Multiscale Methods in Image Modelling and Image Processing

by

Simon K. Alexander

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

Waterloo, Ontario, Canada, 2005

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

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

The field of modelling and processing of ‘images’ has fairly recently become important, even crucial, to areas of science, medicine, and engineering. The inevitable explosion of imaging modalities and approaches stemming from this fact has become a rich source of mathematical applications.

‘Imaging’ is quite broad, and suffers somewhat from this broadness. The general question of ‘what is an image?’ or perhaps ‘what is a natural image?’ turns out to be difficult to address. To make real headway one may need to strongly constrain the class of images being considered, as will be done in part of this thesis. On the other hand there are general principles that can guide research in many areas. One such principle considered is the assertion that (classes of) images have multiscale relationships, whether at a pixel level, between features, or other variants. There are both practical (in terms of computational complexity) and more philosophical reasons (mimicking the human visual system, for example) that suggest looking at such methods. Looking at scaling relationships may also have the advantage of opening a problem up to many mathematical tools.

This thesis will detail two investigations into multiscale relationships, in quite different areas. One will involve Iterated Function Systems (IFS), and the other a stochastic approach to reconstruction of binary images (binary phase descriptions of porous media). The use of IFS in this context, which has often been called ‘fractal image coding’, has been primarily viewed as an image compression technique. We will re-visit this approach, proposing it as a more general tool. Some study of the implications of that idea will be presented, along with applications inferred by the results. In the area of reconstruction of binary porous media, a novel, multiscale, hierarchical annealing approach is proposed and investigated.

Acknowledgements

I wish to thank my supervisors Edward Vrscay and Paul Fieguth for support and encouragement during the research and writing of this thesis. They have both been valuable resources, friendly collaborators and will always be credited with helping to generally make my time here very enjoyable.

This research was supported in various ways by grants and/or scholarships from the National Sciences and Engineering Council, Ontario Graduate Scholarships, the Faculty of Mathematics and my department of Applied Mathematics. I am deeply appreciative of this aid.

Many colleagues and faculty members have offered insight and fruitful discussion over the tenure of this work, which was of great value to me.

CONTENTS

LIST OF FIGURES

CHAPTER 1

Introduction

In this thesis, a program of research will be described that has examined aspects of the use of multiple scales for image processing applications. A general and qualitative discussion is presented in the introduction, in order to facilitate covering a lot of ground. Following sections will introduce more narrow and focused discussion, in particular we will discuss and build upon *fractal* Iterated Function System (IFS) methods, also and stochastic reconstruction/sampling approaches for a particular class of images. These two seemingly disparate topics are tied together by the aspect of relationships between scales.

The above statements give rise to two questions, namely "why is scale interesting?", and "what do I mean by an image?". Before addressing the former, I will tackle the latter point. This turns out to be more difficult than a first glance might suggest.

1.1 What is an image?

Image processing and image science are relatively new areas of study in mathematics. They provide many applications, and admit a large number of possible approaches. There are image processing and image science applications of computational geometry, partial differential equation (PDE) theory & variational methods (especially via level-sets method) [OS88, FK98, MS95], statistical & probabilistic approaches [Win91b, Bré98, Vid99], and information theory [Sha48, JM98, Vio95] to name a few. A large amount of work in the area has been done by the engineering community, especially from the area of digital signal processing, filter-banks, etc.

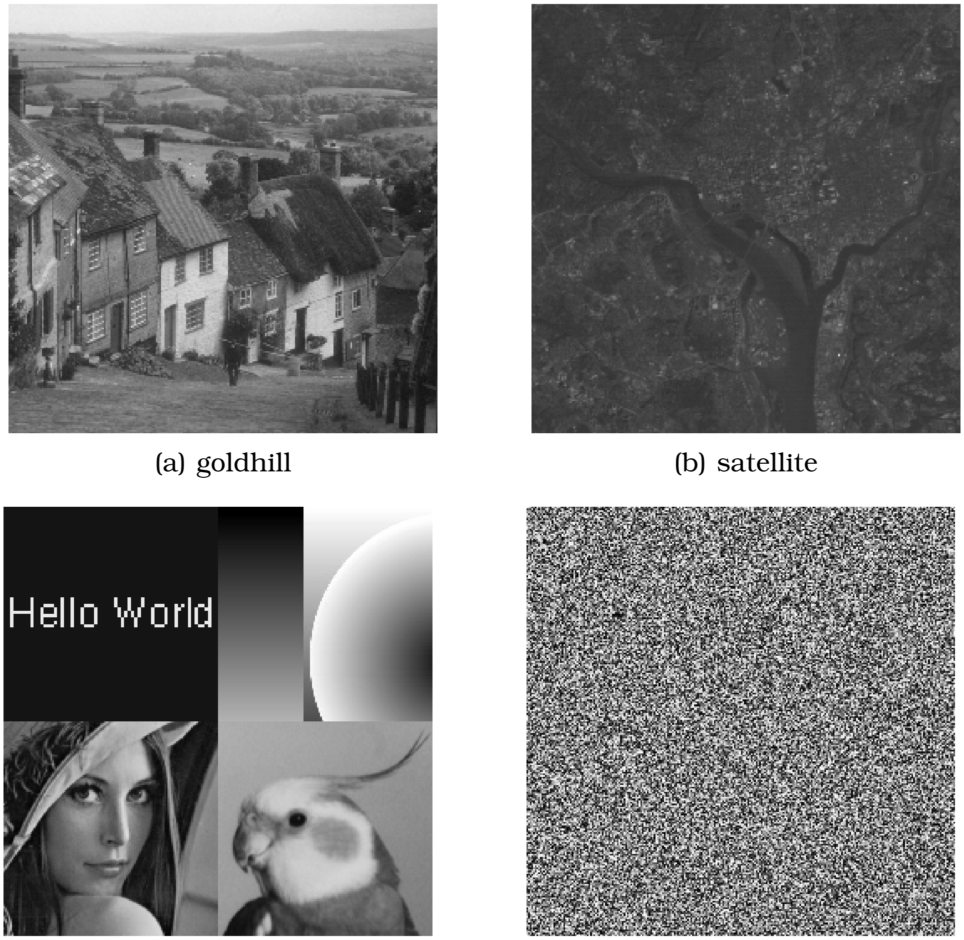
However, many questions in imaging remain unanswered. Even so fundamental a question as "what is an image?" still lacks a satisfactory answer. This is a concept which is easy to roughly convey, but a rigorous formalism for the general case has yet to be established.

In deference to this difficulty, and to motivate further discussion, Figure 1.1 shows a few examples[1]. Particular examples of some test images used in experiments contained in chapter 3 are given in Figure 6.1. Even these examples from a small subset of all possible images live in a fairly large discrete space. In this report, an "image" will generally be constrained to a discrete lattice as above, unless otherwise noted. It is certainly possible to consider digital images as quantisations of some continuous space, for example a subspace of all functions $f : \mathbb{R}^2 \to \mathbb{R}^+$.

In Figure 1.1-a through -c we see three different examples of what may be termed "images". However, brief consideration should convince the reader that the typical member of this space is not an image at all. Figure 1.1-d shows a typical example , with each picture element (pixel) location independently sampled randomly on the integers 0..255.

These examples all have dimensions $256{\times}256$ pixels, and each pixel takes one of 256 intensity values (i.e., it is an 8-bit quantity). Thus they belong to a subclass of images which has $256^{256^2} = 2^{524288} \approx$

[1] The use of 'typical' here is somewhat misleading, as the image 1.1-d is a very particular type of noise (independent, identically distributed (iid) white noise) sample, not really a typical image! The purpose is illustrative, not exact.



(a) goldhill

(b) satellite

(c) montage

(d) ‘typical’ sample

Figure 1.1: Examples from a space containing “natural images”

$10^{150,000}$ members.

The size of this *configuration space*, and the fact that the objects of interest (images) lie sparsely in the space, is at the root of many of the difficulties in image processing. This will have important consequences in §4.2.

This whole process is further complicated by the fact that in practice image processors often wish to deal with *noisy images*, i.e. objects that are not really "images" in the desired sense, but are the result of corruption of an "image" by some (often unknown) noise process. Although this research will not primarily be concerned with the problem of noise, it is an important consideration.

In order to mathematically define operations on images (i.e., perform image processing), we must first be able to define an object upon which such operations may be applied. The problem of attempting to characterise the space of "images" lying within the much larger *configuration space* of pixels may be termed *image modelling*.

Image modelling

As previously suggested, there are many possible approaches to image modelling. In this section, a brief overview of some methods is offered. The focus will be on random field methods and wavelets, leading to the proposed research. Some unrelated areas are briefly discussed.

Image models are essentially approaches to dealing with the problems introduced in the last section. In other words, all image models, to some degree, attempt to answer the question: Given the space of all possible pixel configurations, what does the subset of "images" look like?

One possible answer is to allow all possible configurations to be images. While this allows a simple rigorous definition of an image (continuous or discrete, once you have decided how to define a *pixel*), it is an unsatisfying approach. This 'abstraction' is no help in differentiating the types of images we wish to deal with from pure noise.

Thus we would like a description (model) of images that differentiates the objects we are interested in from the rest of the space. On the other hand, perhaps there is no such clean boundary to define a "subspace of images", especially when considering all possible images. Hence there are three possibilities:

1. There is an as yet undiscovered image model that captures the idea of what a human being means by an "image".

2. There is no such model, but there are classes of analytic models that are close enough (in some sense relevant to the application).

3. There is no such model; stochastic approaches are more suitable.

The first possibility seems unlikely. The human visual system is a complicated psycho-physical process with immense complexity. The question "is this an image?" may not have an unambiguous answer [Kos94, Wan95].

The second possibility is the focus of much research. This approach has been implicitly asserted by many ad-hoc image processing algorithms. There are also many explicit models, such as the region-edge approach of Mumford & Shah[MS89], functions of bounded variation of Rudin & Osher [ROF92, RO94]. Recent developments include Meyers expansion on the Rudin & Osher approach [Mey01].

Third in this list is the idea that stochastic approaches may be the most appropriate representation. This approach, or more accurately a Bayesian interpretation of this approach, will be one focus of this thesis. More detail on this type of approach can be found in several good sources, for example [Win91b, Bré98]. Stochastic image models have enjoyed success in several areas of image processing, such as restoration [GG84, DJ94], texture synthesis [CJ83], and segmentation [BS94].

Bayesian approach

The following equation is usually referred to as Bayes' Law:

$$P(Y)P(X|Y) = P(Y|X)P(X), \tag{1.1}$$

where $P(\cdot)$ denotes probability, $a|b$ denotes event a conditioned on event b. By introducing the suggestive event names X_o for an *observed* event (data) and X_i an *inferred* event, and re-ordering we have

$$P(X_i|X_o) = \frac{P(X_o|X_i)P(X_i)}{P(X_o)}. \tag{1.2}$$

It is worth noting that while the observed event, or data, will normally be an image in this context, there is no reason that the inferred event must also be an image; the inference could be about any feature of interest.

Equation 1.2 should make clear the appeal of the Bayesian framework for the modelling of images. The left hand side describes the probability of an inferred result X_i given an observed result X_o, i.e. the *posterior probability*. The quantity $P(X_i)$ represents the *a-priori* probability of observing X_i independent of observations, so constitutes a *prior model* for the inference. Thus to give an estimate, one needs to find the inference X_i that maximises the posterior probability $P(X_i|X_o)$. By Equation 1.2, we see that it is sufficient to maximise the product of the prior $P(X_i)$ and the *data model* $P(X_o|X_i)$, since for a given observation $P(X_o)$ is constant.

Bayesian modelling is *not* used in this thesis to any significant extent. Two things make it worth mentioning, though. First, it probably represents the first unified and general modelling approach applied to images. Second, parts of the results from this thesis, in particular chapter 5, can be applied in such a framework in a straightforward way. What we are attempting in chapter 5 is in some sense *pure* sampling from a prior model, (e.g. *a priori*). Application of Bayes Law would allow a similar approach to estimation problems via *a posteriori* estimates.

1.2 A note on the comparison on images

The “quality” of a processed image is not easy to quantify[(VQ00]. The result of most *image processing* operations is an approximation to the (perhaps unknown) original signal. What is required is a measure of the inaccuracy of this approximation. Quantification is complicated by the fact that in practice what is actually important is the *perceived* inaccuracy (to a human observer) of the approximation. Since the human visual system is itself a lossy and non-linear process, this becomes difficult to model. Notwithstanding these difficulties, there are several approaches; the error introduced by the approximation is termed the *distortion*.

One obvious approach is to model the image within a metric space (usually $\mathcal{L}^2([0,1])$) and use the distance function to define the error. The usual approach is arrived at from the $\ell^2(\mathbb{N})$ distance and often termed the *mean-squared error*

Definition 1.1 (RMSE) *Define the Root Mean Squared Error (RMSE) between sequences (and similarly for images) $\{x_n\}_{n=1}^N$ and $\{y_n\}_{n=1}^N$ as*

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x_n - y_n)^2} \tag{1.3}$$

□

The above is sometimes reported as Mean Squared Error (MSE), i.e. $\sqrt{\mathrm{MSE}} = \mathrm{RMSE}$.

The resulting values of MSE are somewhat inconvenient to compare. In order to address this problem, introduce the following definition.

Definition 1.2 (PSNR) *The Peak Signal to Noise Ratio (PSNR) is defined in terms of the MSE. If the peak value[2] obtainable by x_n is $\bar{x}$*

[2] For example, 8-bit pixels range from 0 to 255, so the peak value is 255.

then define the PSNR as

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\bar{x}^2}{\mathrm{MSE}}\right) = 20\log_{10}\left(\frac{\bar{x}}{\mathrm{RMSE}}\right) \quad \textit{in decibels [dB]} \tag{1.4}$$

□

The PSNR will, in practice, have values in the range of about 10 to 50 dB (with the exception that the PSNR of an image with respect to itself is infinite). A value of less than 10 dB would suggest that the images are unrelated, whilst 40 dB or above is often imperceptible to the eye.

Although the above method is the *de-facto* standard, it has obvious flaws[3]. In practice, it is reasonably effective in that a high PSNR is usually sufficient to ensure small (to the human observer) distortion. On the other hand, any algorithm that can induce small translations of image features will fare poorly in PSNR, but not necessarily for an observer. The problem of improved distortion measurements seems quite difficult, and remains open. For that reason this work uses PSNR exclusively. Discussion of applying information about the human visual system to distortion metrics may be found in [Pra78, Cam68]. There has been some work done in application of such metrics to image compression, for example [Nil85].

1.3 Image processing

Image processing is a large field, and there are many ‘standard’ tasks that may be performed. Some examples would be:

- restoration : *given an image corrupted (by noise, optical deformities, etc.) estimate the ‘true’ image.*
- compression : *reduce the storage needed for an image.*
- edge detection : *describe the boundaries of ’objects’ in an image.*

[3] Consider an image shifted in any direction by one pixel.

- classification : *separate the objects in an image into 'types'.*
- *...many others*

Equation 1.2 is a very natural framework in which to cast many image processing tasks. Given some observation, and a prior model for a (class of) images, we find the event (inference) that maximises the product of the likelihood of a particular event and the likelihood of seeing our data, given that event.

In the context of Bayesian image processing, performing these tasks involves roughly the same process each time:

1. construct an appropriate prior model.
2. use statistical inference to infer free parameters (parameter estimation).
3. specify the form of the posterior distribution (i.e., how do you actually calculate $P(X_i|X_o)$?).
4. finally, given some data (e.g. and image), estimate the 'true' image or feature.

To be able to address an image processing application , we need to be able to describe a prior model, and to evaluate the given probabilities. The particulars of these problems will in general vary from application to application, and it is of course these details that can make things interesting. In §4.2, we will be particularly interested in describing "energy" based formulations of this approach.

1.4 The importance of scale in images

One intriguing facet of imaging problems is the fact that human beings perform some actions effortlessly that are exceedingly hard to even begin to do algorithmically. One aspect that seems to be very important to humans is the recognition of scales within an image.

In this work, the role of multiple scales in images, and how this may be exploited, is explored. There are, of course, many possible methods for incorporating scale into models and algorithms in images [Fie93, Fro99, BS94, SFAH92]. This thesis will particularly focus on two areas.

Given that scale, or more particularly multiple scales, can be important in image processing, one approach is to look for a representation of data that naturally reflects multiple scales. Although by no means unique, the wavelet approach has proven to be quite effective for some areas of image processing. There is also support for the idea that wavelet transforms mimic some of the methods of the Human Visual System (HVS) [Fie93, Wan95, WA85].

Qualitatively, wavelet bases form a 'spatial-frequency' space, in some ways analogous to the Fourier frequency space. One feature of these bases is that coefficients explicitly represent particular scales of the image. Wavelet bases and related concepts are important representational approaches to multiscale imaging. Section §2.4 will discuss a particular approach leveraging this idea.

1.5 Layout of the thesis

In this thesis, a programme of research investigating particular aspects of multiscale approaches to imaging is described. The area is very large, and so any such work risks overly ambitious goals. The thesis will focus on two areas, for this reason.

The thesis falls into two logical sections, as there are two quite distinct areas of investigation. Following this introduction, background development on IFS as applicable to images is presented in chapter 2. After these preliminaries, a number of experiments and applications of IFS are discussed in chapter 3. Chapter 4 introduces the idea of Random Fields, Sampling, and Simulated Annealing, providing background for the the more specifically focused material to follow. Chapter 5 then focuses on porous media reconstruction as an application, and construction of a Hierarchical Simulated Annealing approach to this problem. Experiments and discussion on such sampling, as well as modelling issues are included. The aforementioned four chapters form the basis of this thesis, and are followed by a brief summary and concluding remarks.

1.6 Preliminaries and Notation

Throughout this thesis, new notation will be introduced as needed. Some common notations are included here.

Sets

Denote the following classical sets as shown

$$\begin{aligned}
\mathbb{R} &= \text{the set of real numbers,}\\
\mathbb{C} &= \text{the set of complex numbers,}\\
\mathbb{Z} &= \text{the set of integers,}\\
\mathbb{N} &= \text{the positive integers including } 0 = \{0, 1, 2, 3, \ldots\},\\
\mathbb{N}^{+} &= \text{the positive integers} = \{1, 2, 3, \ldots\}.
\end{aligned}$$

Limits of Integration

Where limits on an integral or summation are not shown, they are assumed to be infinite. That is

$$\int f(x)\,dx = \int\limits_{-\infty}^{+\infty} f(x)\,dx$$

$$\sum_n x_n = \sum_{n=-\infty}^{+\infty} x_n$$

and similarly,

$$\sum_{j,k} x_{j,k} = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} x_{j,k}$$

Complex Conjugate

The complex conjugate of a variable or function will be denoted by an overbar, such as

$$\overline{x} \qquad \text{and} \qquad \overline{f(x)}$$

Fourier Transform

Denote the continuous Fourier transform as

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} f(x) e^{-i\omega x}\, dx$$

and the discrete Fourier transform for a signal f of period N is

$$\hat{f}[k] = \sum_{n=0}^{N-1} f[n] e^{-2i\pi n \frac{k}{N}}.$$

$\mathcal{L}^p$ Spaces

The space $\mathcal{L}^p(\mathbf{X})$, $p \in \mathbb{N}^+$ describes an equivalence class of Lebesgue integrable functions on **X** is defined as

$$\mathcal{L}^p(\mathbf{X}) = \{ f : \mathbf{X} \rightarrow \mathbb{R} \mid \int_{\mathbf{X}} |f(x)|^p\, dx < \infty \} \ .$$

ℓ^p Spaces

The space $\ell^p(\mathbf{X})$, $p \in \mathbb{N}^+$ of p-summable sequences on $\mathbf{X}$ is defined as the set

$$\ell^p(\mathbf{X}) = \{\{x_n\}_{n \in \mathbb{N}}, x_n \in \mathbf{X} \mid \sum_{n=0}^{\infty} |x_n|^p < \infty\}.$$

Kronecker delta

Denote the Kronecker delta as

$$\delta_j^k = \begin{cases} 1 & j = k, \\ 0 & j \neq k. \end{cases}$$

Proofs, Theorems, and Examples

Proofs will be ended by the symbol ■ at the far side of the text. Similarly, theorems, examples, etc. will be delimited at the end by the symbol □.

Algorithms

Where it is appropriate to do so, algorithms may be presented in pseudo-code. Notation contained in these code blocks is mostly self-explanatory. The notation $\Leftarrow$ will be used for "takes the value of", and braces $\{comment\}$ used for comments, for example.

Algorithm 1.1 Example pseudo-code

```
$x \Leftarrow 1.0$ {$x$ is now 1.0}
for $i = 0$ until 10 do
  do something ...
end for
```

Several algorithms discussed in this thesis are designed to be implemented in a computer; it is hoped that their operation is made clearer by presentation in this format. Computer code used in many of the experiments reported herein, primarily in the Common Lisp language, will be made available upon request.

CHAPTER 2

Fractal Transforms

This chapter will form the background necessary to address the idea of 'fractal coding' of images. The general approach here will be the approximation of an image by the fixed point of a contractive operator.

Before attacking images,though, we will begin with a more general discussion. The Iterated Function Systems (IFS) theory is quite simple, and as many of the required proofs are straightforward and short, it is possible to present here a reasonably complete development of the basic theory. Some proofs are omitted, with citations, to simplify the discussion. A good introduction to IFS can be found in [Bar93], while a more complete discussion of the theory underlying IFS can be found in [Fal97, ch.1,2] and [Fal90, ch.1,2,9].

After development of needed background the inverse problem for IFS is discussed; it may be stated roughly as: "given a particular set, is is possible to generate an IFS whose invariant set will approximate it?". The final goal is to apply these ideas to images, both spatially and in the wavelet domain. These will be presented as *IFSM* and *IFSW*, respectively.

2.1 Overview

The Idea of IFS

Example 2.1 Take, as an initial example, the middle third Cantor set, call it C. Now C is a union of two (left and right) scaled copies of itself. Consider the two mappings $S_1(x) = \frac{1}{3}x$ and $S_2(x) = \frac{1}{3}x + \frac{2}{3}$ defined on $(\mathbb{R}, d)$ where $d(x, y) = \mid x - y \mid$. Then S_1 and S_2 are contraction mappings and

$$C = S_1(C) \cup S_2(C). \tag{2.1}$$

Thus C is invariant under the map $S(X) = S_1(X) \cup S_2(X)$. Invariance under this map completely specifies the set C (as will be shown in the following, see Theorem 2.4). □

Roughly speaking, families of contraction maps like S are called IFS (an exact definition will be presented in the following sections). Many such invariant sets are fractals[1]. The terminology of fractals is quite context dependent, but seems to universally include self-similarity of the set (invariance of scale) in some way.

2.2 IFS Background

Notation 2.1 *Denote a metric space of the metric d on the set $\mathbf{X}$ by $(\mathbf{X}, d)$.* □

Notation 2.2 *Consider a function $f : \mathbf{X} \to \mathbf{X}$. Denote the n^{th}* ***iteration*** *or n-fold composition, of f to be $f^{\circ(n)}$, which is defined as*

$$f^{\circ(1)}(x) := f(x) \tag{2.2}$$

$$f^{\circ(n+1)}(x) := f(f^{\circ(n)})(x) \qquad \textit{for } n \textit{ in } \mathbb{N}^+,\, n > 1\,. \tag{2.3}$$

□

[1] A rather difficult term to define.

Definition 2.1 (attractor) *Consider a function $f : \mathbf{X} \to \mathbf{X}$. If there exists a $y \in X$ such that $lim_{n\to\infty} f^{\circ(n)}(x) = y$, $\forall x \in X$ then y is called the **attractor** of f.* □

Definition 2.2 (fixed point) *If a function f has an argument with the property that $f(x) = x$, call x a **fixed point** of f, and denote it $\bar{x}_f$ (or $\bar{x}$ where there is no possibility of ambiguity).* □

In the language of fractals, the term *attractor* is often used interchangeably with that of a *fixed point* due to the following property:

Proposition 2.1 *Let $f : \mathbf{X} \to \mathbf{X}$ be a continuous function. If f has an attractor $x \in X$, then x is a fixed point.* □

PROOF By continuity, $f(x) = f(lim_{n\to\infty} f^{\circ(n)}(x)) = lim_{n\to\infty} f^{\circ(n+1)}(x) = x$. ■

Clearly it is possible to have fixed points which are not attractors, but these will not be relevant to the development.

Contractive Maps

The following class of functions is central to the idea of IFS.

Definition 2.3 (Lipschitz) *Consider a function f on a metric space $(\mathbf{X}, d)$, $f : \mathbf{X} \to \mathbf{X}$. If there exists an $s \in [0, \infty)$ such that $\forall x, y \in \mathbf{X}$,*

$$d(f(x), f(y)) \le sd(x, y). \tag{2.4}$$

*Then f is called **Lipschitz** on $\mathbf{X}$, and s is called a **Lipschitz constant** for f.* □

Definition 2.4 (contractive) *If a function f has a Lipschitz constant $c < 1$, f is called **contractive**, or a **contraction**.*

Proposition 2.2 (contractivity factor) *Let f be a contractive function on $\mathbf{X}$. Let C be the set of all contractivity factors of f. Define $c_f = \inf(C)$. Then c_f is a contractivity factor of f, and c_f is called **the** contractivity factor of f.* □

PROOF Let $x, y \in X$, then $\forall c \in C$

$$d(f(x), f(y)) \leq cd(x, y) \tag{2.5}$$

which implies,

$$\begin{aligned} d(f(x), f(y)) &\leq \inf(C)d(x, y) \qquad (2.6) \\ &= c_f d(x, y). \qquad (2.7) \end{aligned}$$

Thus c_f is a contractivity factor for f, and

$$d(f(x), f(y)) \leq c_f d(x, y). \tag{2.8}$$

■

Definition 2.5 *Let $(\mathbf{X}, d)$ be a complete metric space. Denote the set of all contraction mappings on $\mathbf{X}$ to be $\mathrm{Con}(\mathbf{X}, d)$.* □

The following properties of the set $\mathrm{Con}(\mathbf{X}, d)$ are important to the applications that will be presented later.

Lemma 2.1 *Let f be a contractive map on the metric space $(\mathbf{X}, d)$. Then f is uniformly continuous.* □

PROOF Let $\epsilon > 0$. Consider the contractivity factor of f, c. Suppose $c \neq 0$, and let $\delta = \frac{\epsilon}{c}$. Thus $\forall d(x, y) < \delta$

$$d(f(x), f(y)) \leq cd(x, y) < \epsilon. \tag{2.9}$$

The inequality holds trivially for the case $c = 0$. ■

The continuity of f is important in that members of $\mathrm{Con}(\mathbf{X}, d)$ will map compact sets to compact sets. A further useful property of

$\mathrm{Con}(\mathbf{X}, d)$ is that, under an appropriate metric, fixed points vary continuously under contractive maps [CV94]. In order to show this, a suitable metric is required.

Proposition 2.3 *Let* $(\mathbf{X}, d)$ *be a compact metric space. Then* $d_{\mathrm{Con}(\mathbf{X},d)} : \mathrm{Con}(\mathbf{X}, d) \to [0, \infty)$ *defined as*

$$d_{\mathrm{Con}(\mathbf{X},d)}(f, g) = \min\{1, \sup_{x \in \mathbf{X}} d(f(x), g(x))\} \quad \forall\ f, g \in \mathrm{Con}(\mathbf{X}, d) \tag{2.10}$$

is a metric on the space $\mathrm{Con}(\mathbf{X}, d)$. *The min with 1 is taken to avoid infinite values.* □

PROOF Let $f, g, h \in \mathrm{Con}(\mathbf{X}, d)$. Assume $\sup_{x \in \mathbf{X}} d(f(x), g(x)) < 1$.

$$d_{\mathrm{Con}(\mathbf{X},d)}(f, g) \geq 0 \qquad \text{as } d(f(x), g(x)) \geq 0\ \forall x \in \mathbf{X} \tag{2.11}$$

$$d_{\mathrm{Con}(\mathbf{X},d)}(f, g) = 0 \iff f = g \qquad \text{by definition of sup} \tag{2.12}$$

$$d_{\mathrm{Con}(\mathbf{X},d)}(f, g) = d_{\mathrm{Con}(\mathbf{X},d)}(g,\ f) \qquad \text{by symmetry of } d. \tag{2.13}$$

And the triangle inequality:

$$d_{\mathrm{Con}(\mathbf{X},d)}(f, g) = \sup_{x \in \mathbf{X}} d(f(x), g(x)) \tag{2.14}$$

$$\leq \sup_{x \in \mathbf{X}} \{d(f(x), h(x)) + d(h(x), g(x))\} \tag{2.15}$$

$$\leq \sup_{x \in \mathbf{X}} d(f(x), h(x)) + \sup_{x \in \mathbf{X}} d(h(x), g(x)) \tag{2.16}$$

$$= d_{\mathrm{Con}(\mathbf{X},d)}(f, h) + d_{\mathrm{Con}(\mathbf{X},d)}(h, g). \tag{2.17}$$

The case where $\sup_{x \in \mathbf{X}} d(f(x), g(x)) \geq 1$ is similar. ■

Now, applying this metric, the continuity of fixed points can be shown. This result (introduced in [CV94]) is important in applications of the Collage Theorem (2.5), which will be described later.

Theorem 2.1 (Continuity of Fixed Points) *Let* $(\mathbf{X}, d)$ *be a compact metric space and* $f, g \in \mathrm{Con}(\mathbf{X}, d)$ *with fixed points* $\bar{x}_f$, $\bar{x}_g$ *and contraction factors* c_f, c_g, *respectively. Define the function* $F(f) : \mathrm{Con}(\mathbf{X}, d) \to$

$\mathbf{X}$ by

$$F(f) = \bar{x}_f \quad \forall f \in \text{Con}(\mathbf{X}, d). \tag{2.18}$$

Then F is continuous under $d_{\text{Con}(\mathbf{X},d)}$. □

PROOF Let $f, g \in \text{Con}(\mathbf{X}, d)$. Let $\epsilon > 0$. Let $\delta = \epsilon(1 - c_f)$. Then for all δ such that $d_{\text{Con}(\mathbf{X},d)}(f, g) < \delta$,

$$\begin{aligned}
d(\bar{x}_f, \bar{x}_g) &= d(f(\bar{x}_f), g(\bar{x}_g)) && (2.19)\\
&\leq d(f(\bar{x}_f), f(\bar{x}_g)) + d(f(\bar{x}_g), g(\bar{x}_g)) && (2.20)\\
&\leq d(f(\bar{x}_f), f(\bar{x}_g)) + d_{\text{Con}(\mathbf{X},d)}(f, g) && (2.21)\\
&\leq c_f d(\bar{x}_f, \bar{x}_g) + d_{\text{Con}(\mathbf{X},d)}(f, g) && (2.22)\\
&< c_f d(\bar{x}_f, \bar{x}_g) + \epsilon(1 - c_f) \qquad \text{by hypothesis.} && (2.23)
\end{aligned}$$

Therefore,

$$d(\bar{x}_f, \bar{x}_g) < \epsilon. \tag{2.24}$$

■

Since we can reverse the roles of f and g, we have the following corollary.

Corollary 2.1 *Let $(\mathbf{X}, d)$ be a compact metric space, and let $f, g \in \text{Con}(\mathbf{X}, d)$, and define $c = min\{c_f, c_g\}$. Then*

$$d(\bar{x}_f, \bar{x}_g) < \frac{1}{1-c} d_{\text{Con}(\mathbf{X},d)}(f, g). \tag{2.25}$$

□

Now that the necessary background has been filled in, the central theorem on which IFS depends may be presented.

The Contraction Mapping Principle

The theory of IFS is centrally reliant on the Contraction Mapping Principle (CMP) (due to Banach [Ban22]). The CMP identifies unique fixed points of contractive maps; these fixed points are the attractors of any iterative process on the maps.

