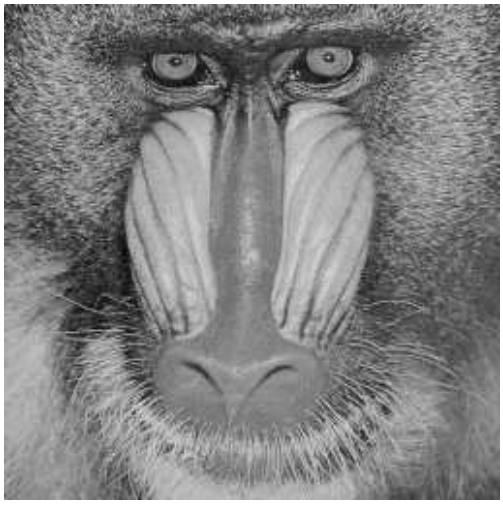
11.3.7 The Final Word

This thesis has been intended as a scholarly exercise for the author to become familiar with some of the existing inverse theory techniques in imaging. The author plans to extend his knowledge and contributions in the field and apply his expertise to the existing real-life problems in engineering and medicine in the future.

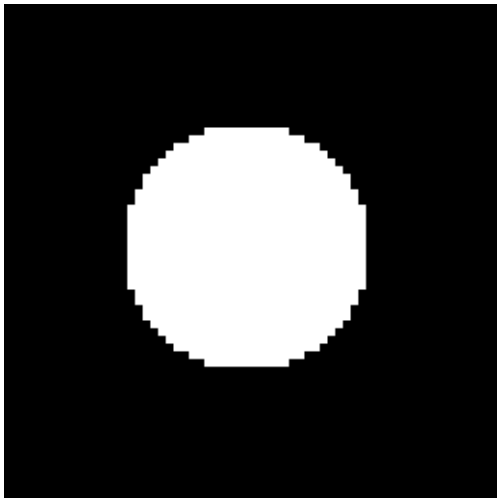
Test Images

(a)

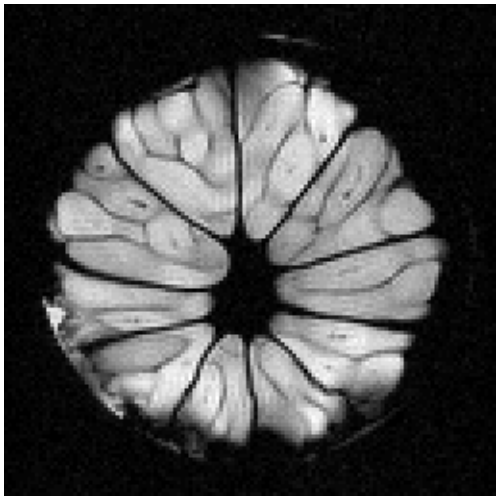
(b)



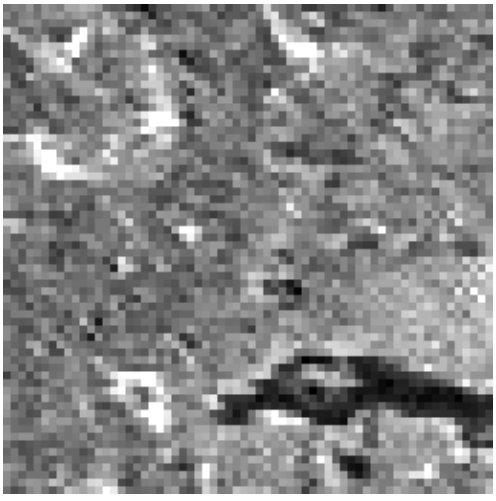
(c)



(d)



(e)



(f)

Description of Test Images

- **(a) Lena**
- **(b) Cameraman**
- **(c) Mandrill**
- **(d) Low Resolution Circle**
- **(e) Lime MRI**, Cross-sectional magnetic resonance image of lime fruit
  Courtesy of Drs. C. Lemaire and H. Peemoeller, Dept. of Physics, University of Waterloo
- **(f) Porous Medium**, Scanning Electron Microscope (SEM) image of a porous medium
  Courtesy of Drs. Carolyn Hansson and Shahzma Jaffer, Dept. of Mechanical Engineering, University of Waterloo