Theorem 2.2 (Contraction Mapping Principle) *Let $(\mathbf{X}, d)$ be a complete metric space, and $f : \mathbf{X} \to \mathbf{X}$ be a contractive map with contractivity factor c. Then f has a unique fixed point $\bar{x}_f$, and $\bar{x}_f$ is an attractor of f.* □

PROOF Construct a sequence $\{x_n\}_{n=0}^{\infty}$ in the following manner:

$$\text{pick any } x_0 \in \mathbf{X}, \quad x_{n+1} = f(x_n). \tag{2.26}$$

Now for any $n, m \in \mathbb{N}^+$ with $m > n$,

$$\begin{aligned} d(x_n, x_m) &= d(f^{\circ(n)}(x_0),\ f^{\circ(m)}(x_0)) && (2.27)\\ &\le cd(f^{\circ(n-1)}(x_0),\ f^{\circ(m-1)}(x_0)) && (2.28)\end{aligned}$$

since f is contractive. The final line follows from Equation 2.8. If f is iterated (from equation 2.4) n times, we find that

$$d(x_n, x_m) \le c^n d(x_0,\ f^{\circ(m-n)}(x_0)). \tag{2.29}$$

Now let $k \in \mathbb{N}^+$ and by repeated application of the triangle inequality:

$$\begin{aligned} d(x_0,\ f^{\circ(k)}(x_0)) &\le d(x_0,\ f(x_0)) + d(f(x_0),\ f^{\circ(k)}(x_0)) && (2.30)\\ &\le d(x_0,\ f(x_0)) + d(f(x_0),\ f^{\circ(2)}(x_0)) + && (2.31)\\ &\quad \ldots + d(f^{\circ(k-1)}(x_0),\ f^{\circ(k)}(x_0)) && (2.32)\\ &\le d(x_0,\ f(x_0)) + cd(x_0,\ f(x_0)) + c^2 d(x_0,\ f(x_0)) + && (2.33)\\ &\quad \ldots + c^{k-1} d(x_0,\ f(x_0)) && (2.34)\\ &= \sum_{i=0}^{k-1} c^i d(x_0,\ f(x_0)). && (2.35)\end{aligned}$$

Hence by comparison to a geometric series in c,

$$d(x_0,\ f^{\circ(k)}(x_0)) \leq \frac{1}{1-c} d(x_0,\ f(x_0)). \tag{2.36}$$

Combining this result and equation 2.29 yields

$$d(x_n, x_m) \leq \frac{c^n}{1-c} d(x_0,\ f(x_0)). \tag{2.37}$$

Since $c < 1$, $d(x_n, x_m) \to 0$ as $n \to \infty$. Thus by equation 2.37 $\{x_n\}$ is a Cauchy sequence. Therefore $\{x_n\}$ converges in the complete space $(\mathbf{X}, d)$. That is, $\exists\ \bar{x}$ such that $x_n \to \bar{x}$. Now suppose that there is another such fixed point, $y \in X$ such that $x_n \to y$. So

$$\begin{align} d(\bar{x}, y) &= d(f(\bar{x}),\ f(y)) && \text{as } \bar{x}, y \text{ are fixed points} \tag{2.38} \\ &\leq c d(\bar{x}, y) && f \text{ is contractive.} \tag{2.39} \end{align}$$

However, this inequality is only satisfied by $d(\bar{x}, y) = 0$, since $0 < c < 1$. And since d is a metric, this implies that $\bar{x} = y$, hence $\bar{x}$ is unique. ■

With this result, the background is nearly in place to define an IFS. Before doing so a convenient complete metric space is introduced, in which to work.

2.2.1 A Complete Metric Space for Iterated Function Systems

This section will briefly outline the development of a complete metric space suitable for IFS. We loosely follow [Bar93] (where a more complete discussion may be found).

Definition 2.6 (Hausdorff Space) *For $(\mathbf{X}, d)$ a complete metric space, the so-called Hausdorff space $\mathcal{H}(\mathbf{X})$ denotes the set of all non-empty compact subsets of $\mathbf{X}$.* □

In approaching a distance function to use in $\mathcal{H}(\mathbf{X})$, we first address the distance between a point and a set.

Notation 2.3 *Let* $(\mathbf{X}, d)$ *be a complete metric space,* $x \in \mathbf{X}$*, and* $B \in \mathcal{H}(\mathbf{X})$*. Define* $d(x, B)$*, the distance from the point* x *to the set* B *as*

$$d(x, B) = \inf_{b \in B}\{d(x, b)\}. \tag{2.40}$$

□

Now consider the distance between two sets.

Notation 2.4 *Let* $(\mathbf{X}, d)$ *be a complete metric space. Let* A*,* $B \in \mathcal{H}(\mathbf{X})$*. Define* $d(A, B)$*, the distance from the set* A *to the set* B *as*

$$d(A, B) = \sup_{a \in A}\{d(a, B)\}. \tag{2.41}$$

□

However, this will not quite form a metric (it clearly lacks symmetry) so the following modification is introduced.

Definition 2.7 (Hausdorff Distance) *Let* $A, B \in \mathcal{H}(\mathbf{X})$*. Define the* ***Hausdorff distance*** *between* A *and* B *as*

$$h(A, B) = \max\{d(A, B), d(B, A)\}. \tag{2.42}$$

□

The following lemma succinctly describes the intuitive sense of the Hausdorff metric: Two sets are ϵ-close under h if and only if they are fully contained in the ϵ extension of each other.

Lemma 2.2 *Let* $(\mathbf{X}, \mathbf{d})$ *be a metric space,* $A, B \in \mathcal{H}(\mathbf{X})$*,* $\epsilon > 0$*. Then*

$$h(A, B) \leq \epsilon \textit{ iff } A \subset B_\epsilon \textit{ and } B \subset A_\epsilon \tag{2.43}$$

where $A_\epsilon = \{x \in X | \inf_{a \in A} d(x, a) < \epsilon\}$ *and* $B_\epsilon = \{x \in X | \inf_{b \in B} d(x, b) < \epsilon\}$*.* □

PROOF Omitted, see (Bar93). ■

Theorem 2.3 *Let $(\mathbf{X}, d)$ be a complete metric space. Then $(\mathcal{H}(\mathbf{X}), \mathbf{h})$ is a complete metric space. Furthermore, if $\{A_n\}_{n=1}^{\infty}$ is a Cauchy sequence in $(\mathcal{H}(\mathbf{X}), h)$, then $A = \lim_{n\to\infty} A_n \in \mathcal{H}(\mathbf{X})$ can be characterised as:*

$$A = \{x \in \mathbf{X} : \exists \textit{ Cauchy sequence } \{\mathbf{x_n} \in \mathbf{A_n}\} \textit{ converging to } \mathbf{x}\} \ . \tag{2.44}$$

□

PROOF See (Bar93, ch.2). ■

With this result, we have arrived at a complete metric space suitable for IFS. arrived at.

2.2.2 Iterated Function Systems Defined

Definition 2.8 (IFS) *An Iterated Function Systems (IFS) consists of a complete metric space $(\mathbf{X}, d)$ and a finite set of contraction mappings $\mathbf{w} = \{w_1, w_2, \ldots, w_N\}$ where $w_n : \mathbf{X} \to \mathbf{X}$, each with contraction factor c_n respectively, for $n = 1, 2, \ldots, N$. The IFS has contraction factor $c = \max\{c_1, c_2, \ldots, c_N\}$. The IFS can be denoted $\{\mathbf{X}; w_n, n = 1, 2, \ldots, N\}$.*□

It remains to be shown that the above system is contractive, and has the desired contraction factor. Before doing so, a few lemmas are introduced, beginning with continuous maps in the metric space inducing maps on the Hausdorff space.

Lemma 2.3 *Let f be a continuous map on the metric space $(\mathbf{X}, d)$. Then f maps $\mathcal{H}(\mathbf{X}) \to \mathcal{H}(\mathbf{X})$.* □

PROOF Let S be a non-empty compact subset of $\mathbf{X}$. Then $\tilde{f}(S) = \{f(s) : s \in S\}$ is non-empty. Consider a sequence $\{y_n\}_{n=1}^{\infty} \in \tilde{f}(S)$. There exists an associated sequence, $\{x_n\}_{n=1}^{\infty} \in S$, such that $y_n = f(x_n)$. Now $\{x_n\}$ is an infinite sequence contained in a compact set, so there exists a convergent subsequence $\{x_{n_k}\}$ with $x_{n_k} \to \bar{x}$ as n (and hence, n_k) tends to infinity. By Lemma 2.1, f is continuous. Thus y_{n_k} converges, that is

$y_{n_k} \to \bar{y} = f(\bar{x})$ as n tends to infinity. Therefore, $\tilde{f}(S)$ is compact, and $\tilde{f}(S) \in \mathcal{H}(\mathbf{X})$. ■

Having shown that sets under such functions will remain in the Hausdorff space, the goal is to to develop a version of the CMP (Theorem 2.2) for IFS. This result follows directly from the following lemmas, and the CMP. Proofs are straightforward, and omitted. Details may be found in [Bar93, Fal90].

Definition 2.9 *Let $f \in \mathrm{Con}(\mathbf{X}, d)$. Define the map $\tilde{f} : \mathcal{H}(\mathbf{X}) \to \mathcal{H}(\mathbf{X})$ by*

$$\tilde{f}(S) = \{f(s) : s \in S\}. \tag{2.45}$$

□

Lemma 2.4 *Let f be a contractive map on the metric space $(\mathbf{X}, d)$, with contraction factor c_f. Then $\tilde{f} : \mathcal{H}(\mathbf{X}) \to \mathcal{H}(\mathbf{X})$, $\tilde{f}(S) = \{f(s) : s \in S\}$ is a contractive on the metric space $(\mathcal{H}(\mathbf{X}), h)$, with contraction factor $c_{\tilde{f}}$ and furthermore $c_f = c_{\tilde{f}}$.* □

Lemma 2.5 $h(A \cup B, C \cup D) \leq \max\{h(A, C), h(B, D)\} \forall A, B, C, D \in \mathcal{H}(\mathbf{X})$.□

Lemma 2.6 *Let $(\mathbf{X}, \mathbf{d})$ be a metric space and $\{w_1, w_2, \ldots, w_N\}$ be a finite set of contraction mappings on $(\mathcal{H}(\mathbf{X}), \mathbf{h})$, with contraction factor c_n for each w_n, respectively. Define $W : \mathcal{H}(\mathbf{X}) \to \mathcal{H}(\mathbf{X})$*

$$W(A) = \bigcup_{n=1}^{N} w_n(A) \qquad \forall A \in \mathcal{H}(\mathbf{X}). \tag{2.46}$$

Then W is a contraction mapping on $\mathcal{H}(\mathbf{X})$ with contraction factor

$$c = \max\{c_1, c_2, \ldots, c_N\} . \tag{2.47}$$

□

Putting these results together, the Contraction Mapping Principle may be restated for IFS.

Theorem 2.4 (CMP for IFS) *Let $\{\mathbf{X}; w_n, n = 1, 2, \ldots, N\}$ be an IFS with contraction factor c. Then the map $W : \mathcal{H}(\mathbf{X}) \to \mathcal{H}(\mathbf{X})$,*

$$W(B) = \bigcup_{n=1}^{N} w_n(B) \tag{2.48}$$

is a contraction mapping on the complete metric space $(\mathcal{H}(\mathbf{X}), \mathbf{h}(\mathbf{d}))$ with contraction factor c. The unique fixed point of W, $\bar{A}_W \in \mathcal{H}(\mathbf{X})$ is such that the following hold:

$$\bar{A}_W = W(\bar{A}_W) = \bigcup_{n=1}^{N} w_n(\bar{A}_W) \tag{2.49}$$

$$\bar{A}_W = \lim_{n\to\infty} W^{\circ(n)}(B) \ \forall B \in \mathcal{H}(\mathbf{X}). \tag{2.50}$$

□

Thus the IFS is a set of maps on a complete metric space $(\mathbf{X}, d)$, with an attractor (Equation 2.50) in $\mathcal{H}(\mathbf{X})$.

2.2.3 Algorithms

Definition 2.10 (affine transformation) *A mapping $S : \mathbf{X} \to \mathbf{X}$ is an affine transformation if*

$$S(x) = T(x) + b\ , \tag{2.51}$$

where $T : \mathbf{X} \to \mathbf{X}$ is a linear transformation. □

If $w_1, w_2, \ldots, w_N$ in Definition 2.8 are all affine transformations, then the unique fixed point of W given by Theorem 2.4 is termed a *self-affine set*. Many interesting fractal sets are of this type.

From the above theory two methods are evident for producing renditions of self-affine sets on a computer. Here is a brief outline of each. Similar descriptions are found in both [Fal90] and [Bar93].

Brute Force Calculation

Start with a set C in $\mathcal{H}(\mathbf{X})$. Apply each map to the set, to generate a first approximation, $W(C)$. Iterate ($W^{\circ(k)}(C)$ is the k^{th} iterate of W on C)

this process until it has converged[2], i.e. the computer representation is invariant under W. Algorithm 2.1 illustrates this process.

Algorithm 2.1 Brute force calculation of IFS

```
pick $C \in \mathcal{H}(\mathbf{X})$
while $W(C) \neq C$ do
    $C \Leftarrow W(C)$
end while
```

Random Iteration Algorithm

Let $\{\mathbf{X}; w_n, n = 1, 2, \ldots, N\}$ be an IFS. To each w_n, assign a probability $p_n > 0$, where $\sum_{n=1}^{N} p_n = 1$. For example let $p_n = \frac{c_n}{\sum_{k=1}^{N} c_k}$, so that more slowly converging maps are chosen more often. Construct a sequence $\{x_i\}_{i=1}^{I}$ in the following manner: Pick $x_0 \in \mathbf{X}$, then repeatedly choose randomly (under the probabilities p_n) a map w_i, and let $x_i = w_i(x_{i-1})$. Algorithm 2.2 illustrates this process.

Algorithm 2.2 Random iteration algorithm

```
pick $x_0 \in \mathbf{X}$
$I \Leftarrow$ number of iterations
for $i = 1$ to $I$ do
    pick random $n$ with probability $p_n$
    $x_i \Leftarrow w_n(x_{i-1})$
end for
```

Under the appropriate conditions [Bar93], the sequence $\{x_i\}_{i=1}^{I}$ converges to the attractor, $\bar{x}$, of the IFS. That is, $\lim_{i\to\infty} x_i = \bar{x}$. This process is often called the "Chaos Game". For large I , $\{x_i\}$ will appear to be randomly distributed across $\bar{x}$. This method can give a good rendering of the attractor, much faster than the previous method if individual calculations are expensive.

[2] To within some tolerance for the numerical representation used.

Example 2.2 (Sierpinski Triangle) The following maps [Bar93] define a IFS on $\mathbb{R}^2$, more precisely $[0,100]^2$, whose attractor is often called the *Sierpinski Triangle*.

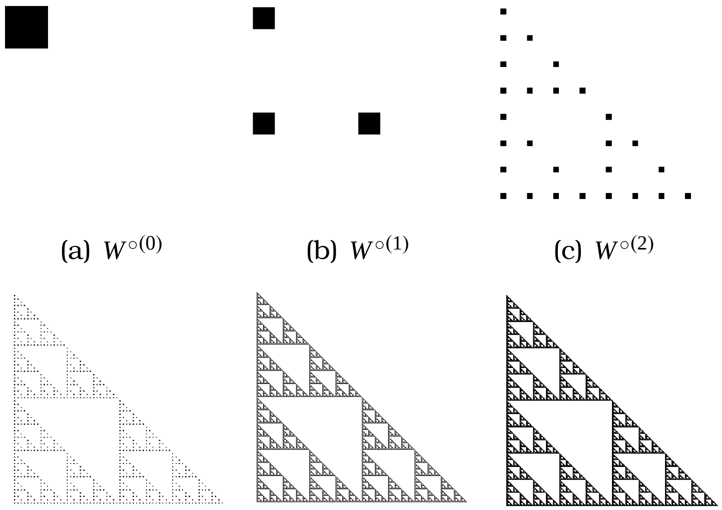
$$w_1\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}0.5&0\\0&0.5\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}+\begin{bmatrix}1\\1\end{bmatrix} \tag{2.52}$$

$$w_2\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}0.5&0\\0&0.5\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}+\begin{bmatrix}1\\50\end{bmatrix} \tag{2.53}$$

$$w_3\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}0.5&0\\0&0.5\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}+\begin{bmatrix}25\\50\end{bmatrix} \tag{2.54}$$

□

Figure 2.1 shows a progression of maps under $W(B)=\bigcup_{n=1}^{3}w_n(B)$, initially applied to the square $B=[0,20]^2$, where W^n is the n^{th} iteration of W on B. Figure 2.2 shows a similar progression of maps, this time starting with a 'box' of 100×100 with walls of width 10. This illustrates the application of $W(B)$ to different sets in $\mathbf{X}$ arriving at the attractor.



(a) $W^{\circ(0)}$ (b) $W^{\circ(1)}$ (c) $W^{\circ(2)}$

(d) $W^{\circ(3)}$ (e) $W^{\circ(6)}$ (f) $W^{\circ(15)}$

Figure 2.1: Sierpinski triangle from $[0,20]^2$

Figure 2.3 is a detail of the 'Sierpinski Triangle', illustrating the self-similarity of this set.

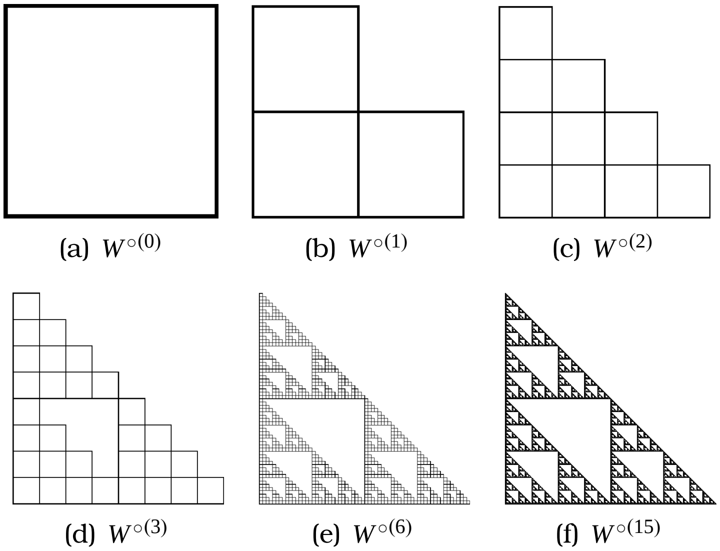

(a) $W^{\circ(0)}$ (b) $W^{\circ(1)}$ (c) $W^{\circ(2)}$ (d) $W^{\circ(3)}$ (e) $W^{\circ(6)}$ (f) $W^{\circ(15)}$

Figure 2.2: Sierpinski triangle from box (walls width 10)

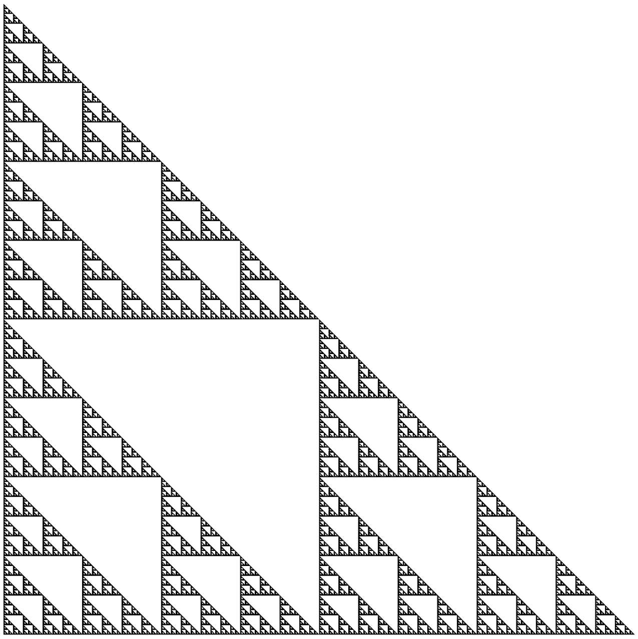


Figure 2.3: Sierpinski triangle

Example 2.3 (Fern) The following maps [Bar93] define a IFS with an attractor that looks very much like a fern.

$$w_1\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}0&0\\0&0.16\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}+\begin{bmatrix}0\\0\end{bmatrix} \tag{2.55}$$

$$w_2\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}0.85&0.04\\-0.04&0.85\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}+\begin{bmatrix}0\\1.6\end{bmatrix} \tag{2.56}$$

$$w_3\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}0.2&-0.26\\0.23&0.22\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}+\begin{bmatrix}0\\1.6\end{bmatrix} \tag{2.57}$$

$$w_4\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}-0.15&0.28\\0.26&0.24\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}+\begin{bmatrix}0\\0.44\end{bmatrix} \tag{2.58}$$

□

Figure 2.4 is a detail of this ’fern’ IFS.

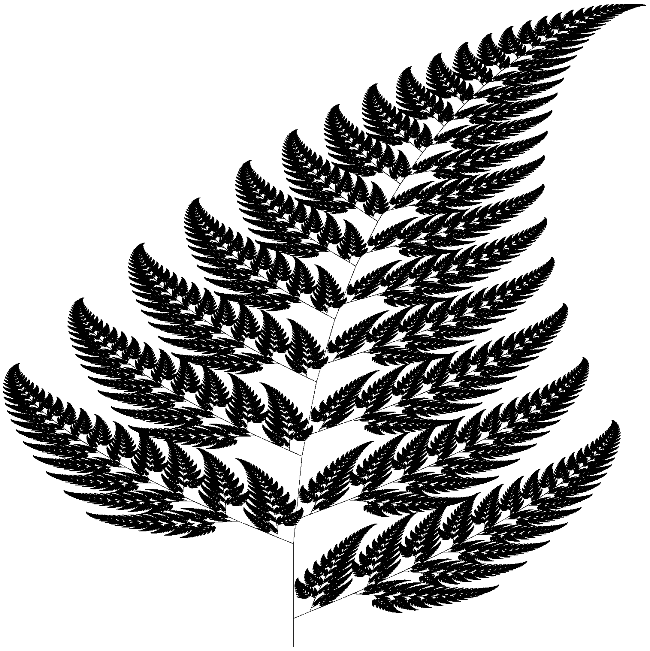


Figure 2.4: Fern IFS

2.2.4 An Inverse Problem for IFS

There has been much interest in the subject of fractals as models for natural objects [Man77], driven by the compelling likenesses such as that shown in Example 2.3. It is therefore natural to consider an inverse problem: given a set A in a complete metric space, (for example, a compact subset of $\mathbb{R}^2$), is it possible to find a function f whose attractor is A? The following states this more carefully.

Inverse Problem

> Let $(\mathbf{X}, d)$ be a complete metric space, and let $x \in \mathbf{X}$. Let $\epsilon > 0$. Does a contractive $f : \mathbf{X} \to \mathbf{X}$ exist such that $d(x, \bar{x}_f) < \epsilon$?

This can be approached as an optimisation problem; given a space of parameters $\mathcal{P}$ defining contraction mappings $f \in \text{Con}(\mathbf{X})$, search for the optimal parameters. That is, find $p \in \mathcal{P}$ such that the induced $f_p \in \text{Con}(\mathbf{X})$ has the property $d(x, \bar{x}_{f_p}) \leq d(x, \bar{x}_{f_q})\ \forall q \in \mathcal{P}$. For the sake of generality, assume that this parameter space is rich enough that the optimal fixed point $\bar{x}_{f_p}$ is acceptably close to x. Finding p, however, can be a tedious process.

The following theorem, a consequence of Banach's contraction mapping principle, offers a more attractive approach. Instead of measuring the distance $d(x, \bar{x}_f)$, the collage distance, $d(x, f(x))$ can be utilised to simplify the problem [BEHL86].

Theorem 2.5 (Collage Theorem) *Let $\{\mathbf{X}; w_n, n = 1, 2, \ldots, N\}$ be an IFS with contractivity factor c (recall $0 \leq c < 1$). Let $\epsilon > 0$ and suppose there exists $A \in \mathcal{H}(\mathbf{X})$ such that $h(A, W(A)) \leq \epsilon$. Then the attractor (fixed point) of W, $\bar{A}_W$ is such that*

$$h(A, \bar{A}_W) \leq \frac{\epsilon}{1 - c} \tag{2.59}$$

or, equivalently,

$$h(A, \bar{A}_W) \leq \frac{1}{1-c} h(A, W(A)) \qquad \forall A \in \mathcal{H}(\mathbf{X}). \tag{2.60}$$

□

PROOF

$$\begin{aligned} h(A, \bar{A}_W) &\leq h(A, W(A)) + h(W(A), \bar{A}_W) & (2.61)\\ &= h(A, W(A)) + h(W(A), W(\bar{A}_W)) & (2.62)\\ &\leq h(A, W(A)) + ch(A, \bar{A}_W). & (2.63) \end{aligned}$$

Therefore,

$$h(A, \bar{A}_W) \leq \frac{1}{1-c} h(A, W(A)). \tag{2.64}$$

■

Corollary 2.2 (Anti-Collage Theorem) *Under the same conditions,*

$$h(A, \bar{A}_W) \geq \frac{1}{1+c} h(A, W(A)) \qquad \forall A \in \mathcal{H}(\mathbf{X}), . \tag{2.65}$$

□

Herein lies the key to a method of attacking the inverse problem [VS99]. Equation 2.64 shows that by finding maps W such that A is mapped close to itself, the attractor $\bar{A}_W$ is forced to be close (in the above sense) to A. Instead of tackling the inverse problem directly, we may consider a related problem.

Re-formulation of the inverse problem

> Let $(\mathbf{X}, d)$ be a complete metric space, and let $x \in \mathbf{X}$. Let $\epsilon > 0$. Does a contractive $f : \mathbf{X} \to \mathbf{X}$ exist such that $d(x, f(x)) < \epsilon$?

The above problem is much more computationally tractable, and forms the basis of most, if not all, fractal compression methods. It is important to note that while Theorem 2.5 points the way toward fractal-based approximation methods, the methods also require Theorem 2.1 (continuity of fixed points with respect to contraction maps), as such methods work by varying the map W.

Clearly, the Collage Theorem need not be stated in terms of IFS under the Hausdorff metric; it may be stated for any complete metric space and contractive function on that space. Together, the Collage Theorem and Anti-Collage theorem give upper and lower bounds on the approximation error $h(A, \bar{A}_W)$, thus a non-zero collage error ensures a positive approximation error. In general, optimising the collage distance does not result in optimal maps. A detailed discussion of inverse problems for IFS may be found in [Vrs96], [FV98a], where a generalised fractal transform is discussed.

2.3 IFS on Grey Level Maps

The IFS demonstrated on the Hausdorff metric have fixed points that are sets. Hence only binary 'pixels' may be represented this way; a 'pixel' is either in the set or it is not. Such images are called *bitmaps*. However, we wish to address a more general class of images; we are typically dealing with grey levels[3], not bitmaps. In this case in/exclusion in a fixed point set contained in of $\mathcal{H}(\mathbf{X})$ is not applicable.

Iterated Function Systems on Grey Level Maps (IFSM), introduced by Forte and Vrscay [FV98b], was developed to provide a way around this problem. Initial forays into IFS on images were made by Barnsley and Demko [BD85, BEHL86], and others. The IFSM is a nice theoretical framework to handle images.

This section briefly describes the application of IFS method over an appropriate function space $\mathcal{F}(\mathbf{X})$. Typically, signals are represented in $\mathcal{L}^p(\mathbb{R})$ (in practice, the space of functions is often $\mathcal{L}^2(\mathbb{R})$). A detailed discussion of IFSM may be found in [FV98a], [Vrs96].

[3] Or colour, as will be addressed later.

Definition 2.11 (IFSM) *We define an (N-map) Iterated Function System on Grey Level Maps to consist of a complete metric space* $(\mathbf{X}, d)$ *and two components*

1. ***IFS component**:* $\mathbf{w} = \{w_1, w_2, \ldots, w_N\}$ *where each* $w_n : X \to X$ *is a contraction with contraction factor* c_n*. It is convenient, although not strictly necessary, to add the constraint that* $\cup_n w_n(x) = \mathbf{X}$*, thus ensuring that for all* $x \in \mathbf{X}$*,* $w_n^{-1}(x)$ *exists for some* n *.*

2. ***grey level component**:* $\Phi = \{\phi_1, \phi_2, \ldots, \phi_N\}$ *where each* $\phi_n : \mathbb{R} \to \mathbb{R}$ *is Lipschitz (i.e.* $\exists s_i \geq 0$ *such that* $|\phi_i(x) - \phi_i(y)| \leq s_i|x - y| \; \forall \; x, y \in \mathbb{R}$*).*

The IFSM may be denoted $\{\mathbf{X}; w_n, \phi_n, n = 1, 2, \ldots, N\}$*.* □

With the IFSM $\{\mathbf{X}; w_n, \phi_n, n = 1, 2, \ldots, N\}$, associate an operator performing the "fractal transform":

Definition 2.12 *Define an operator* $T : \mathcal{F}(\mathbf{X}) \to \mathcal{F}(\mathbf{X})$ *in the following manner. For each* $x \in X$ *define its* N *fractal components as*

$$g_i(x) = \begin{cases} \phi_i(f(w_i^{-1}(x))), & x \in w_i(\mathbf{X}) \\ 0, & x \notin w_i(\mathbf{X}). \end{cases} \tag{2.66}$$

And given an $f \in \mathcal{F}(\mathbf{X})$*, the image of* f *under* T *is*

$$(Tf)(x) = \sum_{i=1}^{N} g_i(x). \tag{2.67}$$

□

Thus the i^{th} "fractal component", $g_i(x)$, scales the grey-level value of f in the preimage $w_i^{-1}(x)$ if said preimage exists. See, for example, (3.1).

An interesting special case of this IFSM operator is arrived at by adding, in the role of a "condensation" function, some map $\theta(x) : \mathbf{X} \to \mathbf{X}$, as

$$(Tf)(x) = \theta(x) + \sum_{i=1}^{N} g_i(x). \tag{2.68}$$

In applications, affine IFS maps, and affine grey level maps (see below) are often used. The Hausdorf metric becomes cumbersome to use, and images are often modelled in $\mathcal{L}^p$ spaces, especially $\mathcal{L}^2(\mathbb{R})$. Under certain conditions, the IFSM is contractive.

Theorem 2.6 *Let* $\mathbf{X} \subset \mathbb{R}^D$*. Let* $\{\mathbf{X}; w_n, \phi_n, n = 1, 2, \dots, N\}$ *be an* N*-map IFSM. Let* $f, g \in \mathcal{L}^p(\mathbf{X})$ *for some* $p > 0$*. Then*

$$d_p(Tf, Tg) \leq \sum_{n=1}^{N} c_n^{D/p} s_n d_p(f, g). \tag{2.69}$$

□

PROOF

$$\begin{aligned}
d_p(Tf, Tg) &= \|Tf - Tg\|_p && (2.70)\\
&= \left[\int_{\mathbf{X}} \left| \sum_{n=1}^{N} \phi_n(f(w_n^{-1}(x))) - \phi_n(g(w_n^{-1}(x))) \right|^p dx \right]^{\frac{1}{p}} && (2.71)\\
&\leq \sum_{n=1}^{N} \left[\int_{\mathbf{X}_n} \left| \phi_n(f(w_n^{-1}(x))) - \phi_n(g(w_n^{-1}(x))) \right|^p dx \right]^{\frac{1}{p}} && (2.72)\\
&\quad \text{where } \mathbf{X}_n = w_n(X) && (2.73)\\
&\leq \sum_{n=1}^{N} \left[c_n^D \int_{\mathbf{X}} \left| \phi_n(f(\xi)) - \phi_n(g(\xi)) \right|^p d\xi \right]^{\frac{1}{p}} && (2.74)\\
&\leq \sum_{n=1}^{N} c_n^{D/p} s_n \left[\int_{\mathbf{X}} \left| \phi_n(\xi) - \phi_n(\xi) \right|^p d\xi \right]^{\frac{1}{p}} && (2.75)\\
&= \sum_{n=1}^{N} c_n^{D/p} s_n d_p(f, g). && (2.76)
\end{aligned}$$

■

Thus the IFSM operator is contractive if $\sum_{i=1}^{N} c_i^{D/p} s_i < 1$. In principle, implementations of this method should check this contractivity factor; in practice it is rarely done.

The inverse problem for IFSM

The inverse problem can again be posed,

> Let $f \in \mathcal{L}^p(\mathbb{R})$ and $\epsilon > 0$. Does a contractive IFSM with fixed point $\bar{f}$ exist such that $d(f, \bar{f}) < \epsilon$?

As was the case with IFS, the key technique is the collage theorem. Again, we re-formulate:

> Let $f \in \mathcal{L}^p(\mathbb{R})$ and $\epsilon > 0$. Does a contractive IFSM map T exist such that $d(f, Tf) < \epsilon$?

Instead of searching for IFSM that converge close to f, search for IFSM with associated operators T such that $d(f, Tf) < \epsilon$.