Abbreviations

CCD	Charged-Coupled Device	**MMSE**	Minimum Mean-Squared Error
CG	Conjugate Gradient	**MNLS**	Minimum Norm Least Square
DCT	Discrete Cosine Transform	**MSE**	Mean Squared Error
DFT	Discrete Fourier Transform	**NLmeans**	NonLocal-means
DWT	Discrete Wavelet Transform	**PDE**	Partial Differential Equations
FFT	Fast Fourier Transform	**PDF**	Probability Density Function
HR	High-Resolution	**POCS**	Projection Onto Convex Sets
IFS	Iterated Function Systems	**PSNR**	Peak Signal-to-Noise Ratio
IFSP	IFS with Probabilities	**RFDA**	Regularized Fractal Decoding Algorithm
IFSW	IFS on Wavelet coefficient trees	**RMSE**	Root Mean Squared Error
KKT	Karush-Kuhn-Tucker	**SD**	Steepest Descent
LR	Low-Resolution	**SEM**	Scanning Electron Microscope
LS	Least Squares	**SVD**	Singular Value Decomposition
MAP	Maximum *A posteriori*	**TV**	Total Variation
MLE	Maximum Likelihood Estimator	**VQ**	Vector Quantization

References

[AH77] H. F. Andrews and B. R. Hunt. *Digital Image Restoration*. Prentice-Hall, NJ, 1977. 3, 20, 22

[AKS91] K. Aizawa, T. Komatsu, and T. Saito. Acquisition of very high-resolution images using stereo cameras. In *Proceedings of SPIE Visual Communications and Image Processing*, volume 1605, pages 318–328, Boston, MA, November 1991. 139

[Ale05] S. K. Alexander. *Multiscale Methods in Image Modelling and Image Processing*. PhD thesis, Department of Applied Mathematics, University of Waterloo, 2005. 39, 41, 44, 46, 93, 104, 106

[AVT08] S. K. Alexander, E. R. Vrscay, and S. Tsurumi. A simple, general model for affine image self-similarity. In *Image Analysis and Recognition, Lecture Notes in Computer, Proceedings of ICIAR 08*, volume 5112, pages 192–203, Berlin-Heidelberg, 2008. Springer. 39, 41, 104, 106

[Bar88] M. F. Barnsley. *Fractals Everywhere*. Academic Press, New York, 1988. 3, 29, 32, 38, 104

[Bay64] T. Bayes. An essay toward solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370–418, 1764. 20

[BCM05a] A. Buades, B. Coll, and J. M. Morel. Denoising image sequences does not require motion estimation. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 70–74, 2005. 3, 42, 140, 141, 145

[BCM05b] A. Buades, B. Coll, and J.M. Morel. A nonlocal algorithm for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 60–65, San-Diego, California, June 2005. 3, 23, 29, 42, 43, 44, 93, 99, 104, 124, 140, 141, 145

[BCM05c] A. Buades, B. Coll, and J.M. Morel. A review of image denoising algorithms, with a new one. *SIAM Journal on Multiscale Modeling and Simulation (MMS)*, 4(2):490–530, 2005. 3, 23, 29, 42, 43, 44, 93, 99, 104, 124, 140, 141, 145

[BD85] M. F. Barnsley and S. Demko. Iterated function systems and the global construction of fractals. In *Proceedings of the Royal Society of London*, number A399, pages 243–275, 1985. 3, 30

[BEHL85] M. F. Barnsley, V. Ervin, D. Hardin, and J. Lancaster. Solution of an inverse problem for fractals and other sets. In *The Proceedings of the National Academy of Sciences*, volume 83, pages 1975–1977, USA, 1985. 30, 31, 34, 35, 46

[BH93] M. F. Barnsley and L. P. Hurd. *Fractal Image Compression.* Wellesley, Massachusetts, 1993. 32

[BK02] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183, 2002. 23

[Bon97] D. J. Bone. Orthonormal fractal image encoding using overlapping blocks. *Fractals*, 5 (Supplementary Issue):187–199, April 1997. 39

[Bor04] S. Borman. *Topics in Multiframe Superresolution Restoration.* PhD thesis, Graduate Program in Electrical Engineering, University of Notre Dame, Indiana, April 2004. 136

[Bra86] R. Bracewell. *The Fourier Transform and its Applications.* McGraw Hill, New York, 1986. 3

[BS88] M. Barnsley and A. Sloan. A better way to compress images. *BYTE Magazine*, pages 215–223, January 1988. 31

[CCH99] W. K. Carey, D. B. Chuang, and S. S. Hemami. Regularity-preserving image interpolation. *IEEE Transactions on Image Processing*, 8(9), September 1999. 40

[CDS98] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998. 23

[Cha01] S. Chaudhuri. *Super-Resolution Imaging.* Kluwer, Boston, MA, 2001. 39, 120, 121, 123, 137

[Com94] P. L. Combettes. Inconsistent signal feasibility problems: Least-squares solutions in a product space. *IEEE Transactions on Signal Processing*, 42(11):2955–2966, November 1994. 77

[CPT04] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, 2004. 23, 124

[CR07] E. Candes and J. Romberg. Sparsity and incoherence in compressive sampling. *Inverse Problems*, 23(3):969–985, 2007. 23

[CW05] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM Journal on Multiscale Modeling and Simulation*, 4(4):1168–1200, November 2005. 25

[DE06] D. Datsenko and M. Elad. Example-based single document image superresolution: A global MAP approach with outlier rejection. *Journal of Mathematical Signal Processing*, 2006. 22, 23

[DJ94] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994. 23

[ED07] M. Elad and D. Datsenko. Example-based regularization deployed to super-resolution reconstruction of a single image. *The Computer Journal*, 50(4):1–16, 2007. 21, 22, 23, 121, 123

[EF97] M. Elad and A. Feuer. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. *IEEE Transactions on Image Processing*, 6(12):1646–1658, 1997. 121, 136, 137

[EL99] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision*, pages 1033–1038, Corfu, Greece, September 1999. 23, 124

[Ela] M. Elad. Archive of presentations. http://www.cs.technion.ac.il/ elad/. xvi, 136

[EV06a] M. Ebrahimi and E. R. Vrscay. Fractal image coding as projections onto convex sets. In *Image Analysis and Recognition*, volume 4141, pages 493–506, Berlin/Heidelberg, 2006. Springer. 3, 149

[EV06b] M. Ebrahimi and E. R. Vrscay. Regularized fractal image decoding. In *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, pages 1933–1938, Ottawa, Canada, May 2006. 3, 149

[EV07a] M. Ebrahimi and E. R. Vrscay. Regularization schemes involving self-similarity in imaging inverse problems. In *Proceedings of Applied Inverse Problems (AIP) 2007*, University of British Columbia, Vancouver, Canada, June 2007. 3, 44, 149

[EV07b] M. Ebrahimi and E. R. Vrscay. Solving the inverse problem of image zooming using "self-examples". In *Image Analysis and Recognition*, volume 4633, pages 117–130, Berlin/Heidelberg, 2007. Springer. 149

[EV08a] M. Ebrahimi and E. R. Vrscay. Examining the role of scale in the context of the non-local-means filter. In *Proceedings of The International Conference on Image Analysis and Recognition*, volume 5112, pages 170–181, Povoa de Varzim, Portugal, June 2008. 3, 41, 42, 149

[EV08b] M. Ebrahimi and E. R. Vrscay. Multi-frame super-resolution with no explicit motion estimation. In *Proceedings of The International Conference on Image Processing,*

Computer Vision, and Pattern Recognition, IPCV, volume 2, pages 455–459, Las Vegas, Nevada, USA, July 2008. 3, 150

[EV08c] M. Ebrahimi and E. R. Vrscay. Self-similarity in imaging, 20 years after "Fractals Everywhere". In *International Workshop on Local and Non-Local Approximation in Image Processing, LNLA2008*, Lausanne, Switzerland, 2008. 150

[Fis95] Y. Fisher. *Fractal Image Compression, Theory and Application*. Springer-Verlag, New York, 1995. 3, 32, 39, 40, 46, 104

[Fis98] Y. Fisher, editor. *Fractal Image Encoding and Analysis*. NATO ASI Series F 159. Springer-Verlag, New York, 1998. 32, 39, 40, 46

[FJP02] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002. 23

[FPC00] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47, 2000. 23

[FREM04a] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Advances and challenges in super-resolution. *International Journal of Imaging Systems and Technology*, 14(2):47–57, August 2004. 136, 137, 138, 145

[FREM04b] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Fast and robust multi-frame super-resolution. *IEEE Transactions on Image Processing*, 13(10):1327–1344, October 2004. 137, 139, 145

[FV98] B. Forte and E. R. Vrscay. *Theory of Generalized Fractal Transforms, in Fractal Image Encoding and Analysis*. Springer Verlag, New York, NATO ASI Series F 159 edition, 1998. 46, 47, 49