Affine grey level maps

In application to image compression, it will be convenient to consider for the grey level ϕ a family of affine maps,

$$\phi_n(x) = \alpha_n x + \beta_n. \tag{2.77}$$

Use of this family of maps greatly simplifies application of the collage theorem. Consider a function f on $\mathcal{L}^p(\mathbb{R})$; we wish to find maps ϕ_n that minimise the collage distance Δ,

$$\begin{aligned}\Delta^2 &= \| f(x) - Tf(x)\|_p^2 = \langle f(x) - Tf(x),\, f(x) - Tf(x)\rangle && (2.78)\\ &= \langle f(x) - \sum_{n=1}^{N} \phi_n(f(w_n^{-1}(x))),\, f(x) - \sum_{n=1}^{N} \phi_n(f(w_n^{-1}(x)))\rangle. && (2.79)\end{aligned}$$

Make the simplifying assumption that the IFS maps 'w_n' are strictly non-overlapping. Denoting $\xi = w_n^{-1}(x)$ as before, each of the N components of Δ may be minimised separately,

$$\begin{aligned}\Delta_n^2 &= \langle f(x) - \alpha_n f(\xi) - \beta_n,\, f(x) - \alpha_n f(\xi) - \beta_n\rangle && (2.80)\\ &= \langle f(x),\, f(x)\rangle + \alpha_n^2 \langle f(\xi),\, f(\xi)\rangle + \beta_n^2 \langle 1, 1\rangle && (2.81)\\ &\quad + 2\alpha_n\beta_n \langle f(\xi), 1\rangle - 2\alpha_n \langle f(\xi),\, f(x)\rangle - 2\beta_n \langle f(x), 1\rangle. && (2.82)\end{aligned}$$

Applying the stationarity conditions $\frac{\partial \Delta_n^2}{\partial \alpha_n} = 0$, and $\frac{\partial \Delta_n^2}{\partial \beta_n} = 0$ results in a linear system that may be solved for α_n, β_n as

$$\alpha_n = \frac{\langle f(\xi), 1 \rangle \langle f(x), 1 \rangle - N \langle f(\xi), f(x) \rangle}{\langle f(\xi), 1 \rangle^2 - N \langle f(\xi), f(\xi) \rangle} \tag{2.83}$$

$$N\beta_n = \langle f(\xi), 1 \rangle - \alpha \langle f(x_i), 1 \rangle \ , \tag{2.84}$$

Where $N = \langle 1, 1 \rangle$ will, in the case of interest, be the number of pixels in the range block. Note that there is potential for ambiguity here, as we may treat the image as a continuous surface or as a discrete set of measurements on a lattice, for example. While the preceding discussion has remained general, in the applications to follow, images will be treated as discrete. In this case (with ℓ^2 norm), we are preforming a linear least squares fit. Note that if $f(x) = k$ is constant, the determinant of the above system is zero. In this case, the system may be solved by setting $\alpha_n = 0$, and $\beta_n = \langle f(x), 1 \rangle = k$.

For reasons discussed in chapter 3, it will sometimes be convenient to consider the case of the associated zero-mean domain and range blocks. If we denote the zero-mean versions with $\tilde{\cdot}$, that is

$$\tilde{f}(x) = f(x) - \bar{f}(x) \ , \tag{2.85}$$

where $\bar{\cdot}$ denotes the mean value.

So now, similarly to above, we can seek the $\mathcal{L}^2(\mathbb{R})$ fit these zero-mean parent-child mappings under affine maps. Approximating the range blocks now by $\tilde{f}(x) \approx a\tilde{f}(\xi) + b$ (supported on the particular range/domain blocks). For the n^{th} component, then we minimise

$$\delta_n^2 = \langle \tilde{f}(x) - a_n \tilde{f}(\xi) - b_n, \tilde{f}(x) - a_n \tilde{f}(\xi) - b_n \rangle \ . \tag{2.86}$$

Imposing the same stationarity conditions with respect to a and b

yields

$$a_n = \frac{\langle \tilde{f}(x),\, \tilde{f}(\xi)\rangle}{\langle \tilde{f}(x),\, \tilde{f}(x)\rangle} \tag{2.87}$$

$$b_n = 0\ . \tag{2.88}$$

Expanding in terms of (2.85) and noting that $\bar{f}(x) = \frac{\langle f(x),1\rangle}{N}$ will yield the previous result. Rescaling back to the original (non zero-mean) versions of $f(x)$ and $f(\xi)$, we find that

$$\alpha_n = a_n, \quad \beta_n = \bar{f}(\xi) - a_n \bar{f}(x)\ . \tag{2.89}$$

We also note that $\Delta^2 = \delta^2$. Furthermore, the variances are related as

$$\mathrm{var}(\tilde{f}(x)) = \langle \tilde{f}(x),\, \tilde{f}(x)\rangle = \mathrm{var}(f(x))\ , \text{ and } \quad \mathrm{var}(\tilde{f}(\xi)) = \mathrm{var}(f(\xi))\ . \tag{2.90}$$

As shown first by Bedford *et al.* [BDK92], if we impose the condition $\delta^2 \le \epsilon$ for some $\epsilon > 0$, then

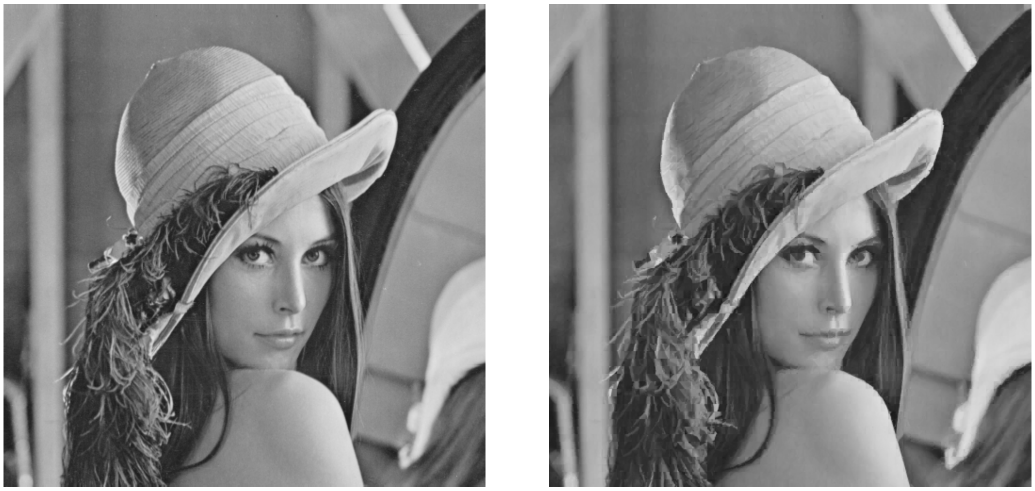
$$\mathrm{var}(f(x)) \ \le\ a^2\,\mathrm{var}(f(\xi)) + \epsilon. \tag{2.91}$$

This last result, contrary to the assumptions of some ‘variance based pruning’ that has been used to accelerate fractal compression, shows that even in the case of high variance regions a good fit is possible with sufficiently small $|a|$.

Practical application of the inverse problem

Implicit in this approach is the assumption that we have defined a rich enough parameter space (in the affine maps) to guarantee the construction of an operator whose fixed point is an acceptable estimate of the original image. While obviously ‘acceptable’ may be problem-dependent, in practice this has never been a problem. Figure 2.5 shows an example.

With these results, we have nearly all of the theoretical framework



(a) Lena

(b) fixed point

Figure 2.5: a) Original b) Fixed point $\bar{u}$ of the IFSM operator T. Error between these images is a PSNR of 31.56

to support the discussion of chapter 3. A brief discussion of wavelets is needed, however.

2.4 IFS in the Wavelet Domain

2.4.1 Wavelets

In this section a brief introduction to the area of wavelets is provided. By necessity this treatment must be incomplete, and is rather narrowly aimed toward a useful representation for digital signal processing and, more particularly for our aims, one that will allow a hybrid fractal-wavelet approach.

Wavelets are an area that has attracted a large amount of research in the last twenty years or so (see [Mal98, Dau92] for overview). In this work, I am not particularly focused on new research on wavelet or wavelet-like bases, but rather on the useful properties of such bases in the representation of scale in images. A detailed treatment of this background may be found in several sources, for example

[Dau92, Mal98].

Haar Wavelets

Much of the background for what is now called *wavelets* was laid by Haar in 1910. He demonstrated [Haa10] the simple piecewise constant function could be used to generate an orthonormal basis of $\mathcal{L}^2(\mathbb{R})$. As a motivating example, first examine this basis. Consider the function on $\mathbb{R}$:

$$\psi(t) = \begin{cases} 1 & 0 \le t < \frac{1}{2}, \\ -1 & \frac{1}{2} \le t < 1, \\ 0 & elsewhere. \end{cases} \tag{2.92}$$

The function ψ is often known as a *mother wavelet*, as it is used to generate a family of wavelets: Take the following translations and dilations of equation 2.92,

$$\psi_{j,k}(x) = 2^{-j/2}\,\psi(2^{-j}\,x - k)\ , \tag{2.93}$$

and note that $\psi_{0,0}(t) = \psi(t)$.

Proposition 2.4 $\{\psi_{j,k}(t)\}_{(j,k)\in\mathbb{Z}^2}$ *forms an orthonormal basis (called the Haar basis) for* $\mathcal{L}^2(\mathbb{R})$. □

PROOF proof for a more general case will be given later. ■

Continuous Wavelets

To begin, some definitions:

Definition 2.13 (Wavelet) *A continuous wavelet is a function* $\psi \in \mathcal{L}^2(\mathbb{R})$ *with zero average, that is,*

$$\int \psi(x)\,dx = 0. \tag{2.94}$$

From ψ, a family of translations and dilations is generated as follows,

$$\psi_{a,b}(x) = |a|^{-1/2}\psi\left(\frac{x-b}{a}\right), \quad a, b \in \mathbb{R}, \ a \neq 0. \tag{2.95}$$

The wavelet function is normalised, which gives $\|\psi\| = \|\psi_{a,b}\| = 1$. It must also satisfy an admissibility condition,

$$2\pi \int |\xi|^{-1} |\hat{\psi}(\xi)|^2 \, d\xi < \infty. \tag{2.96}$$

□

Where $\hat{\psi}(\xi)$ denotes the Fourier transform of ψ.

Definition 2.14 (Continous Wavelet Transform) *For any $f \in \mathcal{L}^2(\mathbb{R})$, the wavelet transform of f is defined as*

$$(Wf)(a,b) = \langle f, \psi_{a,b} \rangle \tag{2.97}$$

$$= \int f(x)|a|^{-1/2}\overline{\psi\left(\frac{x-b}{a}\right)}\, dx. \tag{2.98}$$

□

The Continuous Wavelet Transform (CWT) describes the mapping of f onto a family defined by the mother wavelet ψ. It is natural to ask what conditions suffice in order to be able to recover f from the coefficients $\langle f, \psi_{a,b} \rangle$.

Theorem 2.7 (Reproducibility of f) *Let $\psi \in \mathcal{L}^2(\mathbb{R})$ be a real function satisfying*

$$C_\psi = 2\pi \int |\xi|^{-1} |\hat{\psi}(\xi)|^2 \, d\xi < \infty. \tag{2.99}$$

Then for all $f \in \mathcal{L}^2(\mathbb{R})$

$$f(t) = \frac{1}{C_\psi} \int\int (Wf)(a,b)\, |a|^{-1} \psi\left(\frac{t-b}{a}\right) \frac{da\, db}{a^2}. \tag{2.100}$$

□

PROOF See (Dau92, Mal98). ■

This key result for the continuous transform describes the conditions under which f may be recoverably decomposed into projections on the spaces defined by the family $\psi_{a,b}$ (i.e., Equation 2.95).

In applications, discrete signals (e.g. a continuous signal that is sampled at a uniform interval) are often important. Although it is possible to consider discrete signals as quantisations of continuous signals, and thus operate in the 'original' continuous space, it is not always beneficial to do this. Many image operations on discrete pixels will be interesting, and so a discrete version of the wavelet transform is very useful.

Discrete Wavelets

The wavelet family described in Equation 2.95 may be discretized by fixing particular $a_0, b_0 \geq 1$ in place of a, b, and scaling them by $n, m \in \mathbb{Z}$. The resulting family of functions will be

$$\psi_{a,b}(x) = a_0^{-m/2} \psi\left(\frac{x - nb_0 a_0^m}{a_0^m}\right) \tag{2.101}$$

$$= a_0^{-m/2} \psi(a_0^{-m} x - nb_0). \tag{2.102}$$

Having discretized the family of translations and dilations of ψ, it remains to determine if a "discrete admissibility condition" (i.e. a discrete version of Equation 2.96) exists. Additionally, a discrete version of Theorem 2.7 would be useful. Under certain constraints on a_0, b_0 and ψ, there exists a function $\tilde{\psi}_{m,n}$ such that for all $f \in \mathcal{L}^2(\mathbb{R})$,

$$f = \sum_{m,n} \langle f, \tilde{\psi_{m,n}} \rangle \tilde{\psi}_{m,n} \, . \tag{2.103}$$

For details of this result, see [Dau92]. In application to digital signals, it is especially convenient to choose the values $a_0 = 2$, $b_0 = 1$. This special case will be the only one considered in the remainder of this development.

$$\psi_{j,k}(x) = 2^{-j/2} \psi(2^{-j} x - k). \tag{2.104}$$

Multiresolution analysis

The Multiresolution Analysis (MRA) was introduced by Mallat[Mal89a, Mal89b] as infrastructure for a computationally practical approach to discrete wavelets. An important aspect of the MRA will be the so-called *scaling function*, ϕ. Similar to the wavelet ψ, the translations and dilations of the scaling function are interesting. These form the family $\{\phi_{j,k}\}$ where

$$\phi_{j,k}(x) = 2^{-j/2}\phi(2^{-j}x - k). \tag{2.105}$$

Definition 2.15 (MRA) *Any sequence $\{V_j\}_{j\in\mathbb{Z}}$ of closed subspaces of $\mathcal{L}^2(\mathbb{R})$ is called an MRA if it has the following properties:*

$$\forall j \in \mathbb{Z} \qquad V_{j+1} \subset V_j \tag{2.106}$$

$$\lim_{j\to+\infty} V_j = \bigcap_{j=-\infty}^{+\infty} V_j = \{0\} \tag{2.107}$$

$$\lim_{j\to-\infty} V_j = \text{Closure}\left(\bigcup_{j=-\infty}^{+\infty} V_j\right) = \mathcal{L}^2(\mathbb{R}) \tag{2.108}$$

$$f \in V_j \Leftrightarrow f(2^j\cdot) \in V_0 \tag{2.109}$$

$$f \in V_0 \Rightarrow f(\cdot - j) \in V_0 \;\forall j \in \mathbb{Z} \tag{2.110}$$

$$\exists\, \phi \in V_0 \quad \textit{such that} \quad \{\phi_{0,n} : n \in \mathbb{Z}\} \quad \textit{is an orthonormal basis of} \;\; V_0\,. \tag{2.111}$$

□

From Equation 2.105 we have the family $\{\phi_{j,k}\}$. Together, the Properties 2.111 and 2.109 imply [Dau92] that for each $j \in \mathbb{Z}$, the set $\{\phi_{j,n}\}_{n\in\mathbb{Z}}$ forms an orthonormal basis for V_j. As $\phi \in V_0 \subset V_{-1}$, and $\{\phi_{-1,n}\}_{n\in\mathbb{Z}}$ is an orthonormal basis for V_{-1}, ϕ may be written as

$$\phi = \sum_n \langle \phi, \phi_{-1,n}\rangle \phi_{-1,n}\,. \tag{2.112}$$

Let

$$h_n = \langle \phi, \phi_{-1,n} \rangle, \tag{2.113}$$

then the fundamental scaling result for the scaling function $\phi(x)$ is

$$\phi(x) = \sqrt{2} \sum_n h_n \phi(2x - n). \tag{2.114}$$

The MRA will not define a wavelet ψ; rather, given an MRA it will be possible to (non-uniquely) construct a wavelet.

Theorem 2.8 *Let $\{V_n\}_{n \in \mathbb{Z}}$ be an MRA. Then there exists an associated orthonormal wavelet basis $\{\psi_{j,k}\}_{(j,k)\in\mathbb{Z}^2}$ such that*

$$\psi(x) = \sqrt{2} \sum_n (-1)^{n-1} h_{-n-1} \phi(2x - n) \tag{2.115}$$

$$P_{V_{j-1}} = P_{V_j} + \sum_k < \cdot, \psi_{j,k} > \psi_{j,k} \tag{2.116}$$

where h_n is defined as in 2.113, and $\psi_{j,k}$ as defined in Equation 2.105. Here P_X denotes the a projection operator, i.e. the projection onto the space X. □

PROOF see (Dau92). ■

It is important to note that this does not uniquely determine ψ. In particular we may choose (for future simplicity)

$$\psi = \sum_n g_n \phi_{-1,n} \qquad \text{where } g_n = (-1)^n h_{-n+1}, \tag{2.117}$$

yielding

$$\psi(x) = \sqrt{2} \sum_n (-1)^n h_{-n+1} \phi(2x - n). \tag{2.118}$$

Mallat Algorithm

From Property 2.106, V_{j+1} is a proper subset of V_j. Hence, there is an orthogonal complement of V_{j+1} in V_j, call it W_j. Thus $W_j \perp V_j$ and

$$V_j = V_{j+1} \oplus W_{j+1}\,. \tag{2.119}$$

Thus if Equation 2.119 is recursively applied

$$V_j = V_N \oplus \bigoplus_{k=0}^{N-j-1} W_{N-k}\ \text{, for } j < N \tag{2.120}$$

Within this cascade of spaces, it can be shown that [Mal98, Dau92]

$$\langle f, \psi_{j,k} \rangle = \langle f, \sum_n g_{n-2k} \phi_{j-1,n} \rangle = \sum_n \overline{g_{n-2k}} \langle f, \phi_{j-1,n} \rangle. \tag{2.121}$$

and

$$\langle f, \phi_{j,k} \rangle = \sum_n \overline{h_{n-2k}} \langle f, \phi_{j-1,n} \rangle. \tag{2.122}$$

Where $\bar{\cdot}$ denotes complex conjugation. Reconstruction of the signal can be achieved as

$$\begin{aligned} \langle f, \phi_{j-1,k} \rangle &= \sum_n \overline{(\overline{h_{k-2n}})} \langle f, \phi_{j,n} \rangle + \sum_n \overline{(\overline{g_{k-2n}})} \langle f, \psi_{j,n} \rangle \\ &= \sum_n h_{k-2n} \langle f, \phi_{j,n} \rangle + \sum_n g_{k-2n} \langle f, \psi_{j,n} \rangle. \end{aligned} \tag{2.123}$$

If we adopt an 'operator notation' for these sums, Figure 2.6 illustrates the decomposition and recombination steps. This process is referred to in signal processing theory as *subband filtering*.

Example 2.4 (Haar Basis) For convenience, introduce the following

$$\begin{tikzcd}
a_0 \arrow[r, "\bar{H}"] \arrow[dr, "\bar{G}"] & a_1 \arrow[r, "\bar{H}"] \arrow[dr, "\bar{G}"] & a_2 \arrow[r, "\bar{H}"] \arrow[dr, "\bar{G}"] & a_3 \arrow[r, dashed, no head] \arrow[dr, dashed, no head] & {} \\
{} & d_1 & d_2 & d_3 & {}
\end{tikzcd}$$

(a) decomposition

$$\begin{tikzcd}
a_0 & a_1 \arrow[l, "H"'] & a_2 \arrow[l, "H"'] & a_3 \arrow[l, "H"'] \arrow[r, dashed, no head] \arrow[dr, dashed, no head] & {} \\
{} & d_1 \arrow[ul, "G"] & d_2 \arrow[ul, "G"] & d_3 \arrow[ul, "G"] & {}
\end{tikzcd}$$

(b) recombination

Figure 2.6: Mallat algorithm: (a) the decomposition of Equation 2.121 (b) the recombination of Equation 2.123

interval notation

$$I_{[a,b]}(x) = \begin{cases} 1 & a \leq x \leq b, \\ 0 & \text{otherwise.} \end{cases}$$

Beginning with the scaling function $\phi(x) = I_{[0,1]}(x)$, the filter h is easily determined, since (Equation 2.113)

$$h[n] = \langle \phi(x), 2^{1/2}\phi(2x-n) \rangle.$$

This implies that

$$h[n] = \begin{cases} 2^{-1/2} & \text{for} \qquad n = 0, 1 \\ 0 & \text{otherwise.} \end{cases}$$

Now applying Equation 2.117 yields

$$\begin{aligned}2^{-1/2}\psi(x) &= \sum_n (-1)^n h[-n+1]\phi(2x-n)\\ &= (-1)^1 h[0]\phi(2x-1) + (-1)^0 h[1]\phi(2x)\\ &= 2^{-1/2}\phi(2x) - 2^{-1/2}\phi(2x-1).\end{aligned}$$

Therefore

$$\begin{aligned}\psi(x) &= \phi(2x) - \phi(2x + -1)\\ &= I_{[0,1/2]} - I_{[1/2,0]}\,.\end{aligned}$$

Which is the Haar wavelet given in Equation 2.92. □

Beyond Orthonormal Wavelets?

The approach described above sketches the fundamentals of Discrete Wavelet Transform (DWT) theory. It constitutes a brief introduction to orthogonal wavelet bases. Before continuing, it is worth noting that orthogonal wavelets are hardly the end of the story here. The extension to biorthogonal wavelets, more general frames, or indeed the many wavelet-like approaches developed recently is beyond the scope of this thesis.

Wavelets and images

In order to apply the wavelet approach to images, an extension from 1-d to 2-d is needed. The previous discussion has concentrated on bases for $\mathcal{L}^2(\mathbb{R})$. A method to extend these results to two dimensions is now discussed. In a completely analogous manner, extensions to n dimensions may be made.

Given $\{\psi_{j,k}(t)\}_{(j,k)\in\mathbb{Z}^2}$, an orthonormal basis for $\mathcal{L}^2(\mathbb{R})$, the goal is to create an orthonormal basis for $\mathcal{L}^2(\mathbb{R}^2)$. While straightforward, the extension $\{\psi_{j,k}(x)\psi_{l,m}(y)\}_{(j,k,l,m)\in\mathbb{Z}^4}$q has the inconvenient property of mixing x and y information at different resolutions. In order to avoid this, we construct higher dimension multiresolutions.

Separable Multiresolutions

Suppose $\{V_n\}_{n\in\mathbb{Z}}$ is a MRA of $\mathcal{L}^2(\mathbb{R})$. Consider the tensor product spaces,

$$\mathbf{V}_n^2 = V_n \otimes V_n = \text{span}\,\overline{\{F(x,y) = f(x)g(y) : f, g \in V_n\}}. \tag{2.124}$$

With the appropriate modification to the MRA (Definition 2.15), the $\{\mathbf{V}_n\}_{n\in\mathbb{Z}}$ form a multiresolution in $\mathcal{L}^2(\mathbb{R}^2)$ [Mal98, Dau92]. Essentially following the discussion of §2.4.1, denote the orthogonal complement of $\mathbf{V}_j$ as $\mathbf{W}_j$. Then (dropping the superscript 2 which would appear on all vector quantities)

$$\mathbf{V}_j = \mathbf{V}_{j+1} \oplus \mathbf{W}_{j+1} \tag{2.125}$$

where (using W_j to denote the orthogonal complement of V_j, as before)

$$\mathbf{W}_j = (V_j \otimes W_j) \oplus (W_j \otimes V_j) \oplus (W_j \otimes W_j). \tag{2.126}$$

For simplicity, we denote this as

$$\mathbf{W}_j = \mathbf{W}_j^h \oplus \mathbf{W}_j^v \oplus \mathbf{W}_j^d, \tag{2.127}$$

where the superscripts h, v, and d are for horizontal, vertical, and diagonal respectively. The separable multiresolution results in a family of wavelet functions [Mal98]

$$\Psi^h_{n,j,k}(x,y) = \phi_{n,j}(x)\psi_{n,k}(y) \tag{2.128}$$
$$\Psi^v_{n,j,k}(x,y) = \psi_{n,j}(x)\phi_{n,k}(y) \tag{2.129}$$
$$\Psi^d_{n,j,k}(x,y) = \psi_{n,j}(x)\psi_{n,k}(y) \tag{2.130}$$

and the scaling function,

$$\Phi_{j,k}(x,y) = \phi_{j,k}(x)\phi_{j,k}(y). \tag{2.131}$$

Analogous to §2.4.1, consider the *complete decomposition*, arrived at by repeated application of 2.125

$$\mathbf{V}_j = \mathbf{V}_N \oplus \bigoplus_{k=0}^{N-j-1} \mathbf{W}_{N-k}\,, \text{ for } j < N \tag{2.132}$$

$$= \mathbf{V}_N \oplus \bigoplus_{k=0}^{N-j-1} \left(\mathbf{W}^h_{N-k} \oplus \mathbf{W}^v_{N-k} \oplus \mathbf{W}^d_{N-k}\right), \text{ for } j < N \tag{2.133}$$

where, in terms of the above wavelet functions (overbar denoting closure),

$$\mathbf{V}_n = \text{span}\,\overline{\{\Phi_{n,j,k}(x,y) : 0 \le j,k \le 2^n - 1\}} \tag{2.134}$$

$$\mathbf{W}^h_n = \text{span}\,\overline{\{\Psi^h_{n,j,k}(x,y) : 0 \le j,k \le 2^n - 1\}} \tag{2.135}$$

$$\mathbf{W}^v_n = \text{span}\,\overline{\{\Psi^v_{n,j,k}(x,y) : 0 \le j,k \le 2^n - 1\}} \tag{2.136}$$

$$\mathbf{W}^d_n = \text{span}\,\overline{\{\Psi^d_{n,j,k}(x,y) : 0 \le j,k \le 2^n - 1\}} \tag{2.137}$$

If we restrict ourselves to functions $f(x,y)$ that admit the wavelet expansion

$$f(x,y) = b_{0,0,0}\Phi_{0,0,0}(x,y) + \sum_{i=-\infty}^{0}\sum_{j=0}^{2^{-i}-1}\sum_{k=0}^{2^{-i}-1}\left[c^h_{i,j,k}\Psi^h_{i,j,k}(x,y) + c^v_{i,j,k}\Psi^v_{i,j,k}(x,y) + c^d_{i,j,k}\Psi^d_{i,j,k}(x,y)\right], \tag{2.138}$$

then the coefficients may be conveniently arranged into a "pyramid" (see [Dau92, Mal98]) of blocks at each level of decomposition. There are three *subbands*: horizontal, vertical and diagonal. Denote the high resolution (detail) blocks as D^h_k, D^v_k, D^d_k respectively for each decomposition level $k \geq 0$. At each level, the blocks consist of 2^{2k} coefficients $d^h_{k,i,j}, d^v_{k,i,j}, d^d_{k,i,j}$. If the decomposition stops at some point before reaching level $k = 0$, there will be an *average* image of low resolution coefficients that may be denoted by A_k. Figure 2.7 shows these components arranged into the standard 'pyramid' form. The extension to $\mathcal{L}^2(\mathbb{R}^3)$ etc., is achieved in the analogous way.

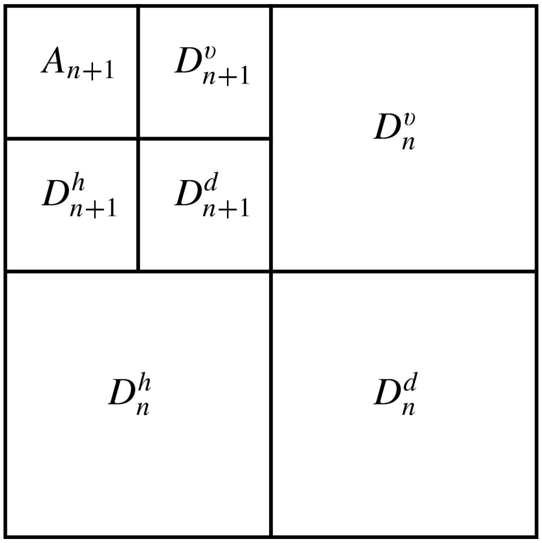
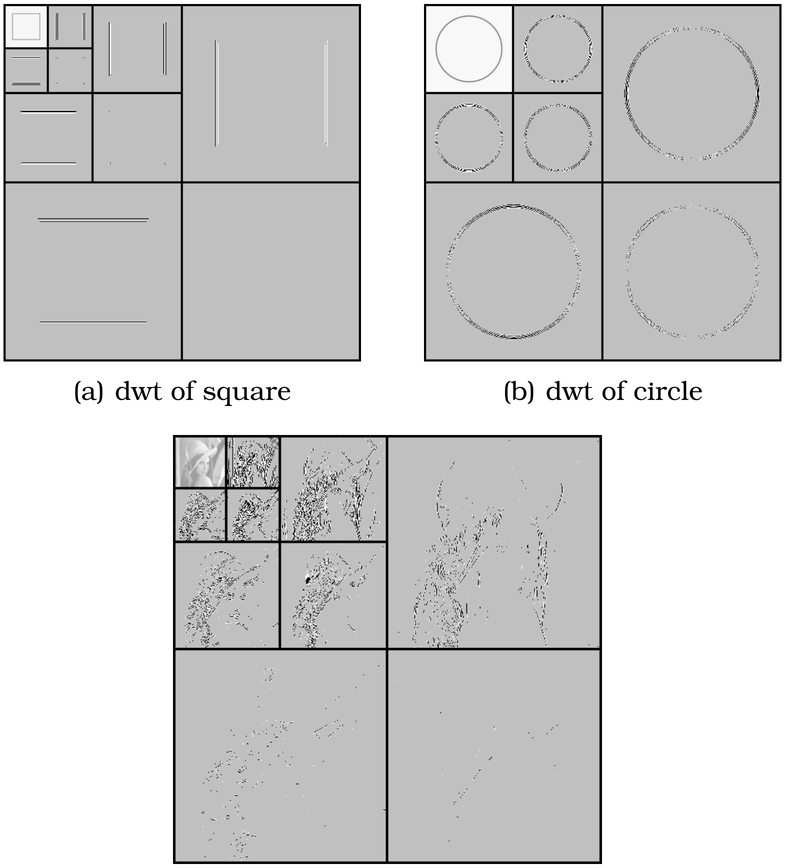


Figure 2.7: Coefficient pyramid showing decomposition at levels n and $n+1$

For the purpose of illustration, Figure 2.8 demonstrates several images under a tensor product basis of the Haar wavelet and scaling function.



(a) dwt of square

(b) dwt of circle

(c) DWT of Lena

Figure 2.8: Examples of DWT with two-dimensional separable wavelet bases. Because the size of the coefficients decreases rapidly for higher resolutions, the images have been manipulated. The upper-left corner contains the average image at the final level. All other areas have been mapped so that the pixel is white for coefficients $c > 0.1$, black for $c < -0.1$, and otherwise grey. (a) a square with 2 level DWT (b) a circle with 2 level DWT (c) “Lena” image with 3 level DWT

2.4.2 Wavelet Coefficient Trees

For any function $f \in \mathcal{L}^2(\mathbb{R})$ and an orthonormal basis $\{\psi_{j,k}\}$ of $\mathcal{L}^2(\mathbb{R})$,

$$f = \sum_{j,k} \langle f, \psi_{j,k} \rangle \psi_{j,k} .$$

If, for reasons that will become apparent in §2.4, we restrict ourselves to wavelet expansions of the following form

$$f(x) = b_{0,0}\phi(x) + \sum_{j=-\infty}^{0} \sum_{k=0}^{2^{-j}-1} c_{j,k}\psi_{j,k}(x) , \tag{2.139}$$

then the coefficients $c_{j,k} = \langle f, \psi_{j,k} \rangle$ (plus $b_{0,0}$, if non-zero) may be represented as a binary tree:

$b_{0,0}$							
$c_{0,0}$							
$c_{-1,0}$				$c_{-1,1}$			
$c_{-2,0}$		$c_{-2,1}$		$c_{-1,2}$		$c_{-1,3}$	
$c_{-3,0}$	$c_{-3,1}$	$c_{-3,2}$	$c_{-3,3}$	$c_{-3,4}$	$c_{-3,5}$	$c_{-3,6}$	$c_{-3,7}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

A more compact representation is

$b_{0,0}$			
$c_{0,0}$			
$c_{-1,0}$		$c_{-1,1}$	
$C_{-2,0}$	$C_{-2,1}$	$C_{-1,2}$	$C_{-1,3}$

where each $C_{j,k}$ represents a binary subtree of infinite depth.

For higher dimensional signals, similar trees exist. With a two-dimensional image, the coefficient tree is a *quadtree*; in three dimensions, the trees are *octrees*, and so on. The wavelet expansion in $\mathcal{L}^2(\mathbb{R}^2)$ analogous to Equation 2.139 will be (using the notation introduced in §2.4.1)

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

The resulting coefficients $c^b_{i,j,k}$ for $b \in \{h, v, d\}$ can be organised into a quadtree as shown in Figure 2.9. These structures will be useful in discussing the manipulation of wavelet coefficients.

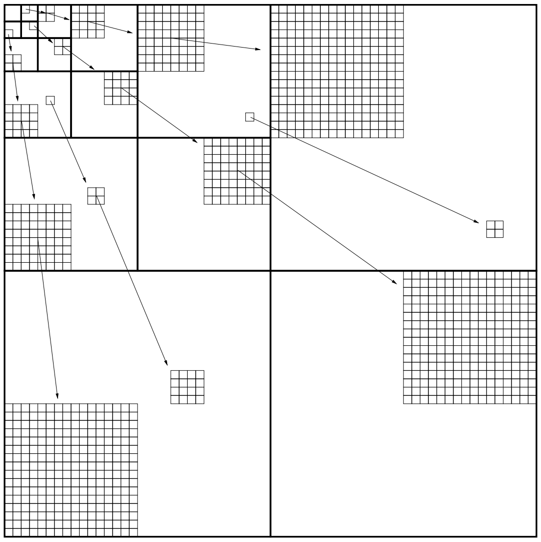


Figure 2.9: Discrete two-dimensional MRA tree structure showing quadtrees rooted at various levels

Bases on an Interval

Until now, we have discussed bases of $\mathcal{L}^2(\mathbb{R})$. In applications, we need to consider the effect of finite length signals, specifically the issue of what happens at the edges. Without loss of generality, consider signals defined on the interval $[0, 1]$. The key question is "what will happen when the support of the wavelet or scaling functions overlaps the interval?". It turns out that a wavelet basis of $\mathcal{L}^2(\mathbb{R})[0,1]$ can be constructed from any wavelet basis of $\mathcal{L}^2(\mathbb{R})$ and the associated scaling functions [Mal98]. These interval bases consist of the set of wavelet and scaling functions

$$\left\{ \{\phi_{J,n}^{\text{int}}\}_{0 \le n < 2^{-J}},\ \{\psi_{j,n}^{\text{int}}\}_{-\infty < j \le J,\ 0 \le n < 2^{-j}} \right\}. \tag{2.140}$$

Where the superscript "int" denotes wavelet and scaling functions modified for the interval. For details, see [DF92, Dau92, CDV93, Mal98, Top98, Bri96, Bri95]

There are three primary methods of constructing such a basis.

Periodic Extension

This method is the simplest to implement; simply periodize the signal f over $\mathcal{L}^2(\mathbb{R})$, resulting in periodic wavelets. The disadvantage of this method is that the boundary wavelets (those whose support overlapped the interval before periodizing) have no vanishing moments [Mal98]. This leads to large wavelet coefficients at the edges [Mal98].

Symmetric Extension, or folded wavelets

This method is a bit more complicated to implement, but has the advantage that boundary wavelets retain a vanishing moment. For many wavelet bases, a single vanishing moment is significantly less than the internal wavelets, so coefficients at the boundary will still be unnecessarily large.

Symmetric extensions can be approached purely from a subband coding point of view, for a larger class of two-channel filter banks than have been considered. Briefly, the input signal is extended by *folding* at the boundaries in a symmetric or anti-symmetric manner. The symmetry of the extension is determined by the length and symmetry characteristics of the filters and input signal, and furthermore by the constraint of perfect reconstruction [Top98, Bri95, Bri96].

The method of Symmetric Extension is used in the applications presented later. Figure 2.10 demonstrates both symmetric and anti-symmetric extensions of a signal.

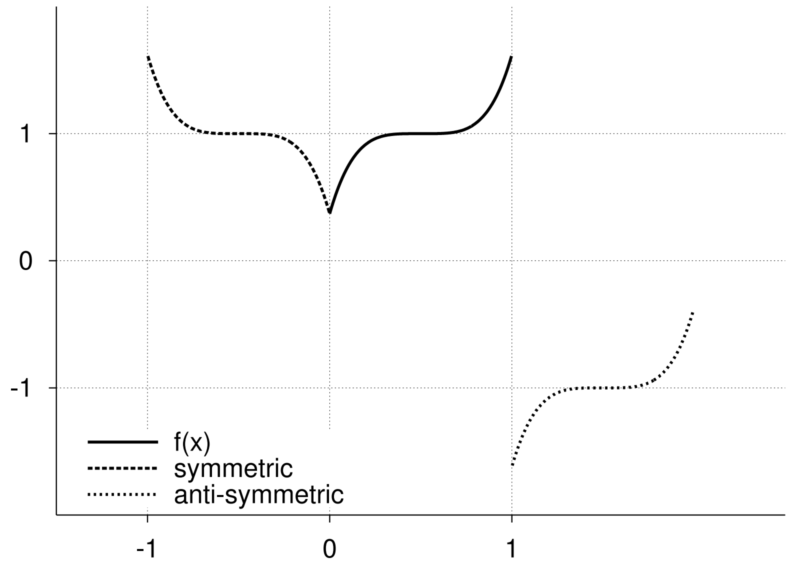


Figure 2.10: symmetric (left) and anti-symmetric (right) extensions of a signal on $[0, 1]$

Boundary Wavelets

It is possible [CDV93] to explicitly construct boundary wavelets with as many vanishing moments as the original wavelet ϕ, thereby avoiding large coefficients at the boundaries. Implementation is more difficult than the previous two methods, and this approach is not used in the applications presented later.

2.4.3 IFS on Wavelets

This section describes an application of the IFSM to the wavelet domain, called Iterated Function Systems on Wavelet trees (IFSW), as introduced in [FV98b, Vrs98, MV97].

Consider $f \in \mathcal{L}^2(\mathbb{R})$ that admit the wavelet expansions[4](see §2.4.2)

$$f(x) = b_{0,0}\phi(x) + \sum_{j=-\infty}^{0} \sum_{k=0}^{2^{-j}-1} c_{j,k}\psi_{j,k}(x). \tag{2.141}$$

Recall from §2.4.1 the wavelet ψ and scaling function ϕ generate families of translated and dilated versions

$$\phi_{j,k}(x) = 2^{-j/2}\phi(2^{-j}x - k) \qquad \forall j, n \in \mathbb{Z} \tag{2.142}$$

$$\psi_{j,k}(x) = 2^{-j/2}\psi(2^{-j}x - k) \qquad \forall j, n \in \mathbb{Z}. \tag{2.143}$$

Recalling the 'coefficient trees' of the previous section, to aid the discussion, to motivate the IFSW, consider the following example.

Example 2.5 Consider a function f under the Haar basis. Recall (Example 2.4) that the Haar basis has

$$\phi(x) = I_{[0,1]}(x) \tag{2.144}$$

$$\psi(x) = I_{[0,1/2]}(x) - I_{[1/2,1]}(x). \tag{2.145}$$

Define the simple two-map IFSM defined by

$$w_1(x) = \frac{1}{2}x, \qquad \phi_1(x) = \alpha_1 x + \beta_1, \tag{2.146}$$

$$w_2(x) = \frac{1}{2}x + \frac{1}{2}, \qquad \phi_2(x) = \alpha_2 x + \beta_2. \tag{2.147}$$

[4] In practice, finite resolution is needed, so j will sum from $L < 0$, not $-\infty$.

The fractal operator (from Equation 2.67) will be

$$
\begin{aligned}
(Tf)(x) &= \sum_{i=1}^{2} g_i(x) && (2.148)\\
&= \alpha_1 f(2x) + \beta_1 I_{[0,1/2]} + \alpha_2 f(2x-1)\beta_2 I_{[1/2,1]} && (2.149)\\
&= \alpha_1 b_{0,0}\phi_{0,0}(2x) + \alpha_1 \sum_{j=-\infty}^{0} \sum_{k=0}^{2^{-j}-1} c_{j,k}\psi_{j,k}(2x) + \beta_1 I_{[0,1/2]} && (2.150)\\
&\quad + \alpha_2 b_{0,0}\phi_{0,0}(2x-1) + \alpha_2 \sum_{j=-\infty}^{0} \sum_{k=0}^{2^{-j}-1} c_{j,k}\psi_{j,k}(2x-1) + \beta_2 I_{[1/2,1]}\,. && (2.151)
\end{aligned}
$$

Note that

$$
\begin{aligned}
\psi_{j,k}(2x-1) &= 2^{-j/2}\psi\left(2^{-j}\,(2x-1) - k\right) && (2.152)\\
&= 2^{-1/2}\,2^{\frac{-(j-1)}{2}}\psi\left(2^{-(j-1)}\,x - (k+2^{-j})\right) && (2.153)\\
&= 2^{-1/2}\,\psi_{j-1,k+2^{-j}}(x). && (2.154)
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
\psi_{j,k}(2x) &= 2^{-1/2}\psi_{j-1,k}(x), && (2.155)\\
\phi_{j,k}(2x-1) &= 2^{-1/2}\phi_{j-1,k+2^{-j}}(x), && (2.156)\\
\phi_{j,k}(2x) &= 2^{-1/2}\phi_{j-1,k}(x). && (2.157)
\end{aligned}
$$

Substituting these into Equation 2.151 and regrouping terms yields

$$(Tf)(x) = \frac{\alpha_1}{\sqrt{2}} \sum_{j=-\infty}^{0} \sum_{k=0}^{2^{-j}-1} c_{j,k}\psi_{j-1,k}(x) + \frac{\alpha_2}{\sqrt{2}} \sum_{j=-\infty}^{0} \sum_{k=0}^{2^{-j}-1} c_{j,k}\psi_{j-1,k+2^{-j}}(x) \tag{2.158}$$

$$+ \frac{\alpha_1}{\sqrt{2}} b_{0,0}\phi_{-1,0}(x) + \beta_1 I_{[0,1/2]} + \frac{\alpha_2}{\sqrt{2}} b_{0,0}\phi_{-1,1}(x) + \beta_2 I_{[1/2,1]} \tag{2.159}$$

$$= \frac{\alpha_1}{\sqrt{2}} \sum_{j=-\infty}^{0} \sum_{k=0}^{2^{-j}-1} c_{j,k}\psi_{j-1,k}(x) + \frac{\alpha_2}{\sqrt{2}} \sum_{j=-\infty}^{0} \sum_{k=0}^{2^{-j}-1} c_{j,k}\psi_{j-1,k+2^{-j}}(x) \tag{2.160}$$

$$+ (\alpha_1 b_{0,0} + \beta_1) I_{[0,1/2]} + (\alpha_2 b_{0,0} + \beta_2) I_{[1/2,1]}\,. \tag{2.161}$$

Ignoring the constant contributions, the coefficient tree may be written as

<table>
<tr><td colspan="4">0</td></tr>
<tr><td colspan="4">0</td></tr>
<tr><td colspan="2">$\frac{\alpha_1}{\sqrt{2}} c_{0,0}$</td><td colspan="2">$\frac{\alpha_2}{\sqrt{2}} c_{0,0}$</td></tr>
<tr><td>$\frac{\alpha_1}{\sqrt{2}} C_{-1,0}$</td><td>$\frac{\alpha_1}{\sqrt{2}} C_{-1,1}$</td><td>$\frac{\alpha_2}{\sqrt{2}} C_{-1,0}$</td><td>$\frac{\alpha_2}{\sqrt{2}} C_{-1,1}$</td></tr>
</table>

Or, more compactly,

<table>
<tr><td colspan="2">0</td></tr>
<tr><td colspan="2">0</td></tr>
<tr><td>$\frac{\alpha_1}{\sqrt{2}} C_{0,0}$</td><td>$\frac{\alpha_2}{\sqrt{2}} C_{0,0}$</td></tr>
</table>

Thus we have defined an operator on the coefficient trees that maps the original tree $C_{0,0}$ onto two scaled copies of $C_{0,0}$, at one level lower in the tree. □

The above example gives the flavour of IFSW maps. Under the Haar basis, this process is equivalent to a local IFSM [FV98a]. This approach can be generalised to other compactly supported orthogonal[5] bases [FV98a, Dav98]. Details may be found in [Vrs98, MV97];

[5] The generalisation to biorthogonal bases has not been done.

here a simple example (from [MV97]) is presented in order to illustrate the process.

Consider mapping from the top level of the wavelet decomposition to the next level down, with four scaling maps:

$$W_1 : C_{-1,0} \to C_{-2,0} \quad C_{-2,0} = \alpha_1 C_{-1,0} \tag{2.162}$$

$$W_2 : C_{-1,1} \to C_{-2,1} \quad C_{-2,1} = \alpha_2 C_{-1,1} \tag{2.163}$$

$$W_3 : C_{-1,0} \to C_{-2,2} \quad C_{-2,2} = \alpha_3 C_{-1,0} \tag{2.164}$$

$$W_4 : C_{-1,1} \to C_{-2,3} \quad C_{-2,3} = \alpha_4 C_{-1,1} \tag{2.165}$$

Denote the fractal wavelet (FW) maps as M; it may be represented in a diagram as

$M : C_{0,0} \Longrightarrow$

<table>
<tr><td colspan="4">$c_{0,0}$</td></tr>
<tr><td colspan="2">$c_{-1,0}$</td><td colspan="2">$c_{-1,1}$</td></tr>
<tr><td>$\alpha_1 C_{-1,0}$</td><td>$\alpha_2 C_{-1,1}$</td><td>$\alpha_3 C_{-1,0}$</td><td>$\alpha_4 C_{-1,1}$</td></tr>
</table>

If M is iterated a second time, the result is

$M^{\circ(2)} : C_{0,0} \Longrightarrow$

<table>
<tr><td colspan="8">$c_{0,0}$</td></tr>
<tr><td colspan="4">$c_{-1,0}$</td><td colspan="4">$c_{-1,1}$</td></tr>
<tr><td colspan="2">$\alpha_1 c_{-1,0}$</td><td colspan="2">$\alpha_2 c_{-1,1}$</td><td colspan="2">$\alpha_3 c_{-1,0}$</td><td colspan="2">$\alpha_4 c_{-1,1}$</td></tr>
<tr><td>$\alpha_1^2 C_{-1,0}$</td><td>$\alpha_1 \alpha_2 C_{-1,0}$</td><td>$\alpha_2 \alpha_1 C_{-1,1}$</td><td>$\alpha_2^2 C_{-1,1}$</td><td>$\alpha_3^2 C_{-1,0}$</td><td>$\alpha_3 \alpha_4 C_{-1,0}$</td><td>$\alpha_4 \alpha_3 C_{-1,1}$</td><td>$\alpha_4^2 C_{-1,1}$</td></tr>
</table>

Iterating this process will converge to a wavelet expansion of a function in $\mathcal{L}^2(\mathbb{R})$ provided that $|a_i| < \frac{1}{\sqrt{2}}$ $\forall i$ [MV97]. Note that mapping from the first to second levels of the coefficient trees was merely one possible choice of maps W_n; many other choices of map are possible. In order to apply this approach to image compression, it is necessary to work in the space of two-dimensional wavelet coefficient trees.

Extension of the IFS theory to wavelet coefficient trees

A generalised approach to IFSW on images is described by Vrscay [Vrs98]. The following is a brief outline of those results. Instead of mapping binary trees as in the above example, quadtrees (or octrees, for three-dimensional images) are mapped to each other. The two-dimensional case is discussed here; the three-dimensional case is analogous For simplicity, make a restriction on the maps considered. There will be a *parent level* and a *child level*, and the roots of all parent trees will appear in one block, whilst the roots of all child trees appear in another block. The parent and child levels are represented by integers i_1^* and i_2^*, respectively, where $i_2^* < i_1^* \leq 0$.

Recalling the notation of §2.4.1, we wish to find functions $f \in \mathcal{L}^2(\mathbb{R})$ that admit wavelet expansions of the form

$$f(x,y) = b_{0,0,0}\Phi_{0,0,0}(x,y) + \sum_{i=-\infty}^{0}\sum_{j=0}^{2^{-i}-1}\sum_{k=0}^{2^{-i}-1}\left[c_{i,j,k}^{h}\Psi_{i,j,k}^{h}(x,y) + c_{i,j,k}^{v}\Psi_{i,j,k}^{v}(x,y) + c_{i,j,k}^{d}\Psi_{i,j,k}^{d}(x,y)\right]. \tag{2.166}$$

Definition 2.16 *Denote the space of all $f \in \mathcal{L}^2(\mathbb{R})$ admitting wavelet expansions of the form given by Equation 2.166 as $\mathcal{L}_0^2(\mathbb{R})$. This space is complete with respect to the usual $\mathcal{L}^2(\mathbb{R})$ metric [Vrs98].* □

The wavelet coefficients may be arranged into a 'pyramid' of blocks (e.g., Figure 2.7), consisting of the zeroth level average block A_0 and the detail blocks $D_{i_1^*,j,k}^{\lambda}$. Here $\lambda \in \{h, v, d\}$ and $i_2^* + 1 \leq i_1^* \leq 0$.

Define the following sets of affine transformations on the blocks. The child blocks are two-dimensional, with $2^{2(-i_2^*)}$ coefficients ranging $0 \leq j,k \leq 2^{-i_2^*} - 1$. To emphasise the fixed parent-child block relationships, we denote (for each subband) the parent's array positions as functions of the child's; for example $j^h(j,k)$ and $k^h(j,k)$. In this setting, the block transforms will be:

$$\begin{aligned}
W^h_{j,k} &: D^h_{i_1^*, j^h(j,k), k^h(j,k)} \to D^h_{i_2^*, j,k}, \qquad D^h_{i_2^*, j,k} = \alpha^h_{j,k} D^h_{i_1^*, j^h(j,k), k^h(j,k)} \\
W^v_{j,k} &: D^v_{i_1^*, j^v(j,k), k^v(j,k)} \to D^v_{i_2^*, j,k}, \qquad D^v_{i_2^*, j,k} = \alpha^v_{j,k} D^v_{i_1^*, j^v(j,k), k^v(j,k)} \\
W^d_{j,k} &: D^d_{i_1^*, j^d(j,k), k^d(j,k)} \to D^d_{i_2^*, j,k}, \qquad D^d_{i_2^*, j,k} = \alpha^d_{j,k} D^d_{i_1^*, j^d(j,k), k^d(j,k)}.
\end{aligned} \tag{2.167}$$

The above defines an IFSW operator, call it M. In order for the IFS theory to apply to this operator, it remains to show that M is contractive in some suitable complete metric space of wavelet coefficients.

Definition 2.17 (Coefficient quadtree space $\mathcal{Q}$) *Similar to the example in one-dimension, denote an (infinite) quadtree as C and the coefficients of C as $c_{i,j,k}$ If a subtree is rooted on subband λ at level i, position j, k, denote it as $C^\lambda_{i,j,k}$. Let $\mathcal{Q}$ denote the set of all real quadtrees C that are square-summable,*

$$\mathcal{Q} = \left\{ C : c_{i,j,k} \in \mathbb{R}, i \leq 0, 0 \leq j, k \leq 2^{-i} - 1, \sum_{i,j,k} |c_{i,j,k}|^2 < \infty \right\}. \tag{2.168}$$

□

Definition 2.18 (Quadtree metric) *Denote the ℓ^2 metric on $\mathcal{Q}$ as $d_{\mathcal{Q}}$. That is,*

$$d_{\mathcal{Q}}(C, D) = \left[\sum_{i,j,k} |c_{i,j,k} - d_{i,j,k}|^2 \right]^{1/2}, \ \forall \ C, D \in \mathcal{Q} \tag{2.169}$$

□

Definition 2.19 (Quadtree product) *The inner product between two quadtrees in $\mathcal{Q}$ is*

$$\langle C, D \rangle_{\mathcal{Q}} = \sum_{i,j,k} c_{i,j,k} d_{i,j,k}, \ \ C, D \in \mathcal{Q} \tag{2.170}$$

□

In application to image compression, the goal will be to approximate functions $f \in \mathcal{L}^2_0(\mathbb{R}^2)$ with wavelet coefficient expansions as given in Equation 2.166. For this reason, we define a subset of $\mathcal{Q}$

for which the wavelet coefficient blocks A_0 and $D^{\lambda}_{i_1^*,j,k}$, $\lambda \in \{h, v, d\}$, $i_2^* + 1 \leq i_1^* \leq 0$ are fixed.

Definition 2.20 *For any $f \in \mathcal{L}^2_0(\mathbb{R}^2)$ and $i_2^* \in \mathbb{N}$, Let $\mathcal{Q}_{f,i_2^*}$ denote the set of all quadtrees in $\mathcal{Q}$ with blocks at levels $i_2^* + 1 \leq i \leq 0$ fixed.* □

The following propositions lay the framework for applying the previously discussed IFSM results to this space [Vrs98].

Proposition 2.5 *The distance function*

$$d_{\mathcal{Q}_{f,i_2^*}}(C, D) = \max_{\lambda,j,k} d_{\mathcal{Q}}\left(C^{\lambda}_{i_2^*,j,k}, D^{\lambda}_{i_2^*,j,k}\right), \;\; C, D \in \mathcal{Q}_{f,i_2^*} \tag{2.171}$$

defines a metric on $\mathcal{Q}_{i_2^*}$. Furthermore, the metric space $(\mathcal{Q}_{f,i_2^*}, d_{\mathcal{Q}_{i_2^*}})$ is complete. □

Proposition 2.6 *Given $f \in \mathcal{L}^2_0(\mathbb{R}^2)$ and $-i_2^* \in \mathbb{N}^+$, the operator M (Equation 2.167) maps $\mathcal{Q}_{f,i_2^*}$ into itself. Furthermore,*

$$d_{\mathcal{Q}_{f,i_2^*}}(M(C), M(D)) \leq c_{f,i_2^*} d_{\mathcal{Q}_{f,i_2^*}}(C, D), \tag{2.172}$$

where

$$c_{f,i_2^*} = 2^{-(i_2^* - i_1^*)} \max_{\lambda,j,k} |\alpha^{\lambda}_{j,k}| \tag{2.173}$$

□

Corollary 2.3 *By the Banach contraction mapping principle, if $c_{f,i_2^*} \leq 1$ then there is a unique fixed point of M. That is, there exists a unique $\bar{C} \in \mathcal{Q}_{f,i_2^*}$ such that $M(\bar{C}) = \bar{C}$. $\bar{C}$ is an attractor of M; iteration of M will converge to $\bar{C}$.* □

The inverse problem for IFSW

Having made the connection between wavelet coefficient trees and IFS type operators, the previously developed theory may now be applied. Thus the collage theorem is again applicable, and the inverse problem can be stated in this context as:

> Let $f \in \mathcal{L}_0^2(\mathbb{R}^2)$ and $-i_2^* \in \mathbb{N}^+$. Let $\epsilon > 0$. Does an IFSW map M exist such that $d_{\mathcal{Q}_{f,i_2^*}}(C, M(C)) \leq \epsilon$?

As was the case with IFSM, the approximation problem has become a question of minimising the collage distance Δ:

$$\begin{aligned}\Delta &= d_{\mathcal{Q}_{f,i_2^*}}(C, M(C)) \\ &= \max_{\lambda,j,k} \Delta_{j,k}^{\lambda}, \end{aligned} \tag{2.174}$$

where

$$\Delta_{j,k}^{\lambda} = d_{\mathcal{Q}_{f,i_2^*}}(D_{i_2^*,j,k}^{\lambda}, \alpha_{j,k}^{\lambda} D_{i_1^*,j^\lambda(j,k),k^\lambda(j,k)}^{\lambda}) \; \lambda \in \{h, v, d\} \; 0 \leq j, k \leq 2^{-i_2^*} - 1 \tag{2.175}$$

This is a quadratic programming problem. In practice, similar to the IFSM case, this minimisation problem is treated as a "least-squares" fit to find the scaling factors $\alpha_{j,k}^{\lambda}$ by

$$\alpha_{j,k}^{\lambda} = \frac{\langle D_{i_1^*,j^\lambda(j,k),k^\lambda(j,k)}^{\lambda}, D_{i_2^*,j,k}^{\lambda} \rangle_{\mathcal{Q}}}{\langle D_{i_2^*,j,k}^{\lambda}, D_{i_2^*,j,k}^{\lambda} \rangle_{\mathcal{Q}}} \tag{2.176}$$

In order to ensure contraction of the operator M, there is a constraint of $|\alpha_{j,k}^{\lambda}| \leq 2^{i_2^* - i_1^*}$. In practice, this is rarely checked. In fact, the nature of wavelet coefficient decay near strong edges in the spatial domain makes this condition difficult to meet for all scaling factors[Vrs98].

Remarks

In this chapter, IFS as a method of approximation of a set was introduced. This forms the core behind fractal methods of compression, but is not directly applicable to non-binary images (the majority of cases of interest). The extension of IFS to the IFSM operating on $\mathcal{L}^2(\mathbb{R})$, and associating the IFS with a series of grey level maps was made. The concept of IFSW, which are IFSMs applied to wavelet coefficients was also introduced.

The inverse problem for all three (IFS, IFSM, IFSW) domains was discussed. The collage theorem was introduced as an approach to the problem. Although the notation becomes quite cumbersome, it should be apparent from the preceding discussion that the approach to deal with wavelet coefficient trees in two dimensions will extend to higher dimensions in a straightforward manner.

CHAPTER 3

IFS Applications

Chapter 2 included a generic development of the ideas behind IFS, IFSM, and IFSW. Before going much further, a more concrete description of how this may be applied to 'an image' is needed. Recalling the discussion (§ 1.1) about *natural images* we can take a natural view and consider a grayscale image to be a series of 'measurements' taken from a surface.

We will consider an image I to be defined by an *image function* $u(x, y)$ supported over an appropriate region $X \in \mathbb{R}^2$. We take x, y as the spatial co-ordinates of a point/pixel in the image. Figure 3.1 shows the Lena image as a mesh representing such a function u.

We begin by breaking the image region X into subblocks, or range blocks, R_i. We take the set $\{R_i\}$ to be a proper partition, i.e. they intersect only on the boundaries (assumed to be of zero Lesbesgue measure in the plane) although construction with overlapping range blocks can be made. Of course in the cases of interest herein, which are discrete images, there is no difficulty with overlap.

We will construct a *fractal transform operator* T. Assume, for the moment, that we have found for each range block R_i a domain block D_i which is a (larger) subblock of the image, with a 1-1 contraction

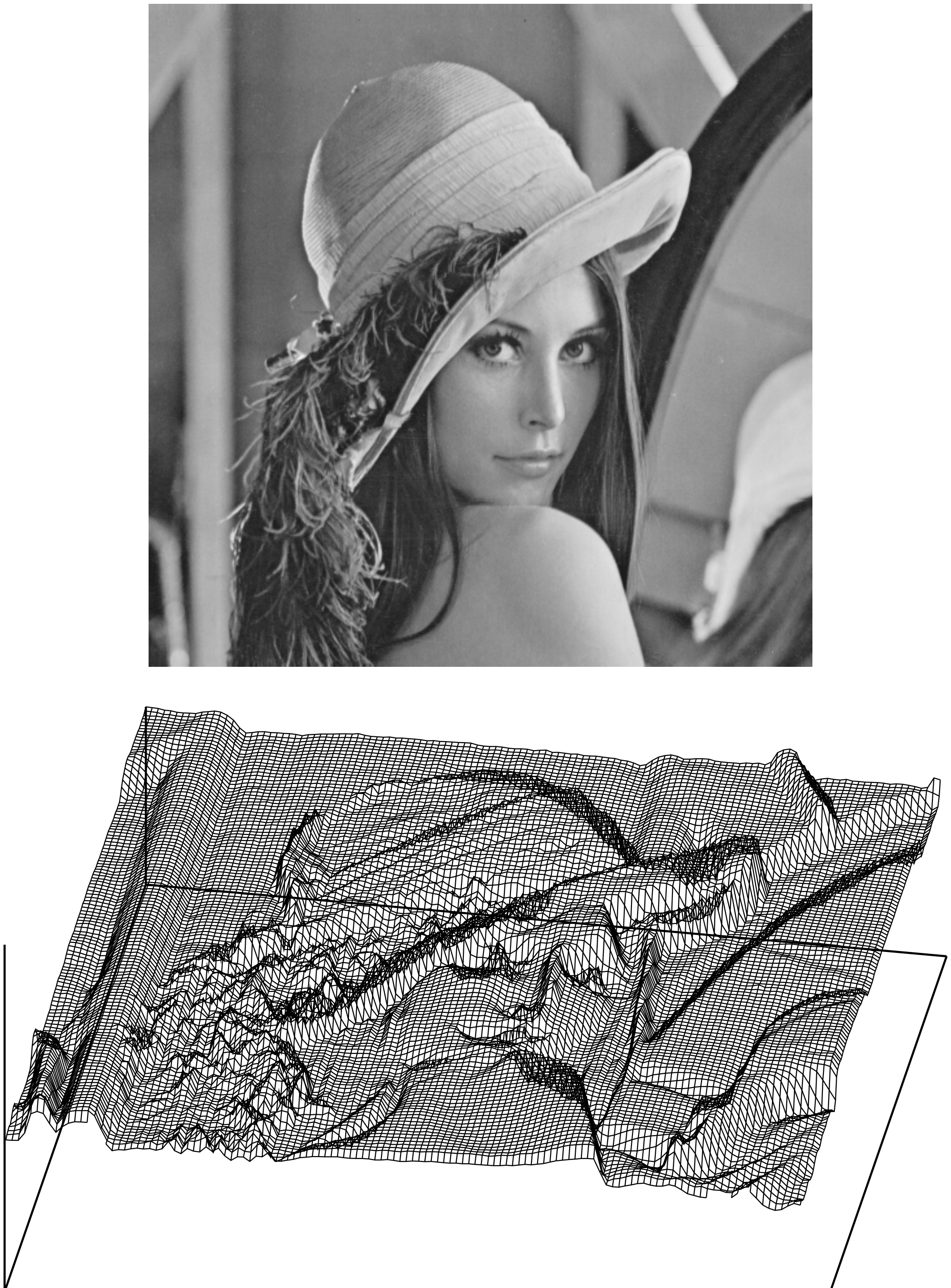


Figure 3.1: Lena as a surface

w_i such that $R_i = w_i(D_i)$. In this way, the support of the domain is mapped to the support of the range. We further make the assumption that the image function supported on the range block has a good approximation by a modified copy of the domain. That is,

$$u(R_i) \approx \phi_i(u(D_i)) = \phi(u(w_i^{-1}(R_i)) \qquad (3.1)$$

is a good approximation, where $\phi_i : \mathbb{R} \Rightarrow \mathbb{R}$ is a *grayscale* map. This map ϕ_i will operate on the grayscale intensity values. Figure 3.2 shows visually what the two parts of this approximation mapping are doing.

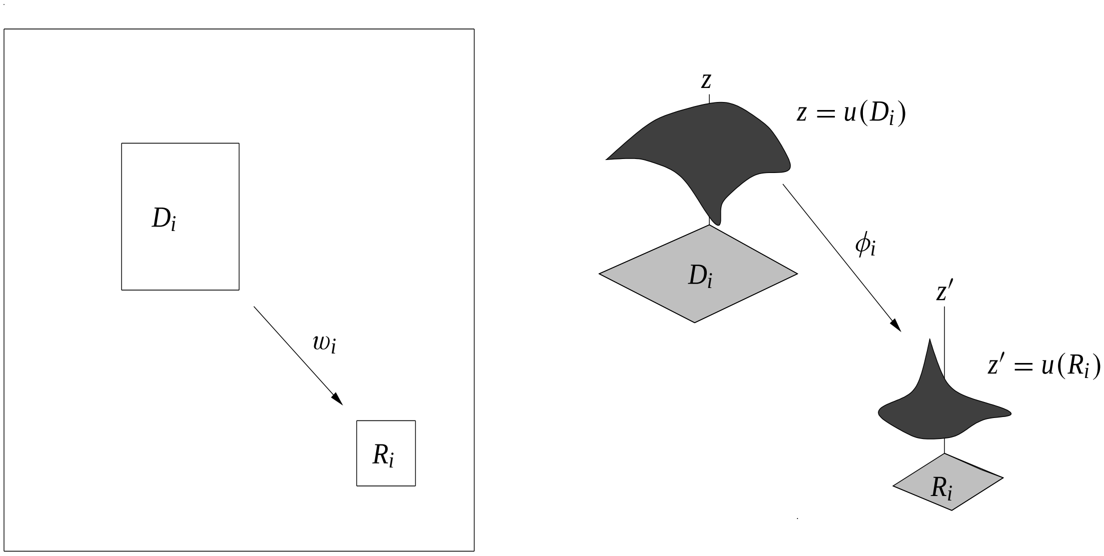


Figure 3.2: Domain and range blocks *Figure provided by E. R. Vrscay*

So to approximate $u(x, y)$, we put together all of the blocks:

$$u(x, y) \approx (Tu)(x, y) = \sum_i \phi_i(u(w_i^{-1}(x, y))) \ . \qquad (3.2)$$

Since this operator is contractive, following the discussion of chapter 2, it has a fixed point $\bar{u}$, which is an approximation to the original image u. If this approximation is good, the *collage distance* $\|u - Tu\|$

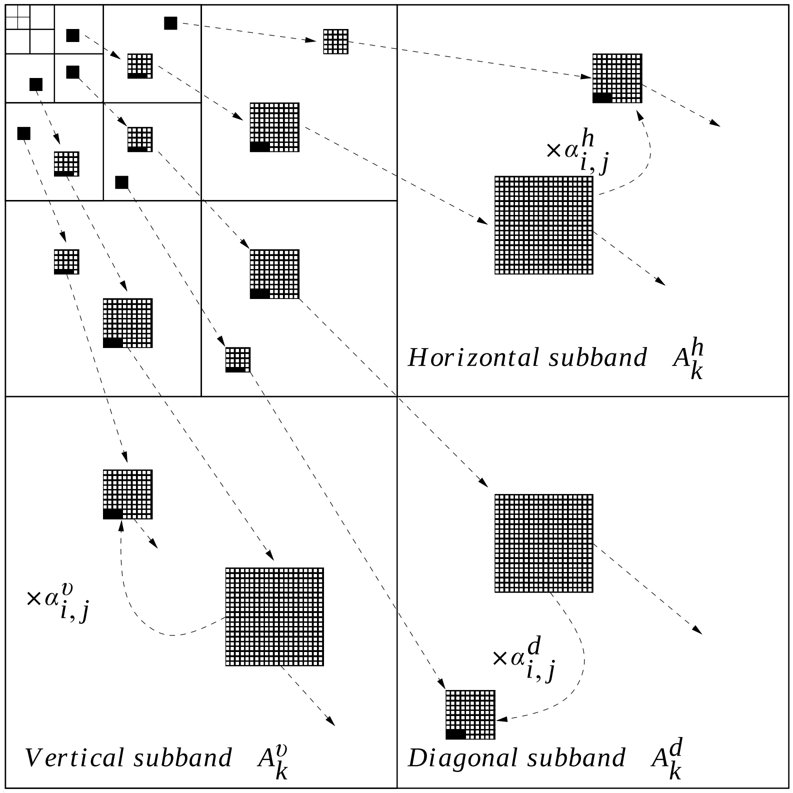

$\times\alpha^h_{i,j}$
Horizontal subband A^h_k
$\times\alpha^v_{i,j}$
Vertical subband A^v_k
$\times\alpha^d_{i,j}$
Diagonal subband A^d_k

Figure 3.3: 'Parent' and 'child' trees in a wavelet decomposition
Figure provided by E. R. Vrscay

will be small, following theorem 2.5.