[GAaYTH97] M. Gharavi-Al., R. DeNardo amd Y. Tenda, and T. S. Huang. Resolution enhancement of images using fractal coding. In *Proceedings of SPIE Visual Communications and Image Processing*, volume 3024, pages 1089–1100, San Jose, CA, 1997. 40

[Ger74] R. Gerchberg. Super-resolution through error energy reduction. *Optical Acta*, (21):709–720, 1974. 123

[GFV03] M. Ghazel, G. Freeman, and E. R. Vrscay. Fractal image denoising. *IEEE Transactions on Image Processing*, 12(12):1560–1578, December 2003. 32, 39, 44, 75, 93, 104

[GFV06] M. Ghazel, G. Freeman, and E. R. Vrscay. Fractal-wavelet image denoising revisited. *IEEE Transactions on Image Processing*, 15:2669–2675, 2006. 39, 44, 93

[Gha99] M. Ghazel. Adaptive Fractal Image Compression in the Spatial and Wavelet Domains. Master's thesis, University of Waterloo, 1999. 46

[Gha04] M. Ghazel. *Adaptive Fractal and Wavelet Image Denoising*. PhD thesis, University of Waterloo, 2004. 33, 39, 46

[Ham98] R. Hamzaoui. *Encoding and Decoding Complexity Reduction and VQ Aspects of Fractal Image Compression*. Taschenbuch Shaker Verlag, Germany, 1998. 36, 56, 79

[HBA97] R. Hardie, K. Barnard, and E. Armstrong. Joint MAP registration and high-resolution image estimation using a sequence of undersampled images. *IEEE Transactions on Image Processing*, 6(12):1621–1633, 1997. 136, 139

[HC97] H. Ho and W. Cham. Attractor image coding using lapped partitioned iterated function systems. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, page 2917, 1997. 39

[Hol96] G. Holst. *CCD Arrays, Cameras, and Displays*. JCD Publishing, Winter Park, FL, 1996. 120

[HT03] E. Haber and L. Tenorio. Learning regularization functionals. *Inverse Problems*, 19:611–626, 2003. 23

[Hut81] J. Hutchinson. Fractals and self-similarity. *Indiana University Mathematics Journal*, 30:713–747, 1981. 30

[IP93] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graphical Model and Image Processing*, 53:324–335, 1993. 136, 138

[Jac89] A. Jacquin. *A Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding*. PhD thesis, Georgia Institute of Technology, 1989. 31

[Jac92] A. Jacquin. Image coding based on a theory of iterated contractive image transformations. In *IEEE Transactions on Image Processing*, volume 1, pages 18–30, 1992. 31, 38

[Jai01] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 2001. 3, 20, 123

[Kar39] W. Karush. Minima of Functions of Several Variables with Inequalities as Side Constraints. Master's thesis, Department of Mathematics, University of Chicago, Chicago, Illinois, 1939. 65

[KT51] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of 2nd Berkeley Symposium*, pages 481–492. Berkeley: University of California Press, 1951. 65

[LR99] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modelling*. ASA and SIAM, Philadelphia, 1999. 3

[Lu97] N. Lu. *Fractal Imaging*. Academic Press, New York, 1997. 3, 32, 38, 39, 40, 104

[Mal98] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic, San Diego, CA, 1998. 3, 23, 40, 104

[Man83] B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman, New York, 1983. 29

[MV07] G.S. Mayer and E. R. Vrscay. Iterated Fourier transform systems: A method for frequency extrapolation. In *Image Analysis and Recognition, Lecture Notes in Computer Science 4633, Proceedings of ICIAR 07*, pages 728–739, Berlin-Heidelberg, 2007. Springer. 40

[MZ92] S. Mallat and S. Zhong. Characterization of signals from multiscale edge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(4):710–732, July 1992. 40, 104

[Ngu00] N. X. Nguyen. *Numerical Algorithms for Image Superresolution*. PhD thesis, Graduate Program in Scientific Computation and Computational Mathematics, Stanford University, July 2000. xvi, 136, 137, 138, 139

[NK03] R. Nakagaki and A. K. Katsaggelos. VQ-based blind image restoration algorithm. *IEEE Transactions on Image Processing*, 12(9):1044–1053, 2003. 23

[NS82] A. W. Naylor and G. R. Sell. *Linear Operator Theory in Engineering and Science*, volume 40 of *Applied Mathematical Sciences*. Springer, 2 edition, 1982. 3, 57