A similar approach is taken in the wavelet domain, as visually represented in Figure 3.3

A comment on partitions

It should be clear that the particular partition used in constructing our 'fractal operator' is important. Indeed, a large amount of effort was expended in devising partitioning schemes for fractal compression coders (see, for example, [Har98]). However, this research was constrained by the necessity of *encoding* the partition information, and hence the question was more 'what can be done cheaply?' (with respect to the 'cost' of encoding this information) that it was 'what should be done?'. In a context where this constraint is lifted, more thought should be given to this aspect than has been the case. Two things are apparent: First, effective partitioning is related to the structure of the image, particularly the edges [Har98, Fis95, SH94], and second edge information is important to human understanding of images[MS89]. Taken together, this suggests a serious investigation of the information trade-offs between partitioning and the component maps would be worthwhile. This is, however, outside the scope of the current work.

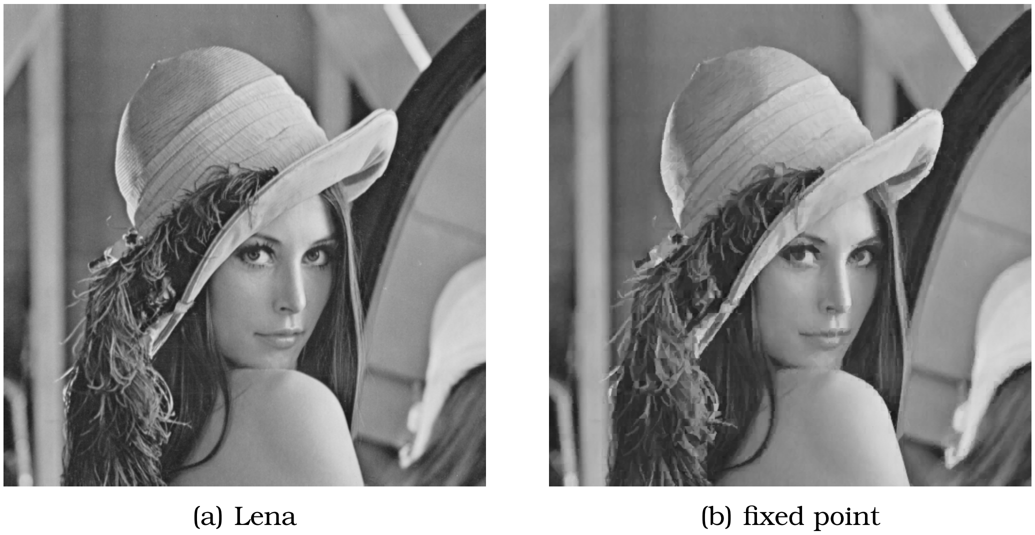
3.1 Life after compression?

As mentioned previously, the use of IFS in image processing was primarily driven by the application of image compression [FV98b, Dav98, Vrs98, Vrs96, BH93, Jac92] which peaked sometime in the early nineteen nineties, to be replaced by wavelet based methods [Sha93, SP96]. Although some work on *fractal zoom* [PD97] and *denoising* [GFV03] has been done, on the whole little has been developed outside the scope of image compression.

One clear question, then, is 'what does IFS buy us?' if not a compact representation. It is important to realise that by representing an image as the fixed point of an IFS operator, we are asserting an *implicit image model*. Specifically, in the case of the IFSM described in §2.11 we are saying that image elements are well represented as scaled affine maps of other regions of the image.

If we believe this is a reasonable thing to do, what sort of information can we glean about the behaviour of so-called *natural images* in such a framework? This chapter will largely detail an initial attempt at answering this question.

Philosophically speaking, essentially what is being asked is: "How does the IFS approach fare as a phenomenological model of images?". At a very abstract level, this is a response to the lack of a model for "natural images". If we can't start with a solid idea of what an image is[1], can we perhaps learn things of interest from the particular image in question?



(a) Lena

(b) fixed point

Figure 3.4: a) Original b) Fixed point $\bar{u}$ of the operator T. Error between these images is a PSNR of 31.56

Figure 3.4 shows the original *Lena* image, and fixed point $\bar{u}$ of the fractal transform operator T, along with the PSNR (recall (1.4)) between them (as defined in (1.4)). The fractal transform T was obtained by collage coding using 4096 8×8 non-overlapping pixel range blocks on the 512×512 image. The domain pool consisted of the set of $32^2 = 1024$ non-overlapping 16×16 pixel blocks, using all eight possible geometric contraction maps. Since all eight geometric transformations w_i were examined, a total of 8192 blocks were examined for each range block. The clamping condition $|\alpha_{\max}| = 1$ was also employed.

The best- and worst-case fits are interesting, and provide some insight into how this process is working. We identify the *best* best-fit and *worst* best-fit range blocks. By this, we mean the range blocks with the least and most collage error for their best-fit domain block, respectively. Figure 3.5 shows the location of these two blocks. The

[1] Or, for that matter, even if we could.

best range block is on the smooth region of Lena's shoulder, while the worst fit one is on the side of the mirror and contains a curved edge.

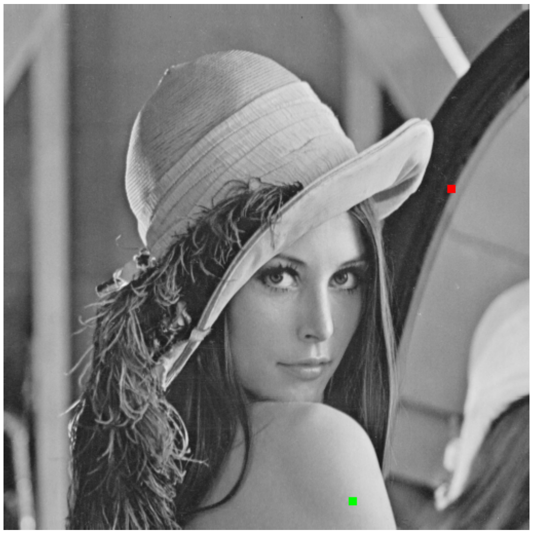
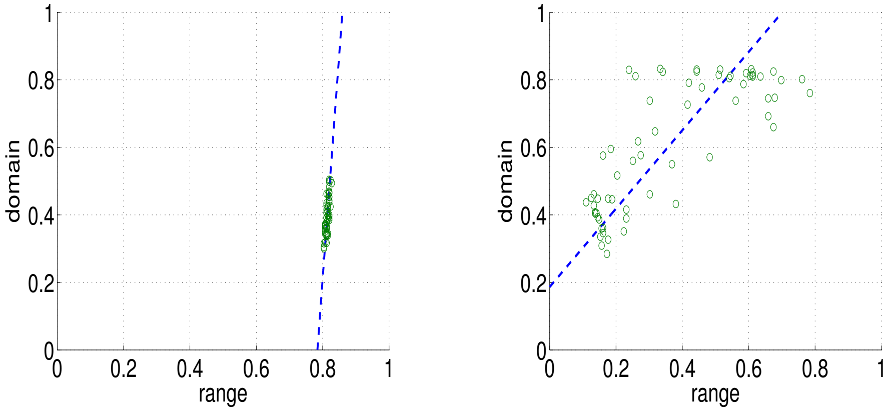


Figure 3.5: Best (green) and worst (red) range blocks.

Figure 3.6 shows scatter-plots of two (of the 4096) actual linear fits used to parameterise the operator T. This shows the blocks with both the *best* (i.e. least collage error) collage fit (a) of all the block and the *worst* collage fit (b). We are here fitting the affine greyscale maps $\phi(t) = \alpha t + \beta$, finding the parent-child matching (for each child) with the lowest collage error. These plots represent parent-child maps from the fractal transform T with fixed point $\bar{u}$ shown in Figure 3.4.

Having identified, in some sense, the *best* and *worst* child blocks in the partition of the image, we consider how other (non-optimal) parent blocks map to each of these. A histogram of the collage-error was generated for both children. These distributions are shown in Figure 3.7, note the same horizontal scaling is used for each in order to facilitate comparison of the mass distribution.

The best fit child is block $(480, 336)$ with a collage error of $1.087e^{-5}$. The worst fit child block, $(176, 432)$, has collage error of 0.0155. As may be expected, this difference of many orders of magnitude is also reflected in the variances of the two blocks. The best fit child has

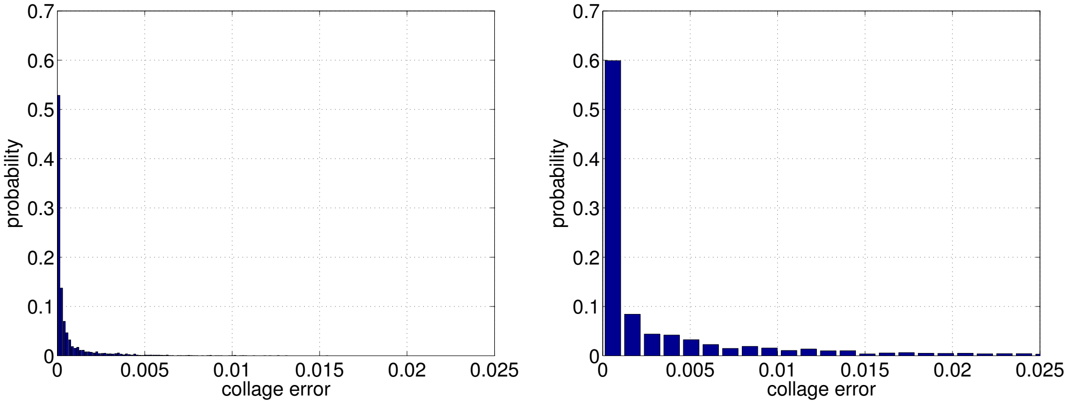


(a) best *best fit* (b) worst *best fit*

Figure 3.6: Plots of domain block vs. range block pixel values used to determine the best (L^2 sense) affine greyscale maps $\phi(t) = \alpha t + \beta$. Left: The "best collage fit", i.e., the parent-child matching with the lowest collage error, used in the fractal transform T with fixed point $\bar{u}$ shown in Figure 1. Right: The 'worst collage fit", i.e., the parent-child matching with the highest collage error, used in T. In both plots, the best-fit affine greyscale maps are also shown. The dashed lines show the linear least squares fit described in (2.83).

variance of $\sigma^2 = 2.6360e^{-05}$, so it is quite 'flat'. For this reason, it is well fit by a piecewise constant block. Since fitting of any parent may achieve this by taking $\alpha = 0$, we see why many good fits are achievable in this case. By comparison, the worst-fit case has variance $\sigma^2 = 0.0424$, and is quite difficult to fit (as seen in Figure 3.7-b).

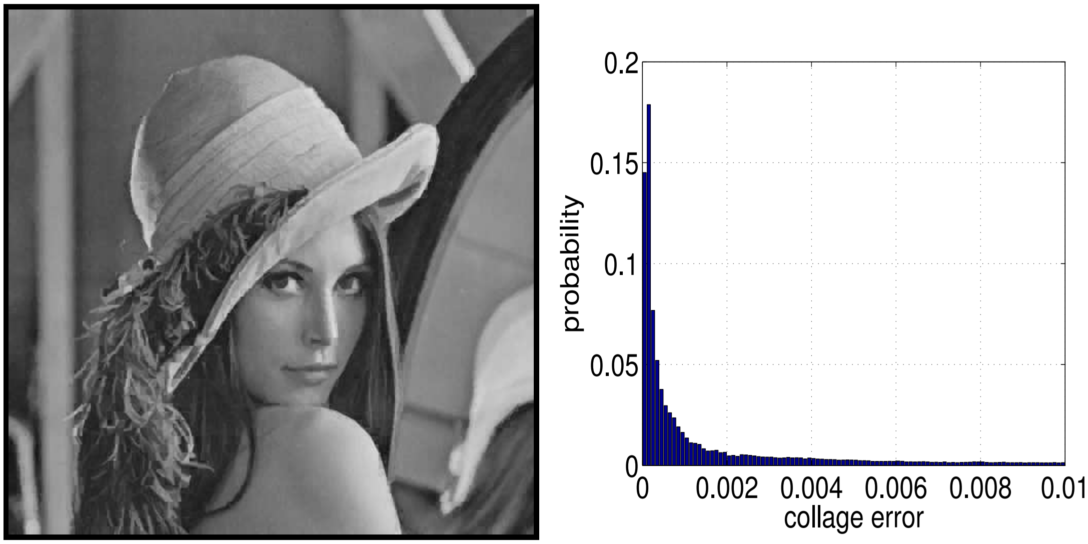
To complete the picture, Figure 3.8 shows the distribution of collage errors for all of the 4096 non-overlapping 8×8 blocks that make up our partition of the image. It is clear that a large percentage of the range blocks are well approximated.



(a) best case

(b) worst case

Figure 3.7: Distributions of collage errors for the “best-” and “worst-fit” range blocks shown in 3.6. All non-overlapping 16×16-pixel domain blocks were used.



(a) fixed point

(b) error distribution

Figure 3.8: Distribution of range-block errors

3.2 Statistics of map parameters

Historically, a major drawback of fractal image coding was the computational time required to search the image for optimal domain blocks. For this reason, a lot of research effort was put into reducing the search time. A variety of strategies were developed, mostly involving some kind of domain pool reduction that was based either on a variance-matching criterion (i.e. low variance blocks only tried against low variance blocks, etc), or some sort of preclassification of blocks based on flatness, gradation or edge orientation. Many of these methods proved effective in practice, and since they are clearly in general finding different domain-range mappings, this success would suggest that there are often at least several good choices of such pairing to approximate the range block.

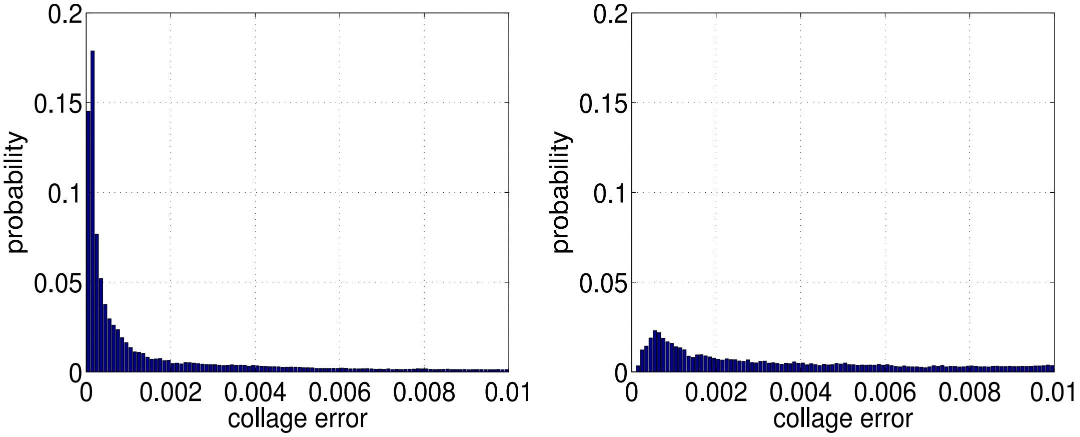
We are not, for the moment, concerned about the mechanics of actually finding such good matches (and hence constructing a contractive 'fractal' operator T, the heart of the fractal coding problem). Rather, let's step back and ask the question: 'How likely is it that portions of an image are well approximated by other portions (under suitable maps)?'

Bear in mind here that this is in a sense the central assertion behind fractal coding as a method of image modelling. Although it is always trivially true that we can find such a map (by partitioning the image with range block of one pixel) this is not interesting. It is clear that for general 'images', i.e. random samples from the configuration space that images live in, it doesn't seem likely that a good fractal coding exists. The assertion being made, then, is that *natural images* tend to have structure that is in some sense self-similar. More specifically, in the sense that allows approximation by the process described here.

This assertion has always lain at the heart of fractal image compression, but does not tend to be articulated, perhaps because of the fundamental difficulty in characterising images. However, *given* that we are making this assumption, it seems reasonable to ask how well it is holding up empirically.

For this purpose, a series of computer experiments were performed on a number of standard images, e.g., *Lena*, *Barbara*, *Mandrill*, *Boat* and *Peppers*. Some results are presented below. One of our main conclusions is that a sizable fraction of range blocks can be well approximated by a good number of domain blocks. It remains to make more quantitative sense of this remark.

In Figure 3.9 are plotted collage-error histogram distributions for the *Lena* and *Mandrill* images. All non-overlapping 8 × 8-pixel range blocks have been considered here. For each range block, all 32 × 32 possible 16 × 16-pixel domain blocks were examined, each with all eight possible isometries, for a total of 33554432 comparisons. As one can see, there is a marked difference between the distributions. The *Lena* image distribution demonstrates a significant peak that lies close to zero-error. In this case, there clearly seems to be a majority of good parent-child matchings. The *Mandrill* distribution is more diffuse, although it does show some peaking near zero-error, but not to the degree of the *Lena* image.



(a) Lena

(b) Mandrill

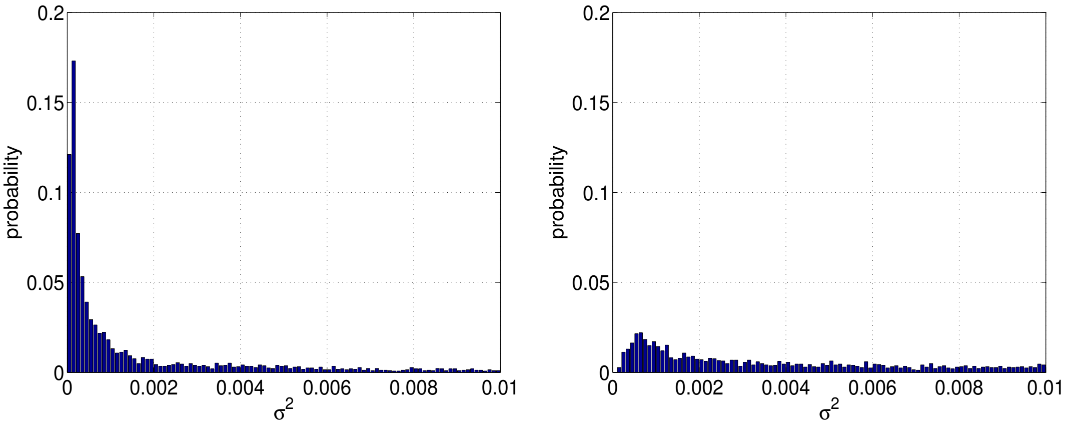
Figure 3.9: Collage error distributions for *Lena* and *Mandrill*

Historically, it has been observed [VS99, Vrs98] that the *Mandrill* image is much harder to compress than most. The usual explanation is that it possesses more high-frequency components or "more activity." Since variance represents one way of characterising activity, it is interesting to examine the distribution of variances of the non-overlapping 8×8 range blocks in the *Lena* and *Mandrill* images as presented in Figure 3.10. The *Mandrill* image statistics show a more even distribution of variances, i.e., a higher proportion of blocks of higher activity, than in the *Lena* image. This concurs with our expectation that low-variance blocks will generally be better approximated than high-variance blocks.

Indeed, the variance distributions of the images in Figure 3.9 demonstrate a significant similarity to their collage error counterparts in Figure 3.9. Such similarity has been observed for other images as well. This should not be totally unexpected since the variance, σ^2, of a block is simply the square of the error in approximating the block by its mean greyscale value, which represents the best L^2 constant fit. This, in turn, corresponds to a special case of fractal coding in which the greyscale coefficients α_i in Eq. (2.77) are "clamped" to zero. As such, one would expect the variance distributions in Figure 3.10 to be some kind of perturbations of the collage error distributions in Figure 3.9. And as the clamping of the α_i coefficients is relaxed (for example, allowing them to take one, two or several quantised values), we expect the resulting collage error distributions to approach the "best-fit" distributions in Figure 3.9.

By comparison, if we look at the distribution of *variances* for these same images, we get the results shown in Figure 3.10.

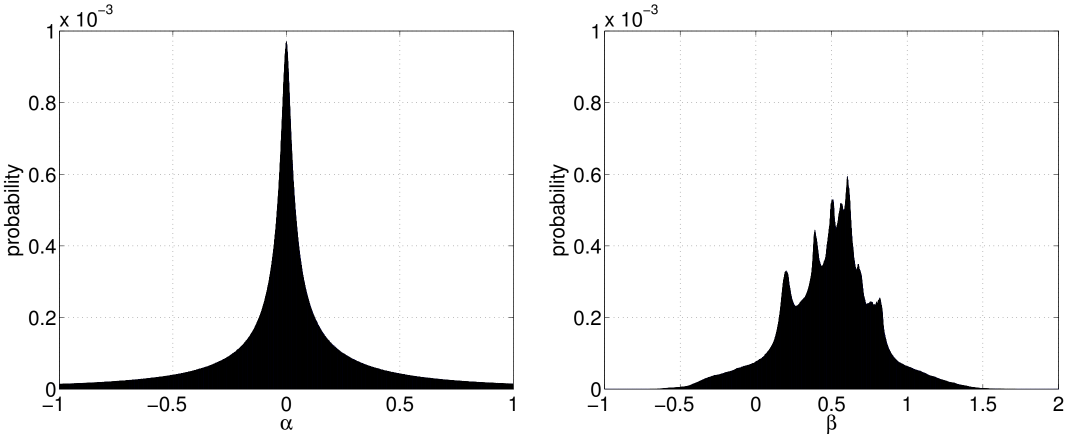
It is also informative to look at the distribution of the parameters α and β. In Figures 3.11 and 3.12 are shown the distributions of α and β greyscale parameters for the *Lena* and *Mandrill* images, respectively. Even though the two images demonstrate quite different variance and collage error statistics, their α and β statistics are quite similar.



(a) Lena

(b) Mandrill

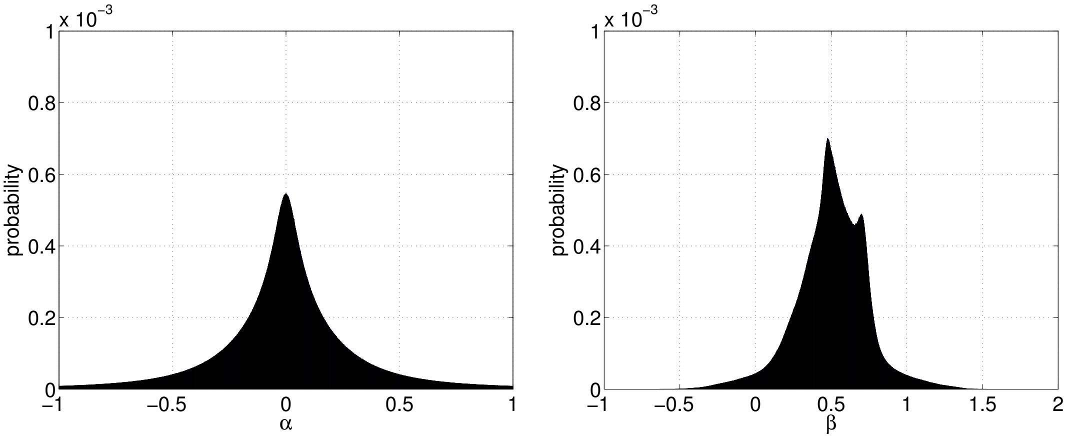
Figure 3.10: Range block variances for *Lena* and *Mandrill* (cf. Figure 3.9



(a) α

(b) β

Figure 3.11: α & β distributions: Lena



(a) α

(b) β

Figure 3.12: α & β distributions: Mandrill

3.3 Effects of noise

In the previous section, some of the statistical properties of the fractal coefficients were discussed, which is interesting in and of itself but does not constitute an 'application'. However, it is also interesting to consider the effect of noise on this process.

Noise in imaging is an important consideration. For the purposes of this discussion, we consider 'noise' to mean the introduction of zero-mean, Gaussian distributed (of some variance σ^2) noise to the pixel values of the image. This is in some sense the simplest case, and is important in practice. A more general discussion of noise in image modelling and processing is beyond the scope of this thesis (see for example [GW02]). We will first consider the effects of noise on the IFSM/IFSW algorithms, which will lead to discussion of denoising applications and beyond.

Figure 3.13 shows (quite typically) the distributions for both noise-free and noisy ($\sigma^2 = 0.0025$) cases, for the Lena image. As is typical for natural images, we see the mass in the distribution has moved away from zero, and is more 'Gaussian' in shape.

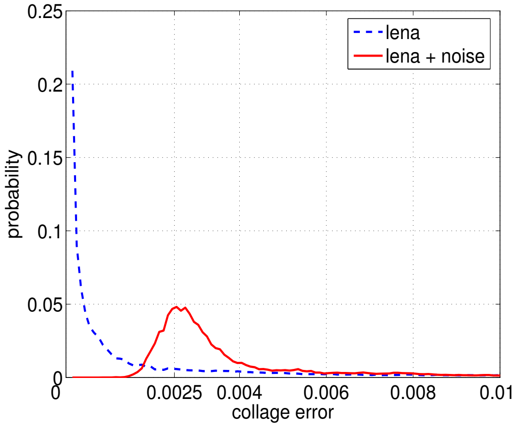
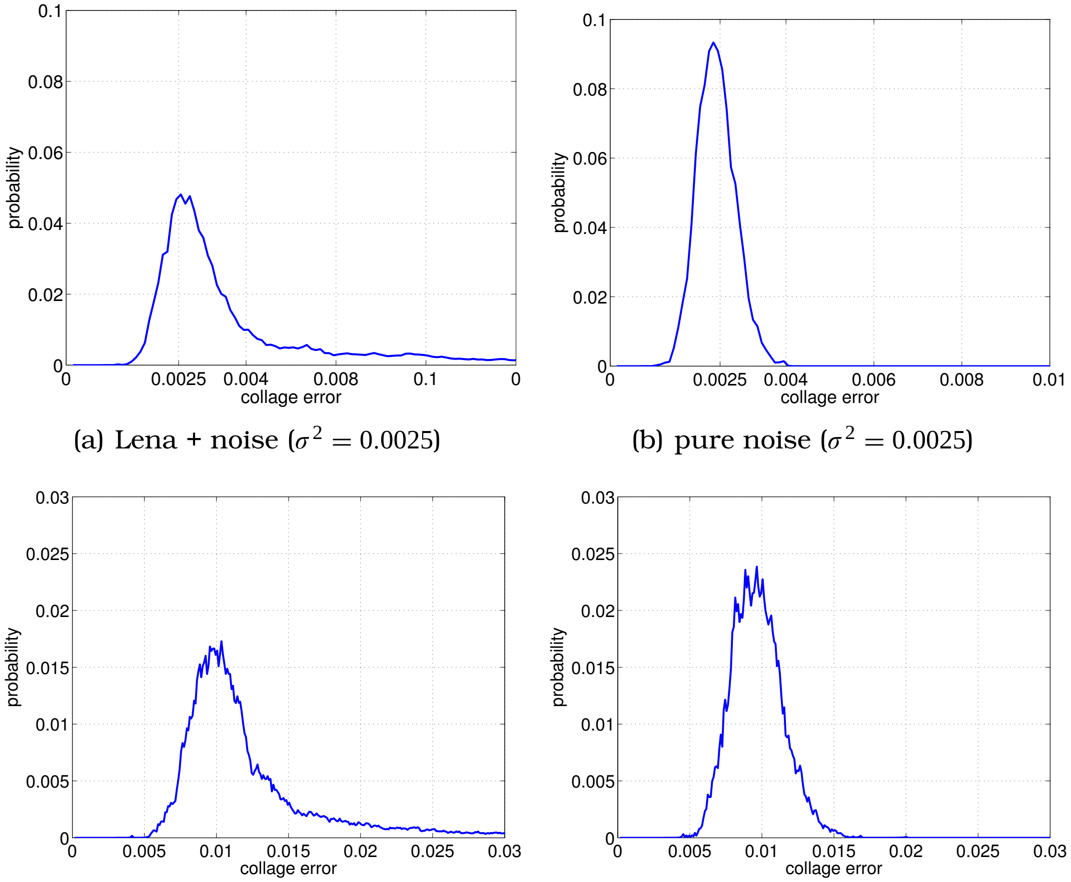


Figure 3.13: Collage error distributions for original Lena and Lena + noise

This is hardly surprising, as a pure noise image will have no preferential fits, and we expect the collage error then to have a Gaussian distribution, centred around the variance of the image. From (2.80) we see that for the white noise case the expected values of $\alpha_n = 0$ and $\beta_n = f(\bar{\xi})$ (the mean of the range block). Hence the expected value of the error is the variance of the image. Figure 3.14 compares histograms of the collage error for exhaustive fit of two different noise variances, and comparison with a 'pure noise' image of the same variance. These results are typical for the natural images we have tried.



(a) Lena + noise ($\sigma^2 = 0.0025$)

(b) pure noise ($\sigma^2 = 0.0025$)

(c) Lena + noise ($\sigma^2 = 0.01$)

(d) pure noise ($\sigma^2 = 0.01$)

Figure 3.14: Collage error distributions for two cases of Lena + noise, compared to collage error when fitting a pure noise image of the same variance.

This suggests a natural role of these histograms as a *variance estimator*. One standard method of estimating the variance of noise added to an image is to histogram the variance measurements over a window of a certain size (say 5x5, or 7x7 pixels) and histogram these results. The mode of this histogram is taken as the estimate. Empirically, using a similar approach with a histogram of the collage error results does not significantly improve on this estimate. The main difficulty is that it is extremely computationally expensive compared to standard methods of variance estimation. Hence the method is worth using if you have already calculated the collage error, but too expensive to justify using as a replacement for standard methods.

The direction looks promising though, so future work on a) modelling the distribution, so as to determine an appropriate curve fit rather than simply taking the mode of the measured histogram, and b) investigating density estimation, so that an estimate can be arrived at with relatively few computations is planned. It is hoped that this will result in a practical method even outside the contexts where one is already doing exhaustive domain-range fits. The intuitive justification for pursuing this avenue is that by using the self-similarity 'information' in the image, one can ameliorate estimation error due to the natural variation of the image.

As would be expected, noise also effects the distribution of the parameters α and β. Figure 3.15 shows a side by side comparison of these parameters (and collage error) for the original Lena image and Lena with noise of variance 0.01 added to it. Recall that all such values are given for normalized images, where the intensity values lie in $[0, 1]$ (rather than discrete $0 \ldots 255$ grayscale values, for example). Again, these results are typical. The histograms of α becomes more heavy-tailed, and the characteristic peaks in β are smoothed out. Recalling that these peaks represent particular intensity values that show up in the image, it is natural that perturbing the intensity values in this way would smooth out these peaks. Again the α value are clipped at ± 1, and these figures ignore the clipped values. The final pair of figures shows collage error histograms, showing the extent to which mass is concentrated near 0 in the noise-free case.