[OEW07] J. Orchard, M. Ebrahimi, and A. Wong. A faster non-local-means image denoising using the FFT, 2007. Submitted to the IEEE Transactions on Image Processing. 44

[OEW08] J. Orchard, M. Ebrahimi, and A. Wong. Efficient non-local-means denoising using the SVD. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2008. 44, 150

[OTS94] M. K. Ozkan, A. M. Tekalp, and M. I. Sezan. POCS-based restoration of space-varying blurred images. *IEEE Transactions on Image Processing*, 3(4):1994, July 1994. 65, 66

[Pap75] A. Papoulis. A new algorithm in spectral analysis and band-limited extrapolation. *IEEE Transactions on Circuits and Systems*, (22):735–742, 1975. 123

[Pap97] A. Papoulis. Generalized sampling expansion. *IEEE Transactions on Circuits and Systems*, 24:652–654, November 1997. 138

[PD97] E. Polidori and J. L. Dugelay. Zooming using iterated function. *Fractals*, 5 (Supplementary Issue):111–123, April 1997. 40

[Puh96] H. Puh. *Set Theoretic Image Coding*. PhD thesis, Courant Institute, New York, 1996. 61

[RB05] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 860–867, San-Diego, California, June 2005. 23

[ROF92] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, November 1992. 22

[SA87] K. Sauer and J. Allebach. Iterative reconstruction of band-limited images from non-uniformly spaced samples. *IEEE Transactions on Circuits and Systems*, pages 1497–1505, 1987. 138

[SC99] H. Shekarforoush and R. Chellappa. Data-driven multi-channel super-resolution with application to video sequences. *Journal of the Optical Society of America A*, 16(3):481–492, March 1999. 138

[Sha49] C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37:10–21, 1949. 139

[Sha98] C. E. Shannon. Classic paper: Communication in the presence of noise. *Proceedings of the IEEE*, 86(2):447–457, 1998. 139

[SO89] H. Stark and P. Oskoui. High-resolution image recovery from image-plane arrays, using convex projections. *Journal of the Optical Society of America*, 6(11):1715–1726, 1989. 138

[SS96] R. Schultz and R. Stevenson. Extraction of high-resolution frames from video sequences. *IEEE Transactions on Image Processing*, 5:996–1011, June 1996. 136, 139

[TA77] A. N. Tikhonov and V. Arsenin. *Solutions of Ill-Posed Problems*. Wiley, New York, 1977. 18, 22

[TH84] R. Tsai and T. Huang. Multi-frame image restoration and registration. In *Advances in Computer Vision and Image Processing*, volume 1, Greenwich, CT, 1984. 139

[TOS92] A. Tekalp, M. Ozkan, and M. Sezan. High-resolution image econstruction from lower-resolution image sequences and space-varying image restoration. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 169–172, San Francisco, CA, 1992. 138

[UG92] H. Ur and D. Gross. Improved resolution from subpixel shifted pictures. *CVGIP: Graphical Models and Image Processing*, 54(2):181–186, March 1992. 138

[Vog02] Curtis R. Vogel. *Computational Methods for Inverse Problems*. SIAM, PA, USA, 2002. 3, 24, 26, 27

[Vrs98] E. R. Vrscay. A generalized class of fractal-wavelet transforms for image representation and compression. *Canadian Journal of Electrical and Computer Engineering*, 23(1-2):69–84, 1998. 39, 40

[Wei98] J. Weickert. *Anisotropic Diffusion in Image Processing.* Teubner, Stuttgart, ecmi series edition, 1998. 22

[Whi29] J. M. Whittaker. The Fourier theory of the cardinal functions. *Proceedings of the Edinburgh Mathematical Society*, 1:169–176, 1929. 139

[WL00] L. Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of SIGGRAPH*, pages 479–488, New Oleans, Louisiana, 2000. 23, 124

[XF] W. Xu and D. Fussell. IFS coding with multiple DC terms and domain blocks. Citeseer article 185324, available at: http://citeseer.ist.psu.edu/185324.html. 39

[You71] D. M. Young. *Iterative Solutions of Large Linear Systems.* Academic Press, New York, 1971. 56

[YW82] D. Youla and H. Webb. Image restoration by the method of convex projections:Part 1-Theory. *IEEE Transactions on Medical Imaging*, MI-1(2):81–94, October 1982. 24

[ZM97] S. C. Zhu and D. Mumford. Prior learning and Gibbs reaction-diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1236–1250, 1997. 21, 22, 23