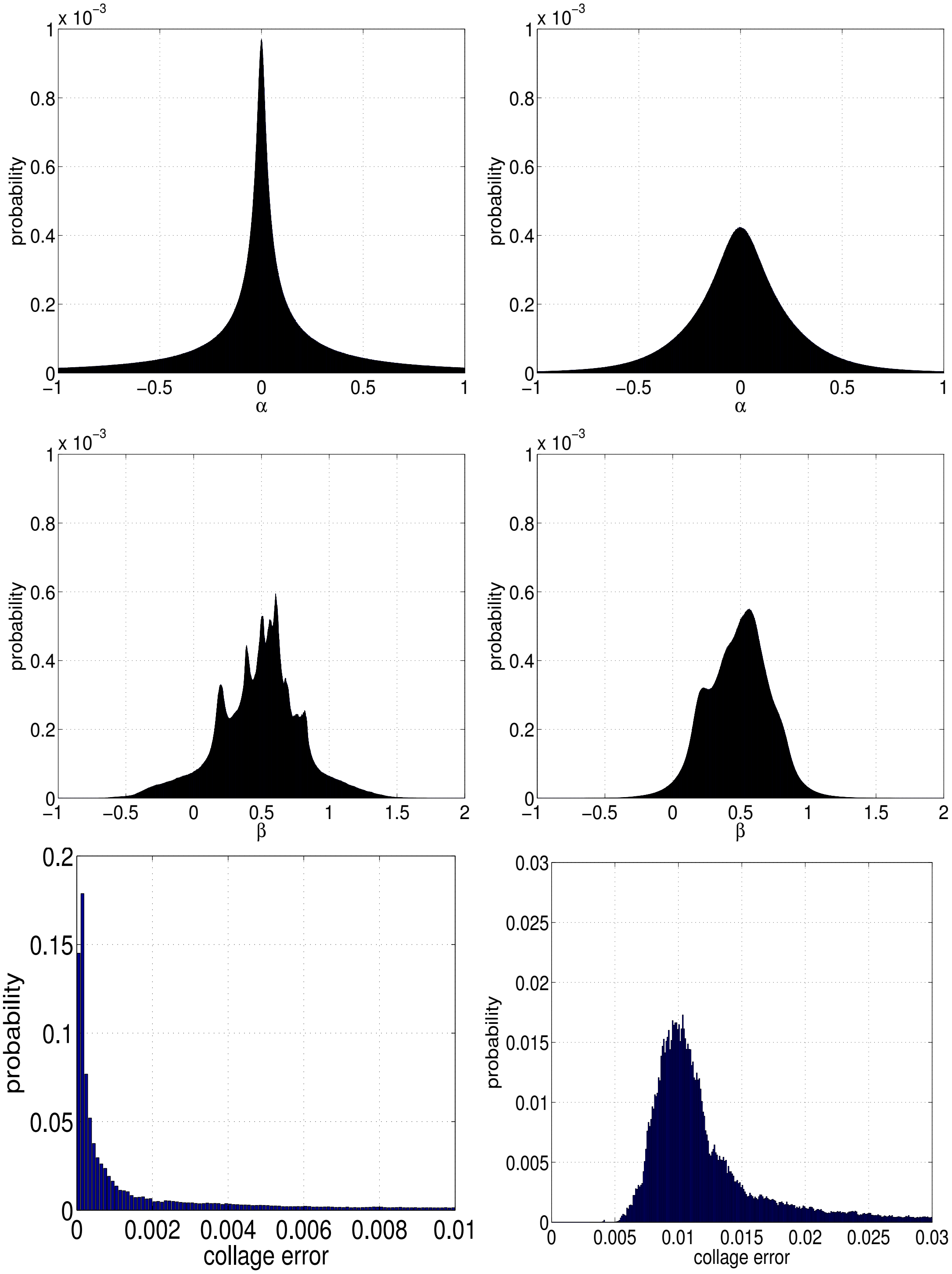
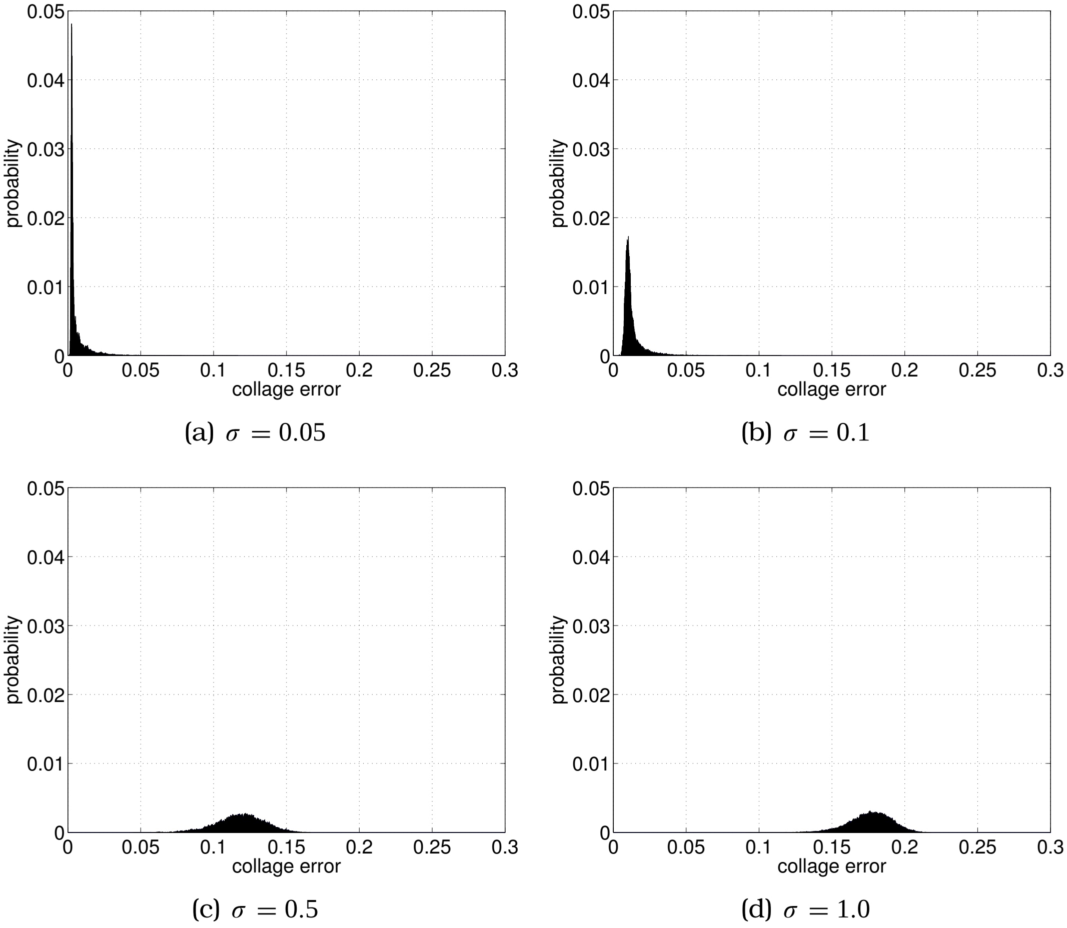


Figure 3.15: α, β, and collage error histograms for Lena (left column) and Lena + noise ($\sigma^2 = 0.01$)

Things are somewhat complicated by larger variances, because of the discrete nature of the images. With larger variance noise added, the pixel values are more likely to end up outside the range $[0, 1]$ (or 0 to 255 in discrete representation) and will be clipped at these values. However, when this happens the distribution of pixels is no longer Gaussian! We see from Figures 3.16 and 3.17 that the collage error histograms for Lena + noise and pure noise images are still comparable, but modelling this distribution would be much more difficult because of the underlying heavy-tailed distribution of pixels given many black (0) and white (255) pixels.



(a) $\sigma = 0.05$

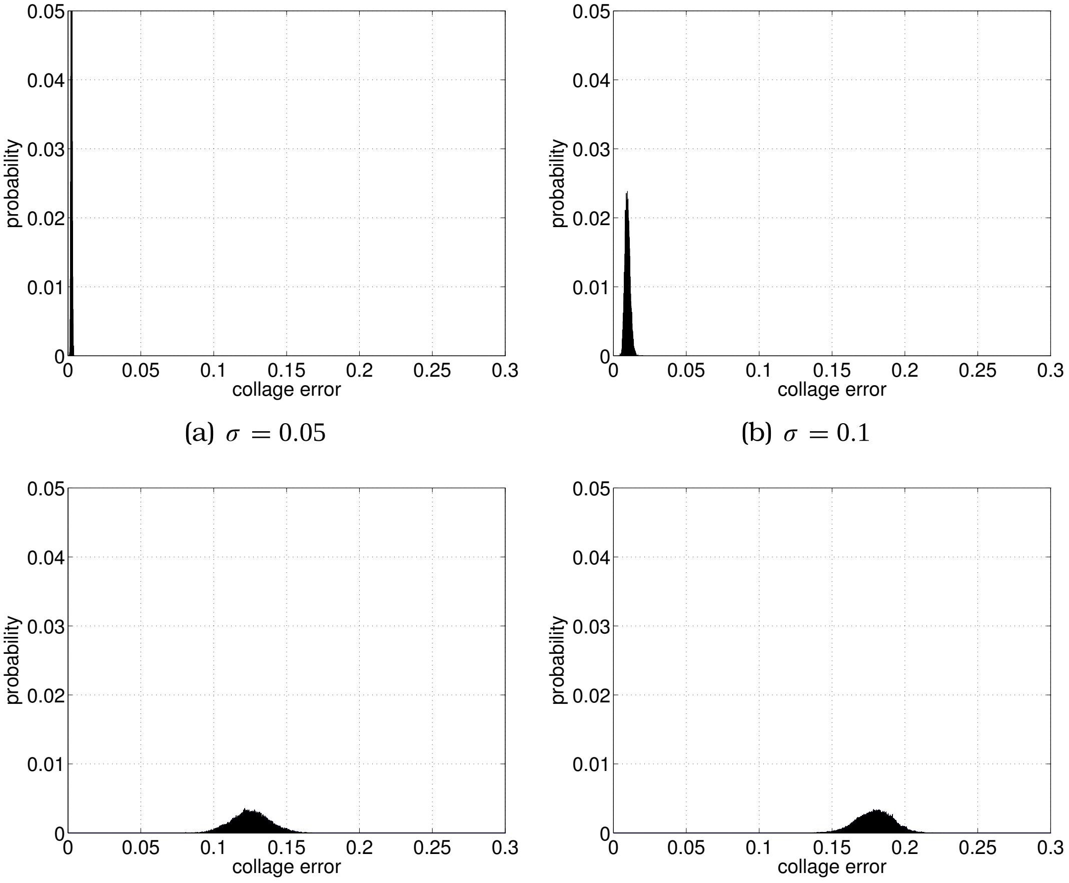
(b) $\sigma = 0.1$

(c) $\sigma = 0.5$

(d) $\sigma = 1.0$

Figure 3.16: Collage-error distributions for *Lena* images with added noise of increasing variance (cf Figure 3.17).

Also included, as Figure 3.18, is a plot of the histograms of α and collage error (there is no β parameter in the IFSW case) for both Lena and Lena + white noise. Again these are representative images. It is interesting to note here how much different the α distributions are.



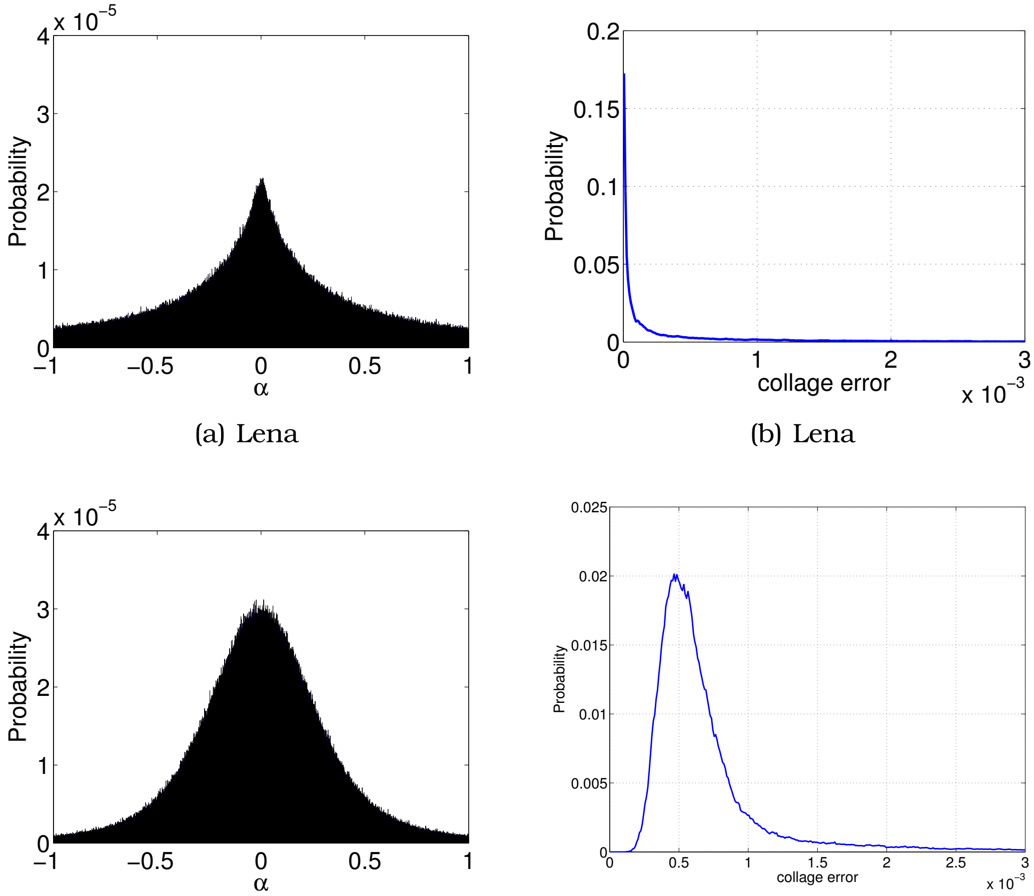
(a) $\sigma = 0.05$

(b) $\sigma = 0.1$

(c) $\sigma = 0.5$

(d) $\sigma = 1.0$

Figure 3.17: Collage-error distributions for pure noise images images of increasing variance (cf Figure 3.16).



(a) Lena

(b) Lena

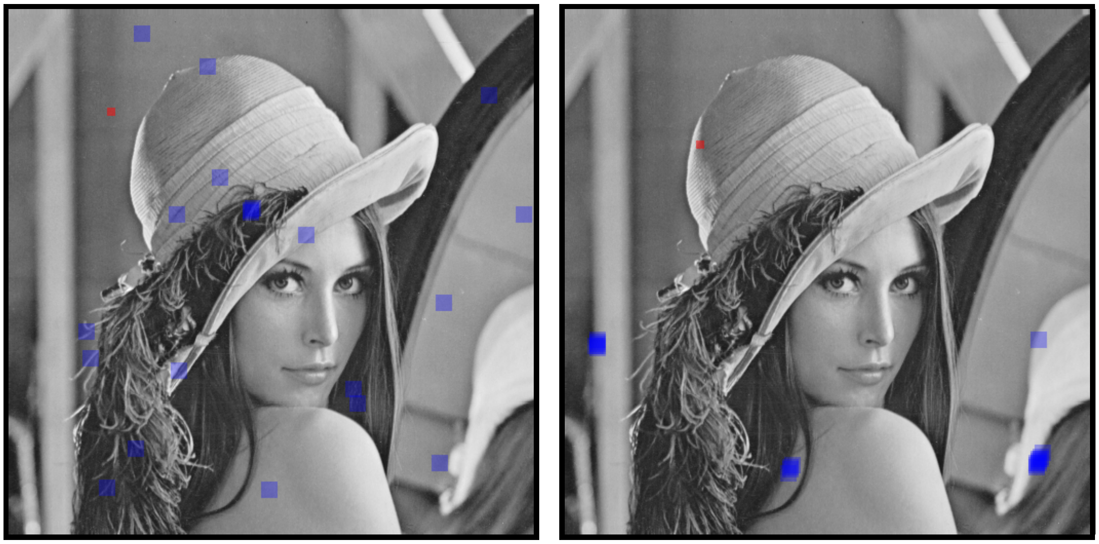
(c) Lena + $\sigma^2 = 0.01$ noise

(d) Lena + $\sigma^2 = 0.01$ noise

Figure 3.18: IFSW parameters for Lena and Lena + noise. Note that for the α the probability axis are not the same, there is *much* less mass in the noisy case (more cases were clipped at ± 1. These histograms calculated for level 6 to level 7 domain-range mappings (corresponding to 4th to 5th level of the wavelet decomposition.

3.4 Multiple Parents

From the previous discussion (e.g. Figure 3.8-b) it is clear that in most cases (given our assumptions about natural images) there are many 'good' fits available for a give range block. It is therefore interesting to consider the distribution of 'good parents' for a given child block. Figure 3.19 shows the best 10 matches for two different children. The interesting thing to note here, is that characteristic clustering of 'good parents' for an *edgy* child, meaning that the child block contains an edge.



(a) smooth child

(b) "edgy" child

Figure 3.19: The best ten parent blocks for two given child blocks

One obvious question then is whether or not we can improve the fixed point estimation of the map T by using information from more of these 'good parents'. As we are working under a norm, we can define 'multiple parent maps' as convex combinations of contractive maps (in this case affine, but it generalises). This follows from the triangle inequality and the contractivity of the constituent maps.

Corollary 3.1 (multiple affine map contraction) *Suppose we have* *two contractive maps $f_0(x)$, and f_1 with contractivity factors c_0, c_1, respectively. Then the convex combination $f(x) = af_0(x) + (1-a)f_1(x)$ is also contractive with factor $c = \max(c_0, c_1)$.* □

PROOF

$$\begin{aligned}
\|f(x) - f(y)\| &= \|af_0(x) + (1-a)f_1(x) - (af_0(y) + (1-a)f_1(y))\| \\
&= \|a(f_0(x) - f_0(y)) + (1-a)(f_1(x) - f_1(y))\| \\
&\leq a\|(f_0(x) - f_0(y))\| + (1-a)\|f_1(x) - f_1(y)\| \\
&\leq c|x - y|
\end{aligned}$$ ■

As this generalises to more than two maps, we can consider (in comparison to (3.2)) the *multimap ifsm*:

Definition 3.1 (Multimap IFSM)

$$(Tu)(x,y) = \sum_i \sum_j c_j \phi_{i,j}(u(w_{i,j}^{-1}(x,y))) \qquad \sum_j c_j = 1 \tag{3.3}$$ □

So we can make contractive maps of this type, but are they better?

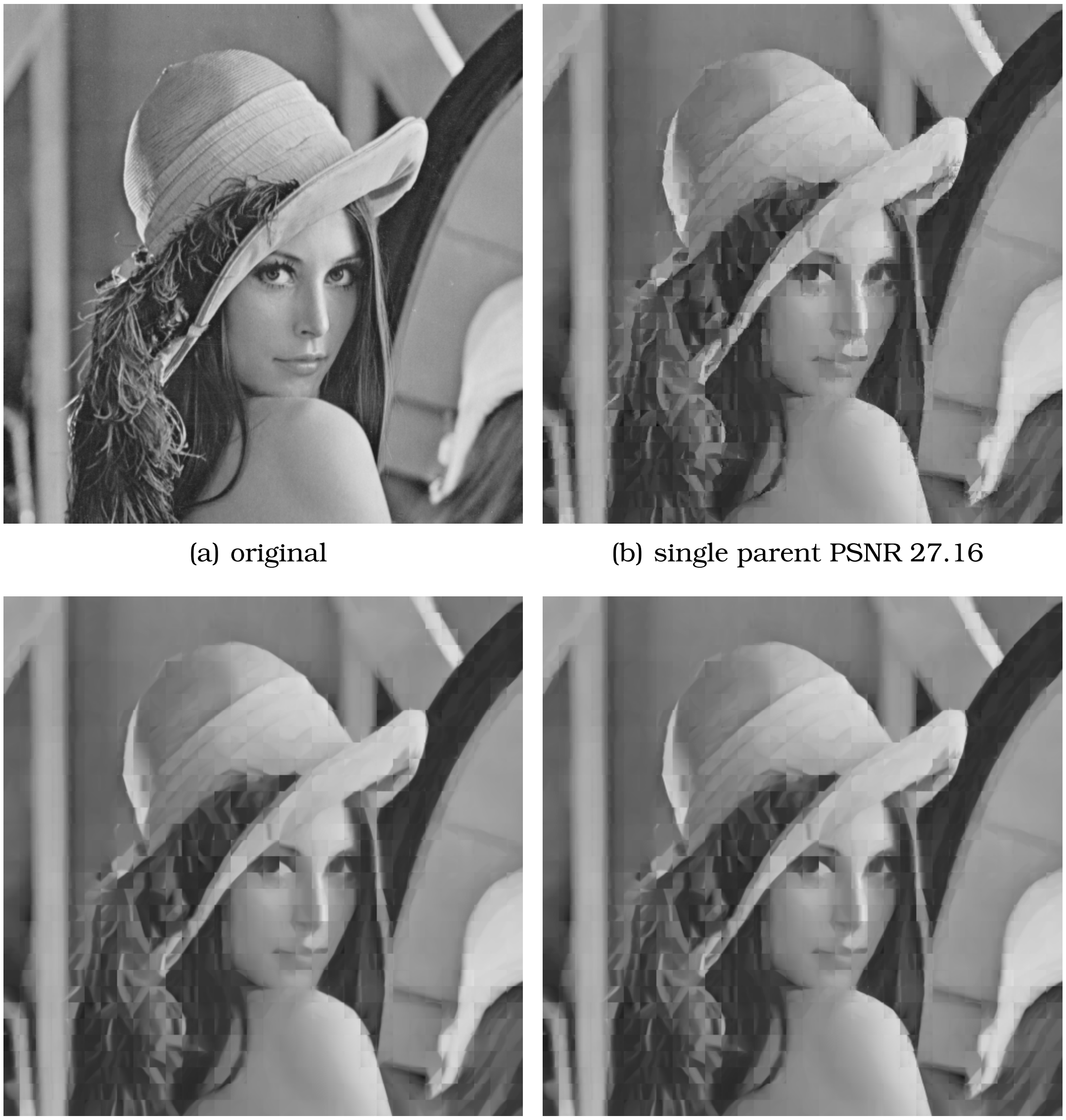
Analytically, it is difficult to say much. To make an improvement in the collage error, this approach implicitly assumes that the estimation errors for the blocks are decorrelated (hence averaging them is likely to reduce error). This, however, is a statistical property of the class of images we consider. For a particular image, then, we could optimise the weights to arrive at a better approximation. However, from an applications point of view, this is not so interesting. In general, except for applications like compression, we won't know the 'true' image. The particular example discussed further in this section will be image denoising — where we do not want to optimise to a better fit of the noise, rather we want to impose an (implicit) image model with less noise (i.e. more smooth, *except* where it shouldn't be...). For this reason we concentrate only on data independent methods of weighting the maps.

Empirically, this approach is quite effective. Recalling the difficulty with defining rigorously what a 'better' image means, consider the two shown in Figure 3.20. Here a coarse partition is chosen to emphasise the 'blocking artifacts' common to IFS approaches. These occur because errors are correlated to the partition structure, in this case a grid. Not only has there been a significant (0.5db) improvement in the PSNR of this estimate in the multiple map case, but the edges are more clearly defined and smooth appearing. The convex combination was made of ten maps. Two weighting strategies are shown, both a simple average and a weighting based on a power of the inverse of the collage error (normalised to form the convex combination) in order to emphasise better matches. Note that the collage errors here tend to be quite small, all bounded above by 1.

This last represents a heuristic approach to dealing with the problem of how many parents to choose, and how to weight the contributions. By weighting by the inverse of the error, we emphasise the better matches. Taking a power of this with further weight towards good maps. This way, if a large number of the parents chosen have large error compared to a few 'good fits', the contribution of the 'bad fits' will be negligible.

Figures 3.21 and 3.22 show (successive) 'zooms' into a section of Lena's shoulder. We see that the multiparent maps vastly improve the accuracy of this strong, isolated edge. The accuracy is really quite impressive, given the coarse 16×16 partition. We also see that by strictly averaging the maps, we tend to 'smear out the line' more than when the better matches are emphasised. There is a gain of about 0.04 dB, also, which is not very significant. However, the visual improvement is noticeable in contrast across edges, at least on this area of the image.

So why is this working? Without *a priori* assumptions on the maps and image, there isn't much we can say. The collage error on the multiple maps will be bounded below by the single map case. Empirically, then, it is clear that these results suggest that the error in the



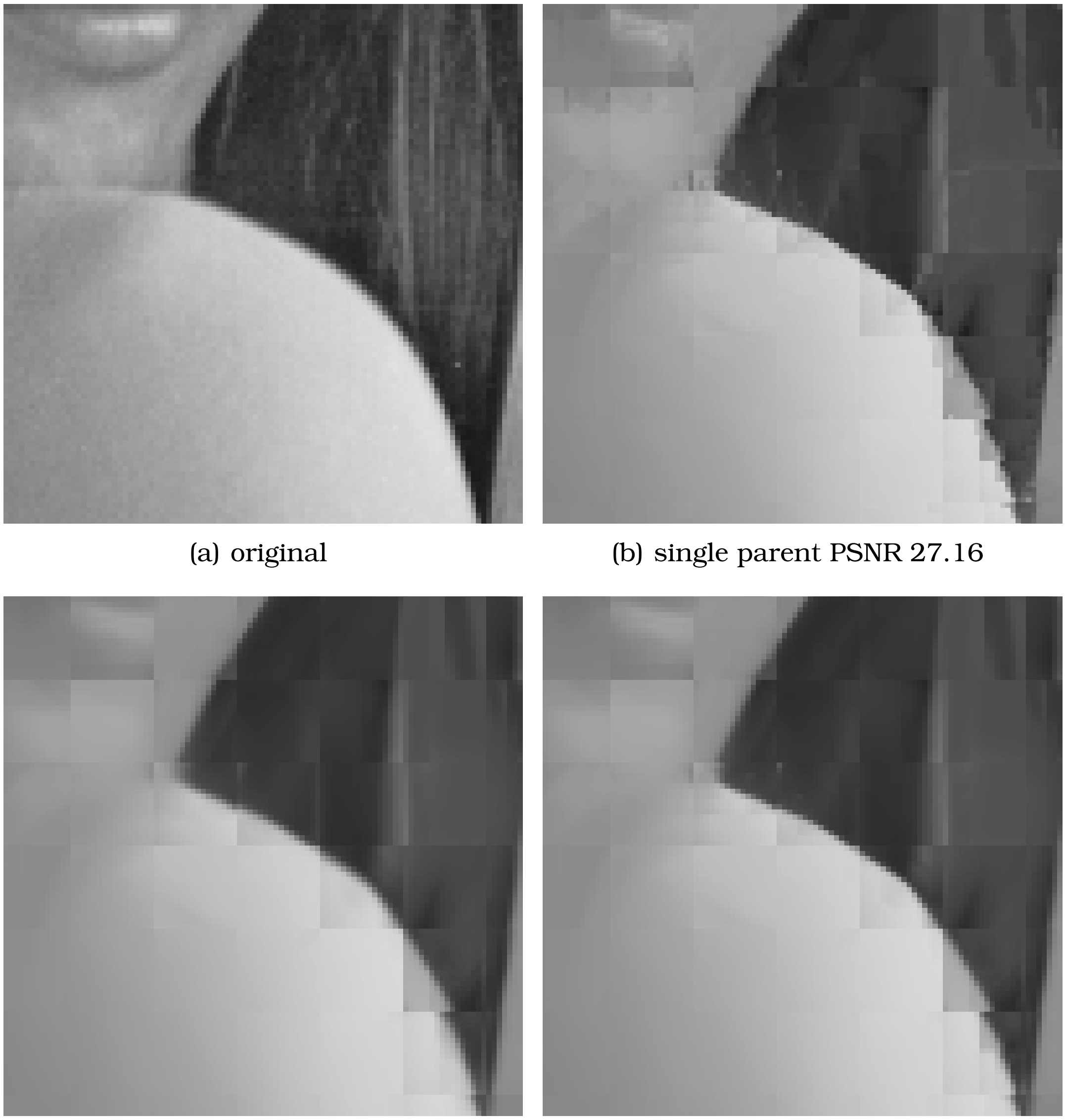
(a) original

(b) single parent PSNR 27.16

(c) multiparent averaged PSNR 27.64

(d) multiparent weighted PSNR 27.68

Figure 3.20: Lena partitioned into 16×16 pixel range blocks. a) original b) estimated by a conventional IFS scheme c) averaged result over 10 parent blocks d) weighted average (as $1/err^4$) of 10 parents.



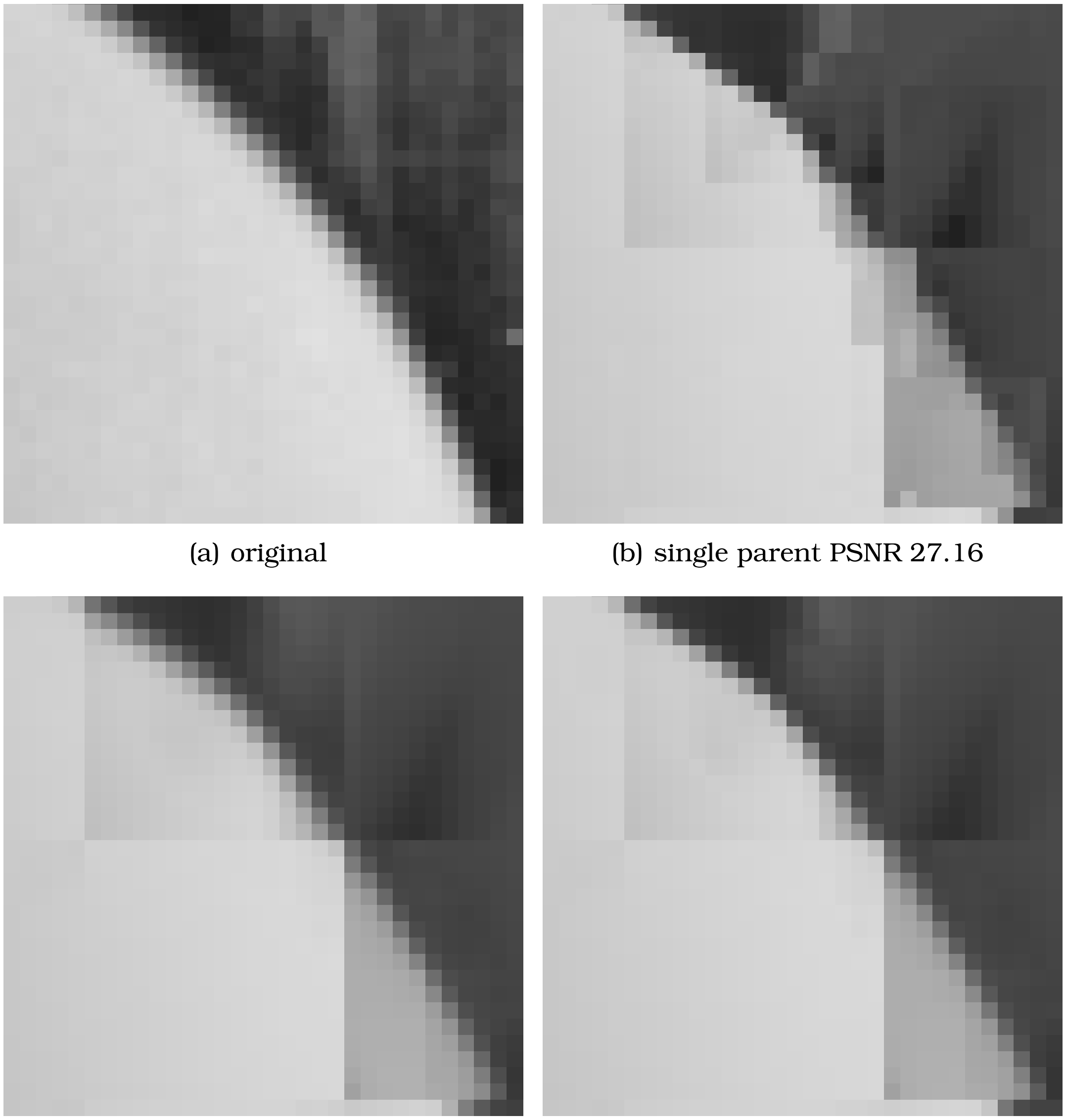
(a) original

(b) single parent PSNR 27.16

(c) multiparent averaged PSNR 27.64

(d) multiparent weighted PSNR 27.68

Figure 3.21: Zoom views of Figure 3.20



(a) original

(b) single parent PSNR 27.16

(c) multiparent averaged PSNR 27.64

(d) multiparent weighted PSNR 27.68

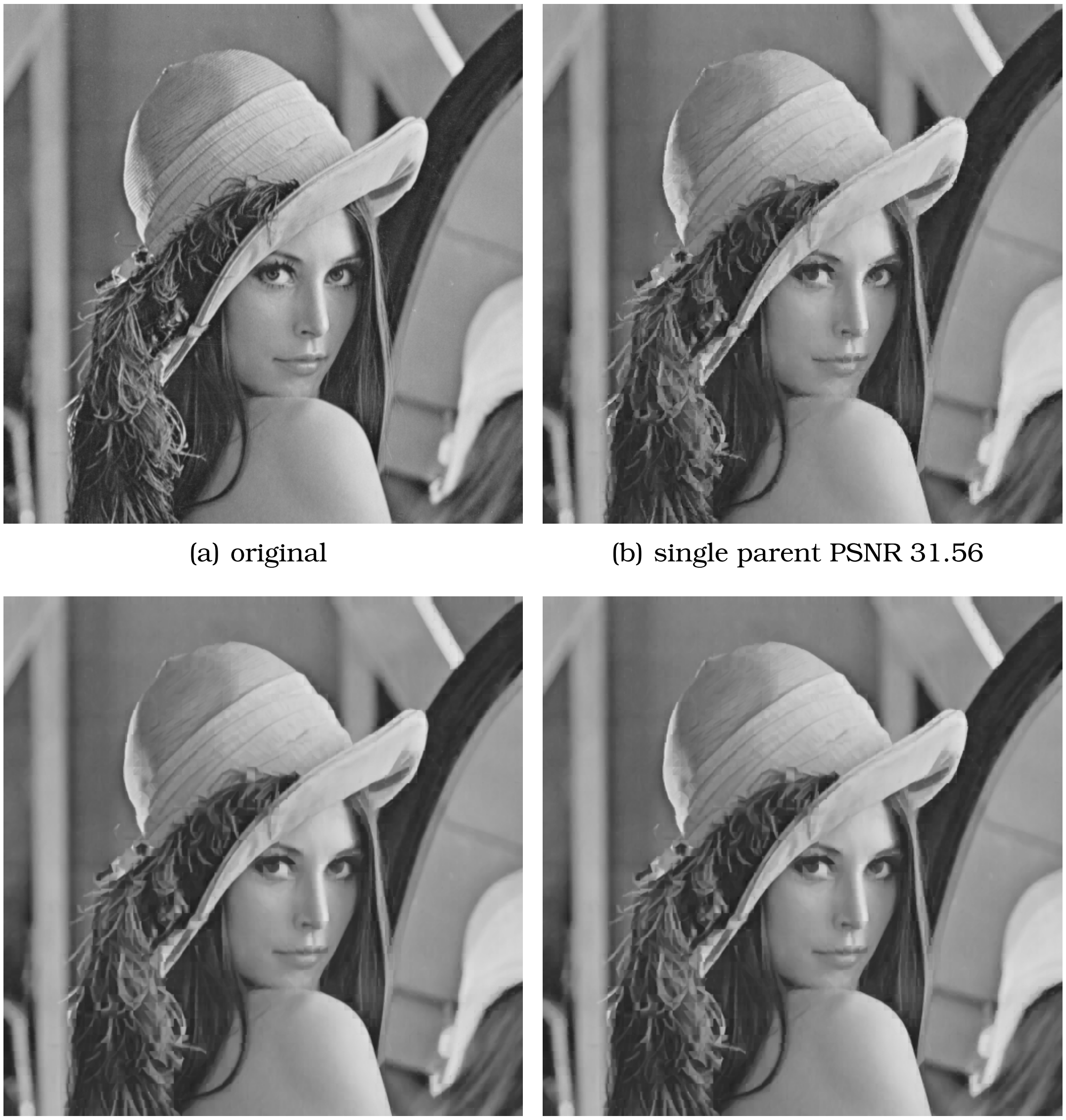
Figure 3.22: Small zoom views of Figure 3.20 (same location as Figure 3.21)

estimates of the range blocks is uncorrelated, hence averaging them will perform a sort of 'variance reduction'.

It is less clear why the weighting scheme above, of imposing a weight based on an accelerated decay of the error measurements, will improve the results.

Obviously such a coarse partition is causing quite noticeable artifacts. Similar results are shown for a more accurate partition into 8 × 8 range blocks in Figures 3.23 and 3.24. Here we see again a significant but not huge gain, of nearly 1dB between the single parent IFS and the multiple-parent case. Again the gain between simple averaging and the weighting schemes is relatively small, about 0.13 dB, but visually apparent.

These images emphasise the performance on a relatively clear, strong edge. In more complicated regions of the image things are less clear. Figure 3.25 shows a different region of the images, where the complexity in Lena's hair makes fitting difficult. In some cases of complicated regions of the image, the 'false texture' introduced by estimation error in the the single map IFS may be visually appealing. This is similar to the idea of *fractal zoom*.



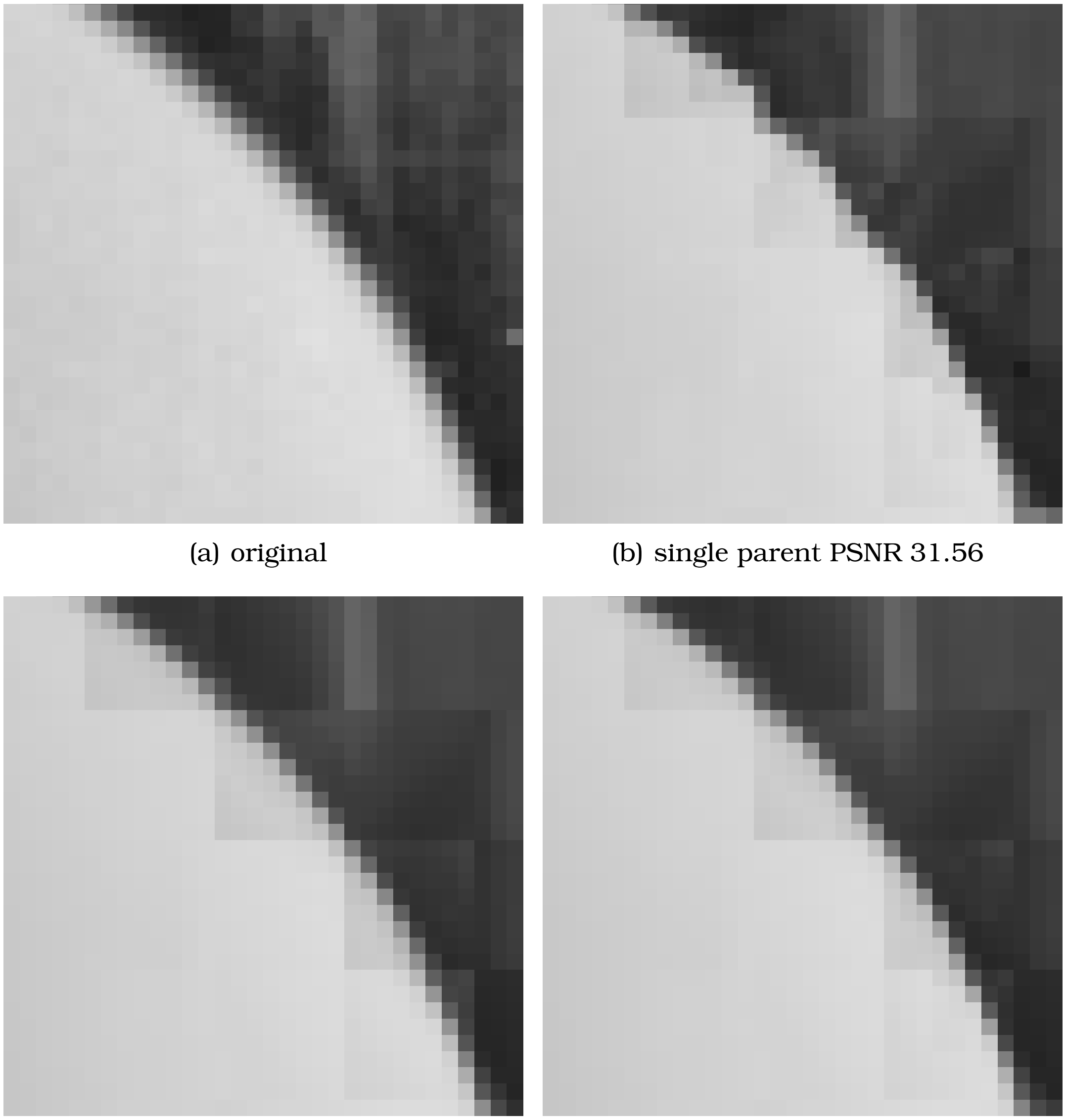
(a) original

(b) single parent PSNR 31.56

(c) multiparent averaged PSNR 32.23

(d) multiparent weighted PSNR 32.36

Figure 3.23: Lena partitioned into 8×8 pixel range blocks. a) original b) estimated by a conventional IFS scheme c) averaged result over 10 parent blocks d) weighted average (as $1/err^4$) of 10 parents.



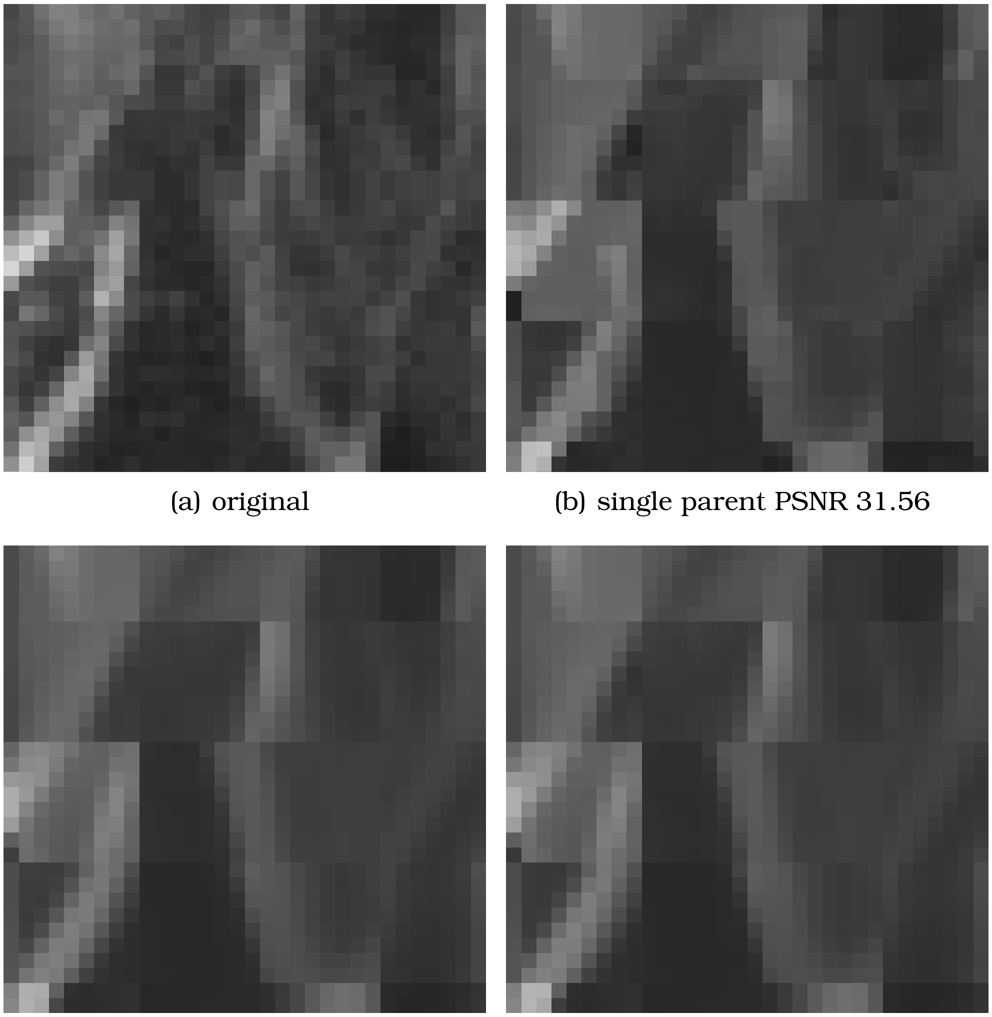
(a) original

(b) single parent PSNR 31.56

(c) multiparent averaged PSNR 32.23

(d) multiparent weighted PSNR 32.36

Figure 3.24: Zoom views of Figure 3.23



(a) original

(b) single parent PSNR 31.56

(c) multiparent averaged PSNR 32.23

(d) multiparent weighted PSNR 32.36

Figure 3.25: Another region of Figure 3.23

3.5 Multiparent Maps and Noise

Now, what happens when there is noise added? We repeat the process for a version of *Lena* with noise of variance $\sigma^2 = 0.01$ added. Figure 3.26 shows (like Figure 3.19) the best parents for two blocks. The interesting thing here is to note that typically the distribution of parents for an 'edgy' child is much more robust with respect to the introduction of noise than a 'smooth' child. In other words, the set of best parents in the noisy case is most likely to contain some of the same parents as the noise-free case if the child contains an edge. Intuitively this is reasonable.

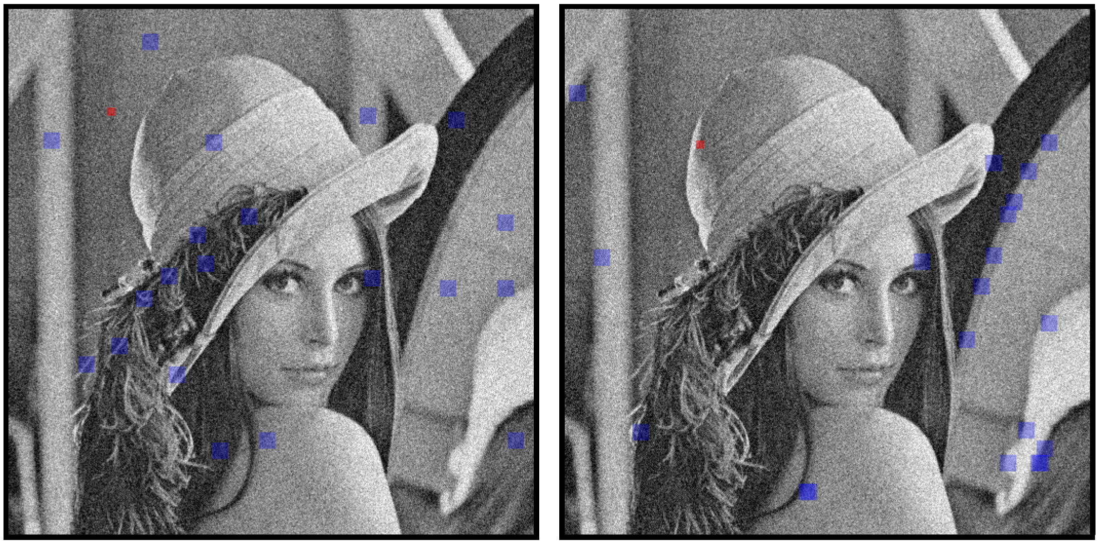
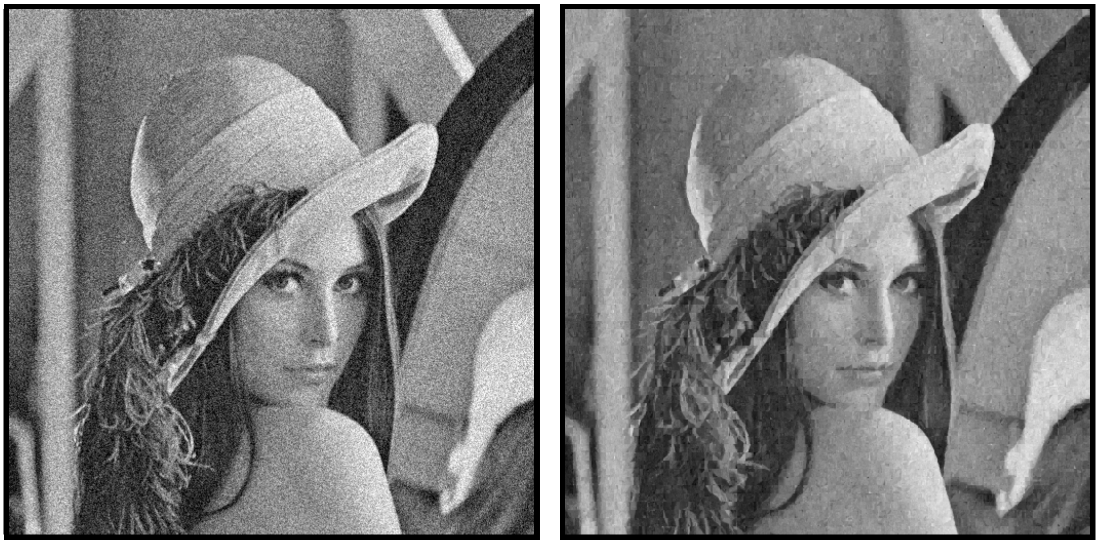

(a) smooth child (b) "edgy" child

Figure 3.26: Best few parents with noise of variance $\sigma^2 = 0.01$ added

It has long been known that IFS coding is naturally denoising (see, for example, discussion in [GFV03]), as we can see from Figure 3.27 where the noisy Lena is compared to the fixed point of an IFS constructed from this version. The PSNR of each relative to the original image is given. This is, in part, due to the 'variance reduction' resulting from spatial averaging of the noise. If we have a white noise



(a) $\sigma = 0.01$ PSNR 20.0535

(b) PSNR 26.6731

Figure 3.27: Single parent denoising

process w_i and construct from it an averaged version, $\sigma \Rightarrow \sigma/4$.

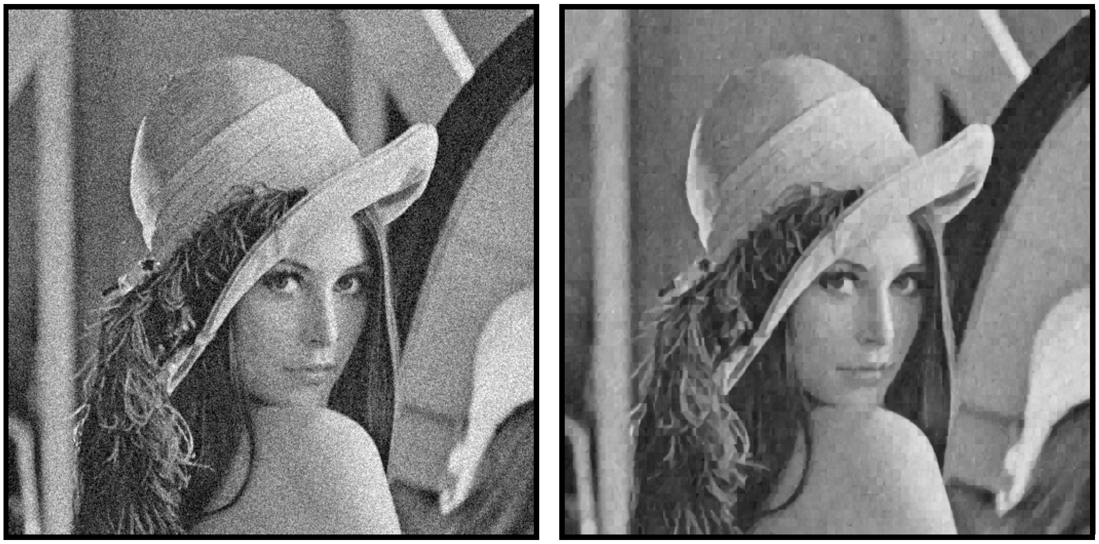
Because of the nature of the ℓ^2 fit we are doing, it is clear that the high-frequency components introduced by additive Gaussian white noise will be more difficult to fit. This enhances the smoothing effect when the image in question does contain a lot of low-frequency information (clearly this would not be the case with, say, a pure noise image).

The above, along with the previous determination that for most range blocks in a natural image we (empirically) expect to find many ‘good’ parents, leads to a simple but intriguing idea.

A standard technique for ‘noise reduction’ in physical systems with additive white noise is to average multiple samples. So long as you are able to in some sense draw repeated samples (for example, you believe your measurement system is stable enough to be measuring essentially the same spatial area multiple times) then averaging the output will reduce the variance of the estimate.

Analogously, if the image of each parent block is a good estimate of the child block plus noise, we can attempt a multiple parent noise reduction scheme. It follows from the triangle inequality and the contractivity of the maps that in this setting we may apply convex combinations of contractive maps to arrive at a contractive map.

So a simple average could be used, or *as in the following results* a weighted average where the weights are generated from the relative collage error, as discussed in the previous section. It is unclear what the correct number of parents to take would be. By weighting according to the collage error, the effect of too large a set of parents should be ameliorated. Figure 3.28 shows the result of such a scheme applied to the same image as Figure 3.27.

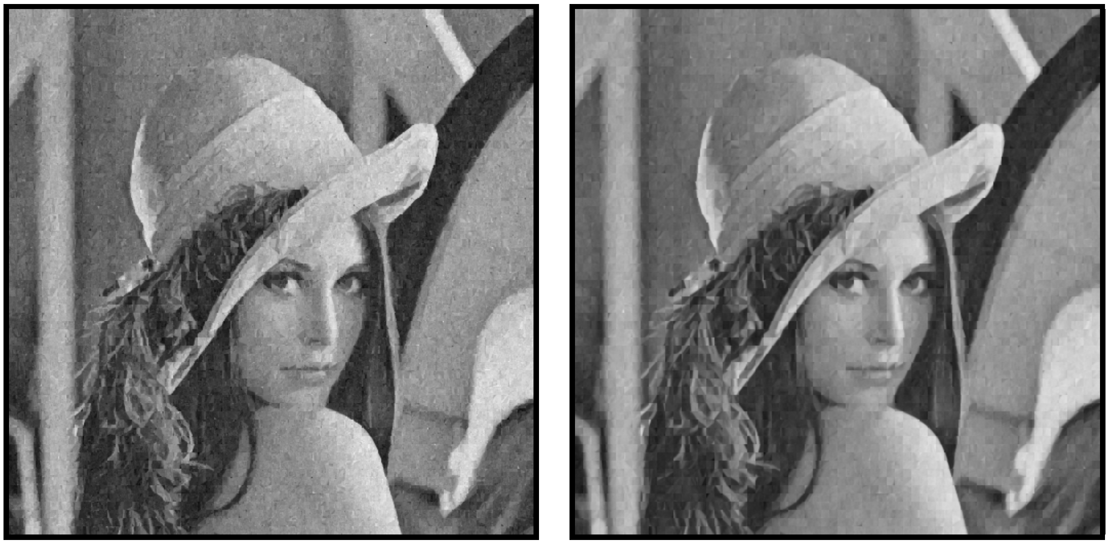


(a) $\sigma = 0.01$ PSNR 20.0535

(b) PSNR 29.0326 (10 parents)

Figure 3.28: Multiparent denoising

There is a gain of roughly 2.5 dB. Of course, as previously noted, PSNR is an insufficient measure of 'image quality'. It is arguable that the gains are better than PSNR alone suggests. To move this statement beyond the anecdotal, however, would require a carefully designed study. Figure 3.29 shows the two fixed points side by side for the purpose of visual comparison. It should be emphasised that this gain is available at no significant additional computational cost.



(a) PSNR 26.6731

(b) PSNR 29.0326 (10 parents)

Figure 3.29: Single parent (a) and Multiple parent (b)

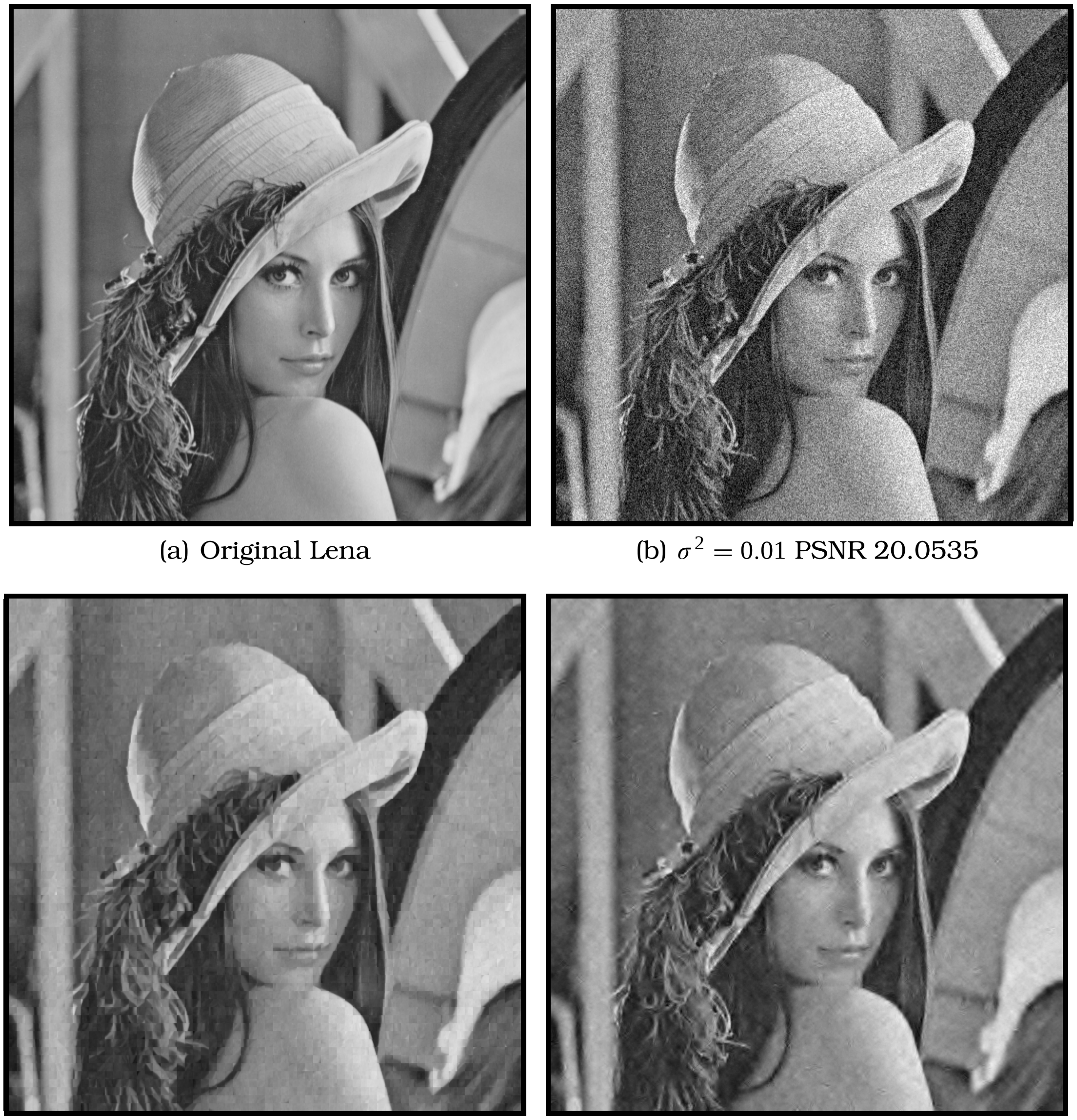
The approach can be generalised in a straightforward way to the wavelet domain, as described in §2.4.3 again with multiple parents. PSNR Results are typically slightly improved from the spatial case. Beyond what PSNR can show, artifacts in spatial IFSM tend to expose the block structure of the partition. Since the HVS tends to respond strongly to straight lines and blocky structures, even with equivalent PSNR results, the wavelet approach may 'look better'. Artifacts due to approximation error in the wavelet coefficients will show up as a 'ringing' or Gibbs phenomenon [Mal98, Dau92]. Figure 3.30 demonstrates this.

Finally, a comparison with another (wavelet based) denoising method (using a Hidden Markov Model (HMM) [RCB99]) is offered in Figure 3.31, showing a slight improvement in performance. The wavelet scheme is based on heuristics to decide re-partitioning criteria (i.e. what level of tree to use) and number of parents. Further analysis and experiment may reveal more satisfying criteria. This would require study of a large corpus of natural images and noise.

Again, we note that what has been presented are representative images. Relative performance on other images is typically similar, within the constraints of a fairly small testing set (on the order of 10 images).

A note on partitioning

All of these examples have used a straightforward grid partitions of the image, which have obvious limitations in terms of arbitrary boundaries, rather than using any image information. A thorough investigation of partitioning for IFS (e.g. quadtree, horizontal-vertical, triangular meshes) in the presence of noise would be interesting, but is beyond the scope of this thesis.



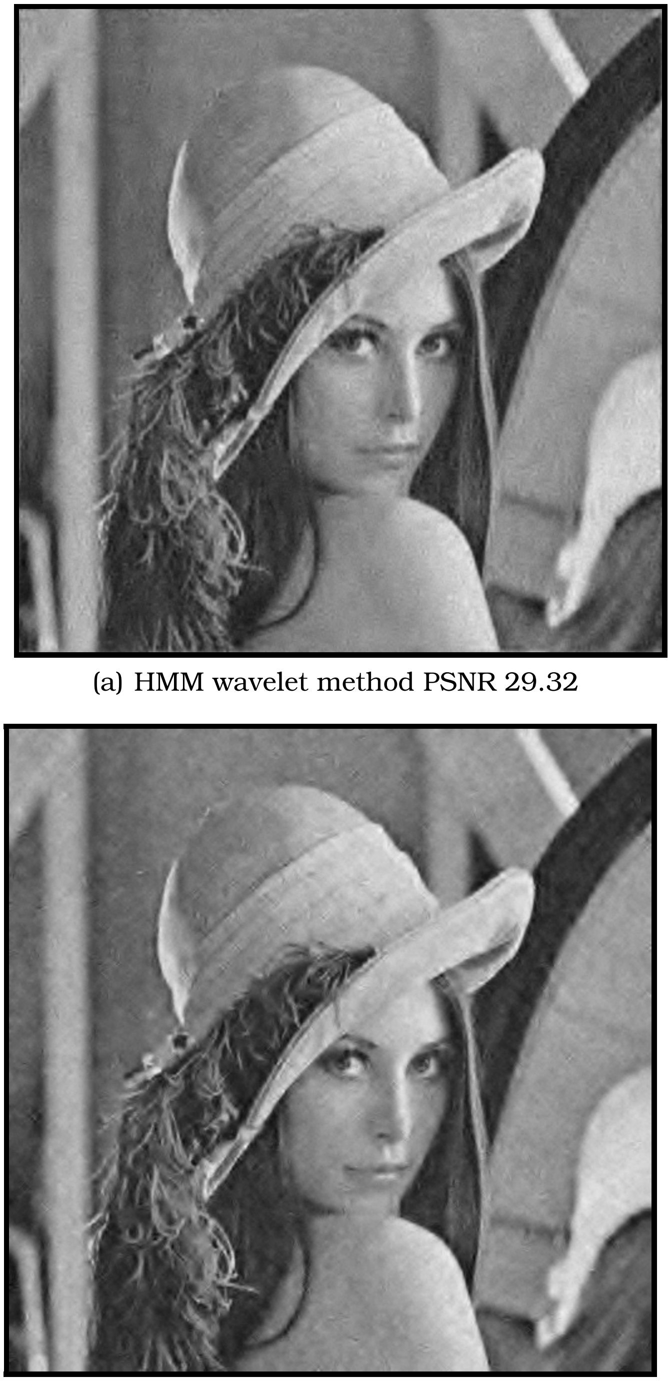
(a) Original Lena

(b) $\sigma^2 = 0.01$ PSNR 20.0535

(c) IFSM PSNR 29.03 (10 parents)

(d) IFSW PSNR 29.49 (15 parents)

Figure 3.30: Original (a) and noisy (b) Lena images. Denoised versions using (c) spatial IFSM and (d) wavelet IFSW.



(a) HMM wavelet method PSNR 29.32

(b) IFSW PSNR 29.49 (15 parents)

Figure 3.31: Comparison of the multi-map IFSW method with a HMM [RCB99]

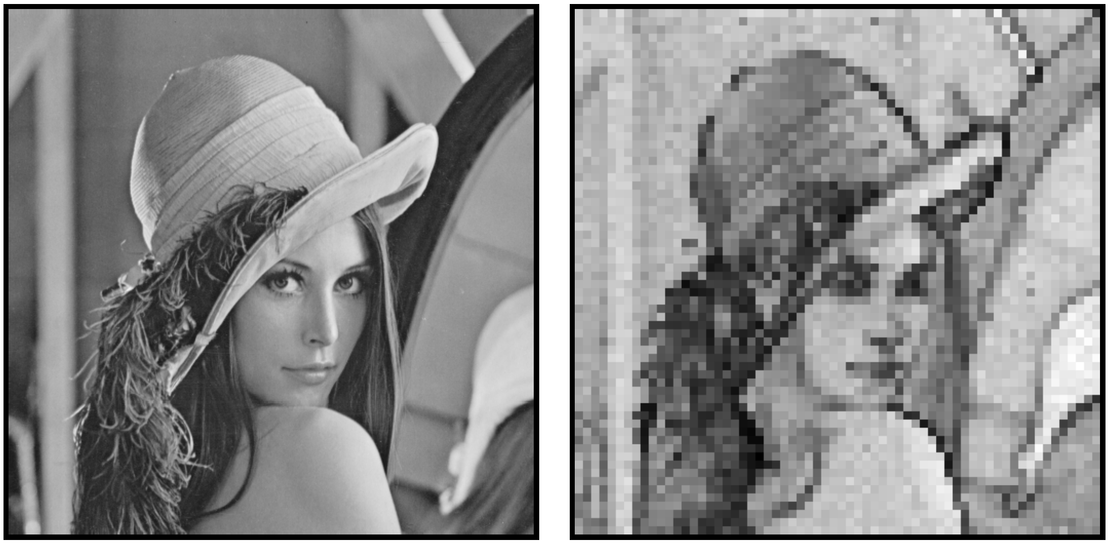
3.6 Fit variation and edges

Another interesting approach is to consider the information that the fitting process tells us about individual range blocks. If, as we have seen, range blocks containing edges are more difficult to fit then conversely the difficulty of fitting a block tells us something about its makeup.

What follows is a description of quite preliminary ideas. The area of classification has a rich literature, and a large amount of space could be spent on this aspect alone. In this thesis, however, we will present some ideas implemented with only very naive 'classification', simple because they illustrate some interesting aspects of the IFS process.

In particular, this leans toward *data driven* analysis of images. In other words, we are suggesting one way of asking 'what is the image telling us about its own makeup'. It is felt that this direction is necessary to attack some of the more difficult open problems in image processing.There is every reason to expect that with careful improvement of the technique described here, performance can be greatly improved – and the way opened for several other application areas (e.g. denoising) that might benefit from an accurate representation of the location of edges.

Figure 3.32 shows the Lena image, and an 'edge' image, v. We construct v by associating with the pixels in each range block (hence the blockiness of Figure 3.32-b) a suitably normalised measurement of the variance of the collage error when comparing this range to all possible domain blocks. Hence high variance suggests a 'difficult fit', which we conjecture is associated with edges (this seems borne out in the figure). For increased visibility while viewing on paper, *the values have been inverted*, hence white is lowest variance, black highest. As expected, Figure 3.32-b shows high variance around the edges of the image, and low variance where the image is smooth. In this way, we have constructed a crude 'edge map', or algorithmically this is acting



(a) Lena

(b) “edginess”

Figure 3.32: Fit Statistics

as an *edge detector*. We can think then of the greyscale value of the the pixels as a measure of ‘edginess’.

However, the result is quite crude, for two reasons. The primary problem is that we expect the edges to be contained within the edge blocks, but our map classifies the entire range block the same way, which is clearly inadequate. There is a second issue, which is that clearly the variance of the error of fit is quite a crude way to measure the ‘edginess’ of a region of the image. It is less clear, however, what the right quantity to measure is, especially in the absence of a good model of what edges are!

The first, and more fundamental problem may be addressed by considering that the error statistics of the fit are not the only information we have gathered. The structure of the ‘fractal operator’ T, is also important. Assuming that the fixed point is a good approximation of the image, the edges will be reproduced well in this approximation $\bar{u}$. It is then instructive to as “Where did these edges come from?”. At one level of iteration, at least, the edges are images of edges in the parent block, under the appropriate affine map.

In this sense, the edge information is propagated around the image under the action of the operator T. This suggests we might subject our variance information to the same procedure, at least geometrically. There is no reason to expect in general that the affine maps will be meaningful for the error variance.

For this reason, we apply a modified version of the operator T, considering only the geometric part, but also suppressing 'noise' incurred by mapping non-edgy parts of the image. The new operator is

$$(\tilde{T}v)(x,y) = \sum_i v(x,y)v(w_{i,j}^{-1}(x,y)) \quad . \tag{3.4}$$

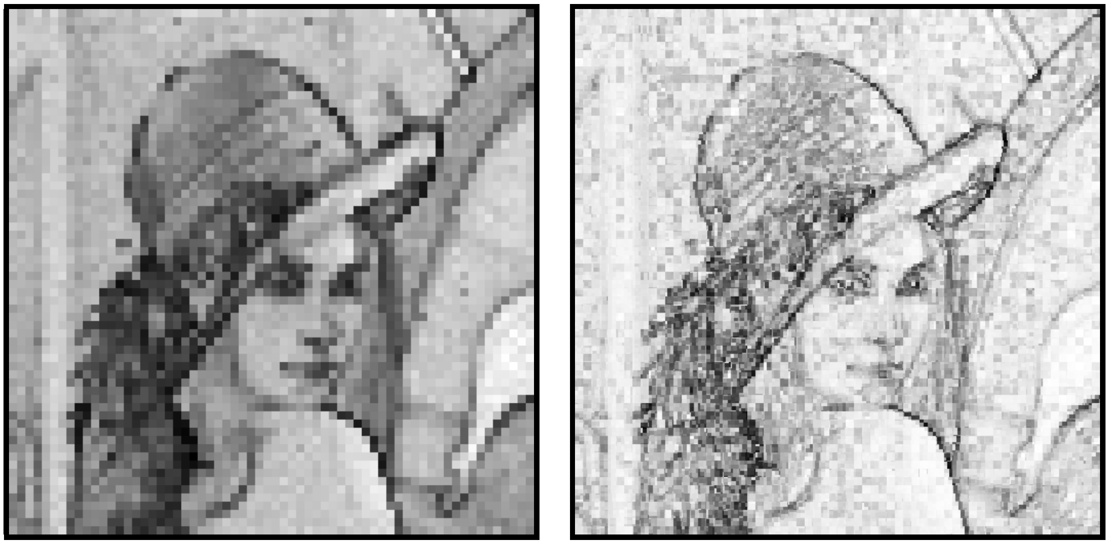
Here we are suppressing 'noise' by product with the current 'edginess' of the range block, and emphasising strong edges by taking the log of the result. However, this alone tends to emphasise background noise too much, so an additional 'refinement' step is taken: we logarithmically stretch the pixel values and then normalise.

Of course this entire approach is heuristic, and it should once again be stressed that this approach is very preliminary, and the above operator is clearly ad-hoc. The following experiments are, however, interesting and suggest some avenues for future work. For this reason it is felt that the material is worth including in this discussion of IFSM methods.

Figure 3.33 shows the result of applying $\tilde{T}$ to the error map once. Here we clearly see that the edge information has been refined, in the sense that it is now showing 'edges' at sub-range-block resolution.

Repeated application of this process will further refine the edge information, but clearly begin to lose 'weak edges'. In general, the best approach here will be application specific (i.e. does one want only strong edges? all edges?).Figure 3.34 shows the difference between one and two applications of the mapping.

At this point, it is reasonable to ask if such a method is competitive with other edge detection routines. Figure 3.35 shows a comparison



(a) ‘edginess’

(b) mapped error

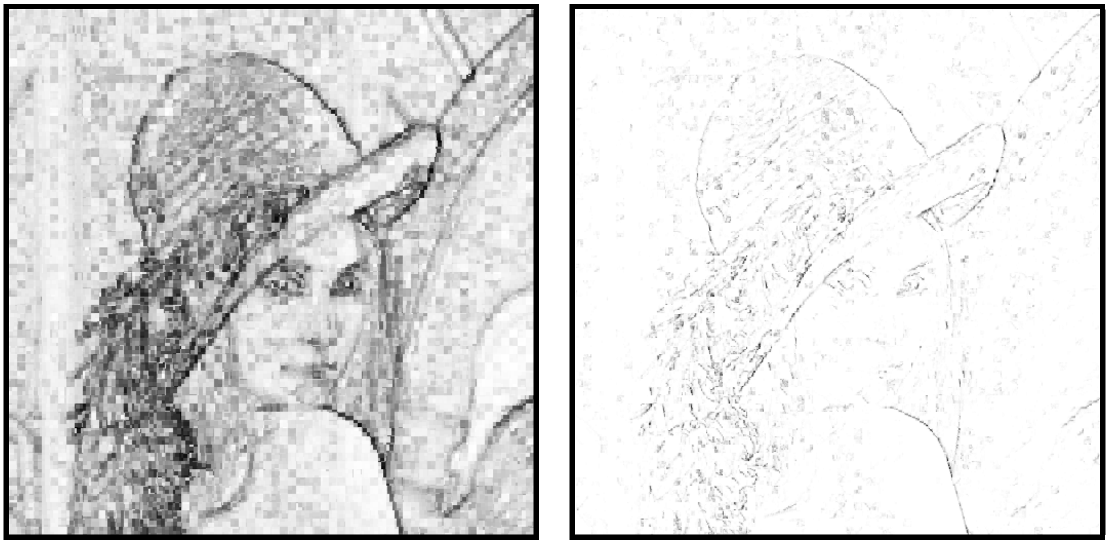
Figure 3.33: (a) ‘edge map’ and (b) ‘edge map’ after one iteration of the IFS.

with the standard Sobel filter approach.

Clearly our result is not as good. Of course, this approach is much more expensive than a simple filter, too! These are two strikes against the approach, it would seem, but that might be a bit hasty. Although there is reason to believe that the classification method itself can be improved, even as it currently stands there are two reasons that such a method may still be of interest.

The first reason is that if we are considering any sort of IFS based image processing (be that compression or anything else) then we have already paid the price of the optimisation procedure, so this ‘edge’ information is essentially free — furthermore, rather than an edge map arrived at by another algorithm (such as the Sobel result demonstrated above) this classification tells us where the fit *actually* had a lot of error.

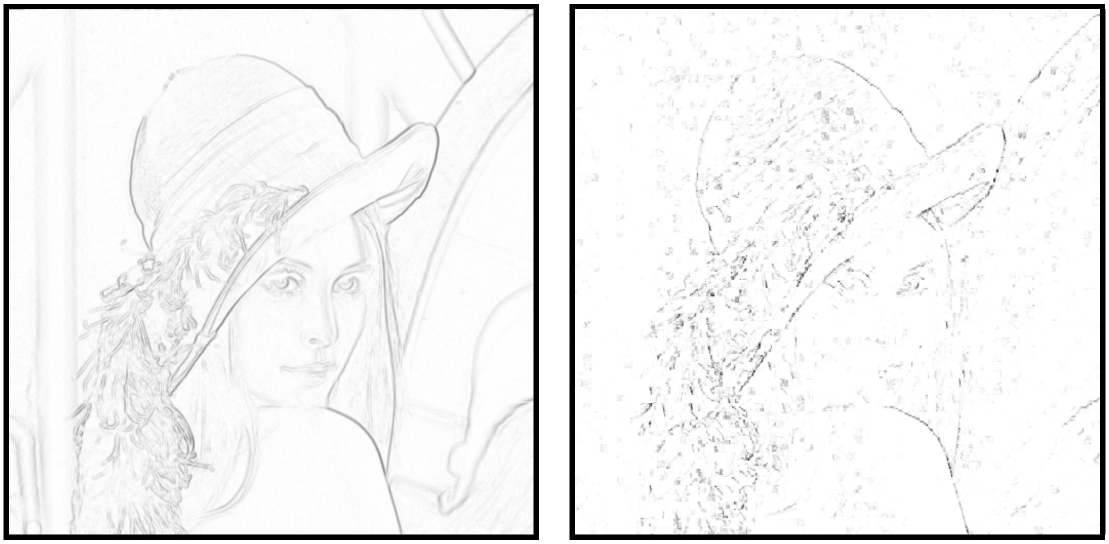
A second reason is potentially even more interesting, and discussed in the following section.



(a) once

(b) twice

Figure 3.34: 'edge map' after (a) one and (b) two mappings through the IFS.



(a) Sobel

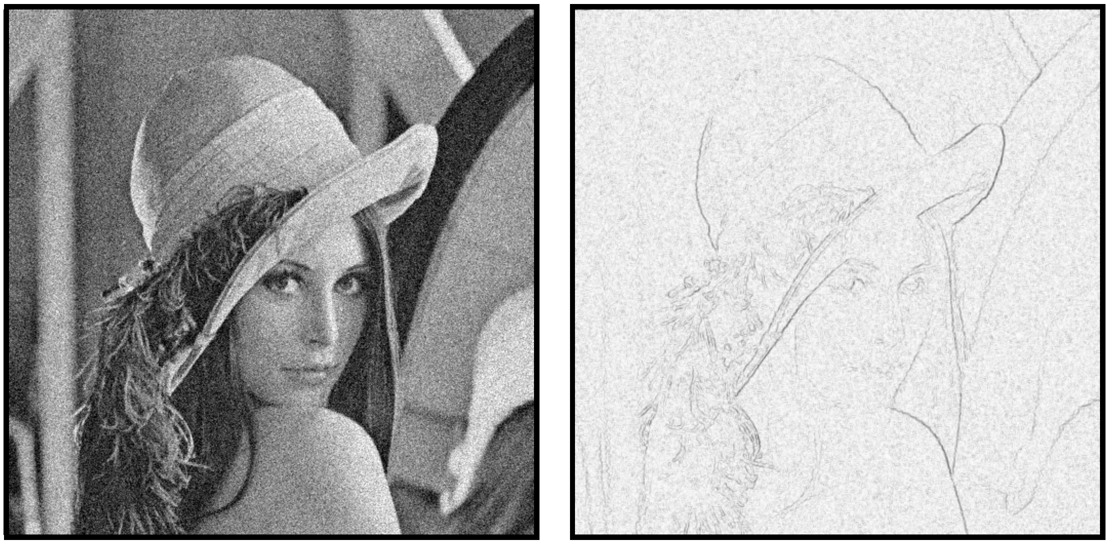
(b) IFS method

Figure 3.35: Comparison of IFSM edges with Sobel filter output.

3.7 Edges in the presence of noise

Edge detection and noise reduction are two of the classic image processing applications. When taken together, they are a source of much added difficulty. That is, naive noise reduction schemes tend to destroy edges, and conversely edge detection is made much more difficult by the presence of noise [Can86].

For a relevant example, if we apply the same Sobel edge detection to Lena with added noise, a huge number of extraneous 'edges' are introduced, at the same time as some 'clear' edges are no longer identified. Figure 3.36 demonstrates this. Note that major edges are still stronger, so one could threshold or smooth the results. However, such post-processing will also destroy finer edges that are not just artifacts of the noise.



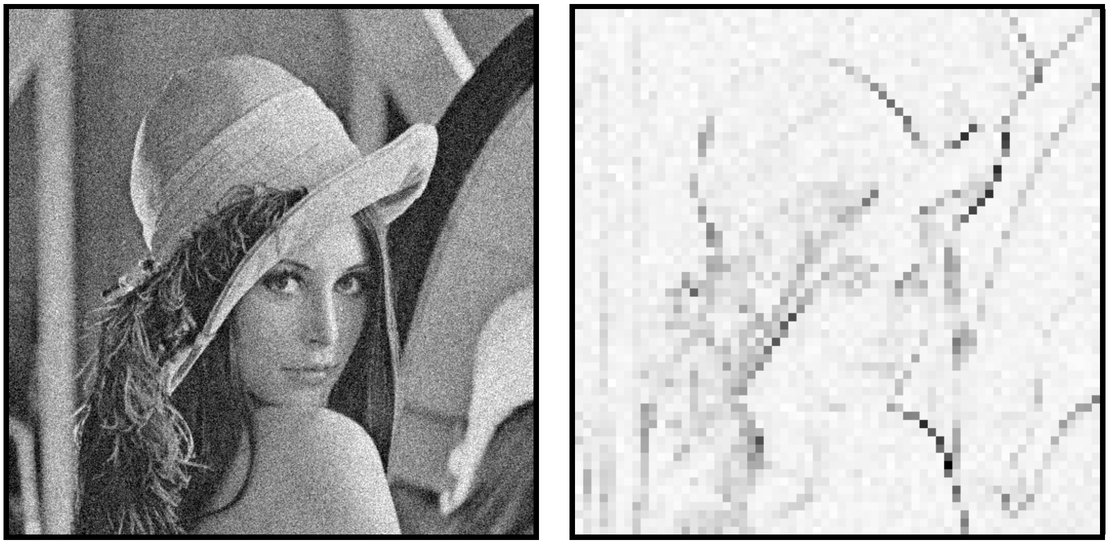
(a) Lena

(b) Sobel

Figure 3.36: Edge detection in the presence of noise is problematic, as demonstrated by applying a Sobel filter to (a) a image of Lena + noise, resulting in (b)

An intriguing aspect of the IFS approach is that rather than the usual introduction of extra 'edges' due to noise artifacts, the method can find *fewer* edges. The introduction of noise will increase the mean error of the parent-child fit, clearly. However, edgy children will still have significantly higher error variance, when tried against all parents! It is also clear that this difference will narrow as more and more noise is introduced, but that is also natural in the setting.

Figure 3.37 shows the input image, and the variance map (comparable to Figure 3.32). Figure 3.38 compares the noise-free and noisy Lena IFS variance maps (without the action of the IFS map). Figure 3.39 shows refinement of the map, while Figure 3.40 compares the final output of both methods.


(a) Lena

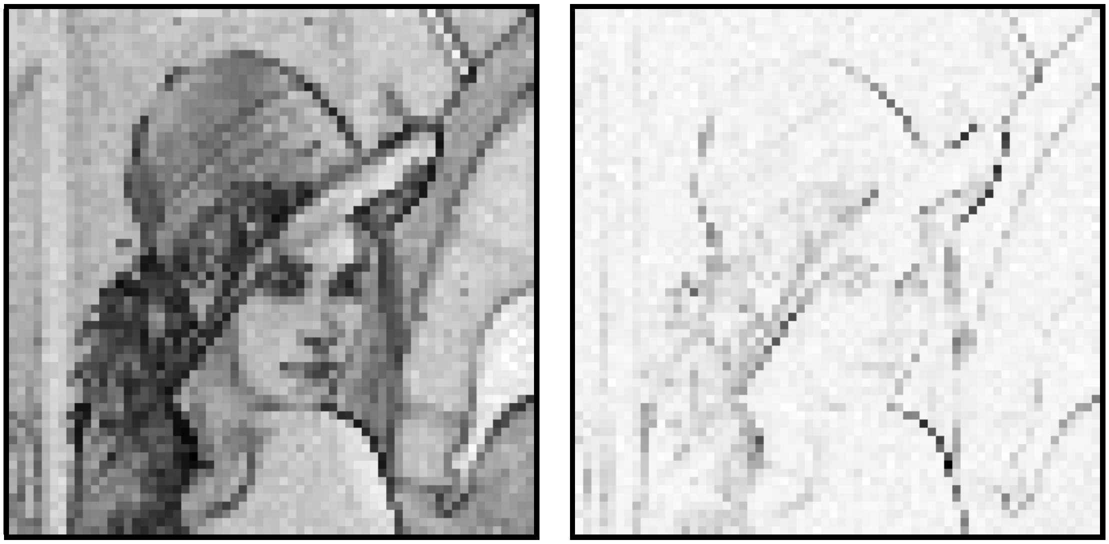
(b) IFS method

Figure 3.37: Variance map of noisy image

Where is this going?

As mentioned, the preceding is quite preliminary. For this reason, the ‘edge detection’ results in the previous section show a comparison with the results of a Sobel filter, rather than a more sophisticated result, such as the Canny [Can86] edge detector.

The reason for this is that the results shown here are envisioned as a low-level input to a less naive approach. Much as the Sobel



(a) Lena

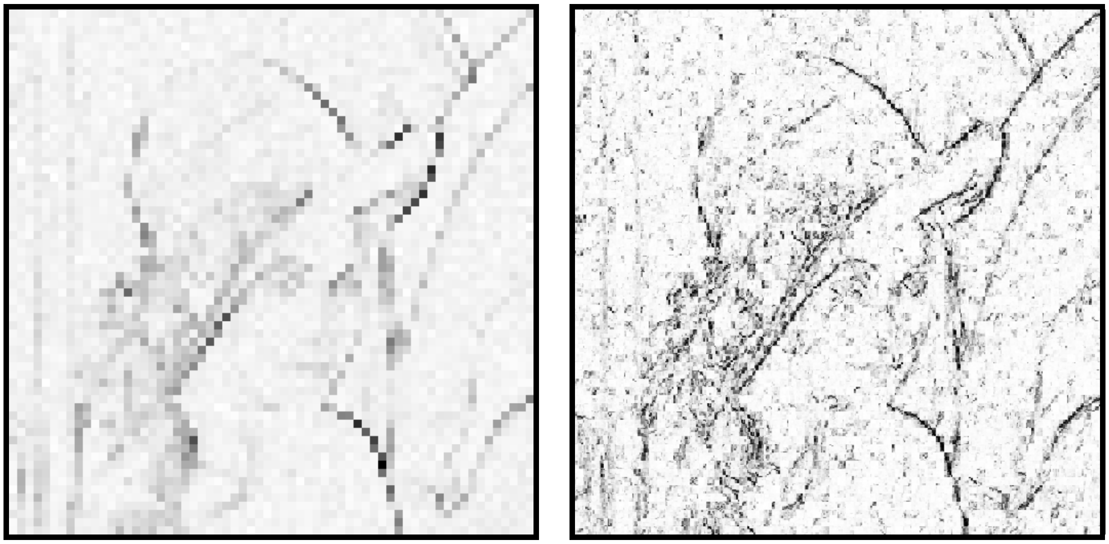
(b) noisy Lena

Figure 3.38: IFS based edge map for noise free (a) and noisy (b) Lena images

is essentially gradient information that may be fed to a higher level ‘edge detector’, these IFSM ‘edge maps’ are envisioned as an initial step.

What this boils down to is a conjecture, if you will, that the IFSM will allow us to interrogate the image in some sense about its structure. In order to fully realise a practical algorithm based on this, several further steps are needed.

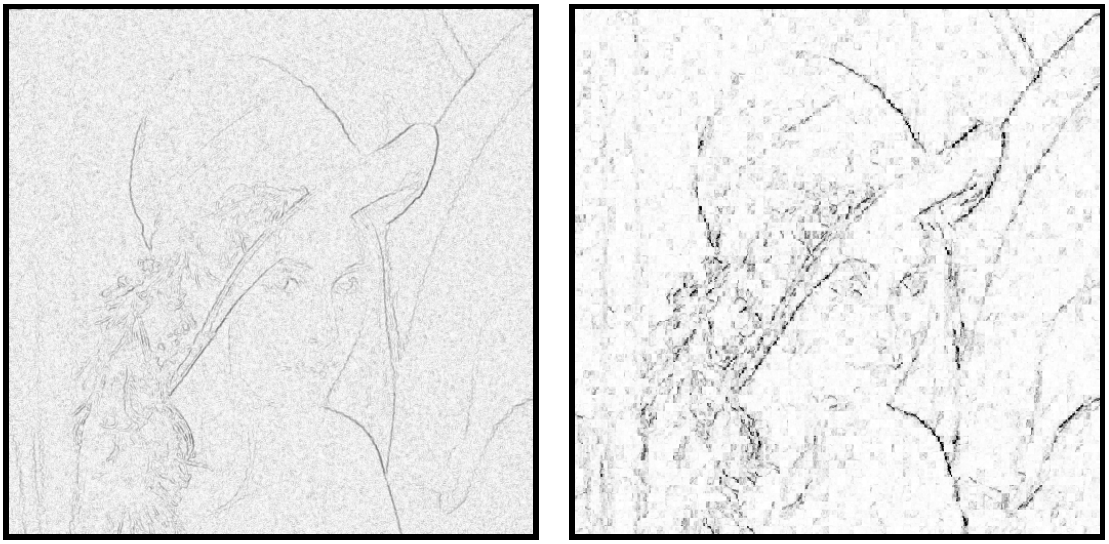
Foremost, the idea of classification of range blocks based on information in the IFSM fitting process must be done. It is not clear what the best approach will be, but there are several obvious things to try. Once this has been achieved, it opens the way both for informed post processing, and hopefully for removing reliance on heuristic or ad-hoc algorithmic decisions.



(a) IFS method

(b) refined once

Figure 3.39: Refined edge map



(a) Sobel

(b) IFS method

Figure 3.40: Comparison of Sobel edge map (a) and IFS edge map (b) for the noisy Lena case

CHAPTER 4

Random Fields and Simulated Annealing

As previously noted, modelling of natural images has turned out to be difficult in general. In practice, many approaches make simplifying assumptions that are valid for specific applications.

4.1 Why Random Fields?

However, one of the most general approaches is to consider images as realisations of a random field. Rather than assuming an underlying functional form, physical process, or perhaps localised structure, in general one can consider each pixel value to be a realisation of a random variable which is potentially dependent on the state of all other pixels. This is a very general model, which has been broadly applied. Markov Random Field (MRF)[1] approaches have been used for a variety of applications, such as texture synthesis, texture

[1] Which will be defined shortly.

analysis, image restoration, image segmentation, classification, etc. [DK89, KZB99, KBZ93, CJ83, GG84] For example, Figures 4.1 and 4.2 show two 'texture images'. There is a large literature devoted to the sampling and classification of such textures utilising MRF techniques, see for example [HPB98, Gra87, CJ83, CJ91]. Synthesis of such textures would be achieved by either purely *a priori* or posterior sampling of a field, whereas classification of textures could be done by *maximum a posteriori* (MAP) estimates of several different MRF models (corresponding to different 'textures'). Figure 4.2 shows a very simple classification such as this, based on Gaussian Markov Random Field (GMRF) (i.e. the random variables are all Gaussian).

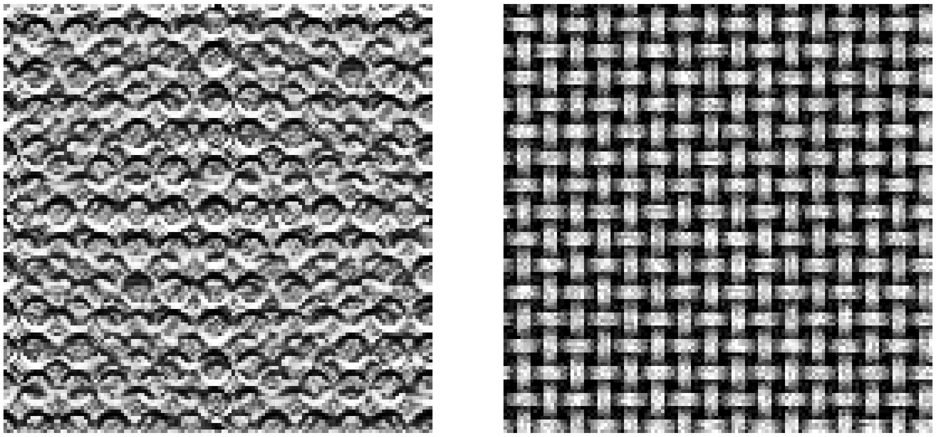


Figure 4.1: Textures such as the above are one area to which Markov fields have been widely applied.

Another application area which has enjoyed success is image restoration. Any discussion of Markov Random Field (MRF) methods in image processing is indebted to the seminal paper by Geman & Geman [GG84], which introduced the approach to an imaging context and includes discussion of image restoration, segmentation and other estimation problems, indeed setting up a quite general framework for imaging problems.

Multiscale approaches occur quite naturally in some of these settings. To date, most such approaches have involved hierarchical

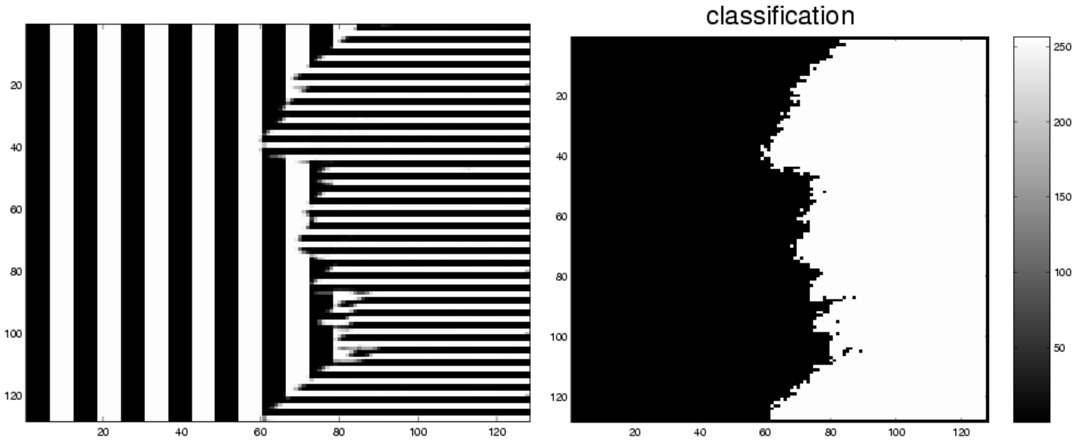


Figure 4.2: Extremely simple classification of binary textures with GMRF models. The left hand image is a two-texture image for classification, and the right hand image shows a classification based on two GMRF models (i.e.'classes') which capture vertical and horizontal textures, respectively.

structures built *above* a random field model, in a bottom-up approach (see for example [ZKB94, PB99, PL98, BS94]). By contrast, in this thesis we will be presenting a top-down, 'pure' (in some sense) annealing technique. Before getting into details though, a bit of background information is presented.

4.2 Random Fields

The history of the development of these ideas draws heavily on three separate areas of study: image processing, spatial statistics, and statistical mechanics. In order to proceed with a statistical description of images, we need to have a framework for modelling images. As previously mentioned, the configuration space of images is typically very large, and this can make theoretically straightforward operations intractable in practice. For this reason the development of Markov random fields is of particular interest to imaging applications.

The study of *Random Fields* has its roots in statistical physics, and a long history. Unlike the previous chapters on IFS methods, it is not practical to reproduce a complete development here. Instead, a rather narrow path to the results we will need (in particular, to simulated annealing) will be traced, with mention of the theorems needed and pointers to references for a complete discussion. We begin by defining discrete fields of discrete random variables.

Definition 4.1 (Random Field) *Let S be a finite set of sites, and denote a site $s \in S$. Let Λ be a finite set of states. A random field on S is a collection $X = \{X(s)_{s \in S}\}$ of random variables $X(s)$ on the state space. Denote the* configuration space *as Λ^S, and call $x \in \Lambda^S$ a* configuration. □

In the application to images, the random field would typically have S the set of pixel locations, and Λ the set of possible pixel values (e.g. $\Lambda = 0 \ldots 255$ for 8-bit pixels), but this is not the only possibility.

The idea of a Markovianity property can be used to reduce the complexity of the computations needed for our models. Very loosely stated, this property encapsulates the idea that the conditional probabilities for a random variable conditioned on some subset of a space are independent of the state of the rest of this space. In terms of images, what we would like to do is define models such that densities are dependent only on local state (for some definition of local). All this will be made explicit.

The intuition for why this is important to images is quite simple. Consider an element X_n of an image $\mathcal{I}$, modelled as a random variable. If we wish to consider the probability of X_n taking on a particular value, the simplest approach would be to condition on the rest of the image. This has two major drawbacks. The first is that it is computationally very expensive, and the second is that there is no reason to expect (from the optics of image generation, or other sources of images [Wan95, Bie85]) that a pixel is dependent on all other sites in the image. A formal development of the desired property is found in Markov random fields.

4.2.1 Markov Random Fields

Markov Property

Attempting to model something with a stochastic process made of iid random variables is often not very fruitful. In order to introduce more interesting variability, it may be useful to allow some dependencies. In general then, one can imagine conditioning on the history of the process, but it is problematic to keep track of this. One possibility is to introduce dependence on the previous state only. An example of this behaviour is a discrete-time homogeneous Markov chain. That is, a stochastic process $\{X_n\}_{n\geq 0}$ on a finite state space with states $i_0, i_1, \ldots, i_{n+1}$ with the property that

$$P(X_{n+1} = i_{n+1}|X_n = i_n, X_{n-1} = i_{n-1}, \ldots X_0 = i_o) = P(X_{n+1} = i_{n+1}|X_n = i_n) \ . \tag{4.1}$$

In other words, given the current state, we can predict the next state of the chain without knowledge of the past. This sort of ‘decoupling’ is called the Markov property. For that reason, in situations where (4.1) holds, the process is called a Markov Chain. A general discussion of Markov Chains and their properties is beyond the scope of this thesis, but may be found in [Bré98].

Markov Property in two dimensions?

We wish to define an extension of the Markov property in one dimension, to objects of two (or more) dimensions. In higher dimensions the natural ordering of a 1-d process is lost, in the sense of time sequences it is no longer possible to define ‘past’ and ‘future’ in a natural way. However, the essence of the Markovian property can be maintained by viewing it as a requirement for *local* dependence. What remains then is to define what we mean by local.

Definition 4.2 (Neighborhoods) *A neighbourhood system on* S *is a set* $\mathcal{N} = \{\mathcal{N}_s\}_{s\in S}$ *of subsets of* S*, such that for all* $s \in S$*:*

1. $s \ni \mathcal{N}_s$

2. $t \in \mathcal{N}_s \Rightarrow s \in \mathcal{N}_t$

The subset $\mathcal{N}_s$ *is called the* neighbourhood *of* s. □

Having defined a neighbourhood, we now define *cliques* as a way of describing neighbourhood relationships.

Definition 4.3 (Clique) *A singleton* $s \in S$ *is a clique. Any* $\mathcal{C} \subset S$ *is a clique if and only if any distinct* $s, t \in \mathcal{C}$ *are neighbours. The clique is called maximal if* $\forall s \in S \setminus \mathcal{C}$, $\mathcal{C} \cup \{s\}$ *is not a clique.* □

This definition follows from the graph theoretic approach, but it is also a convenient way to view neighbourhood relationships. Figure 4.3 shows how these definitions relate to image data. The sites S in this case are taken to be a grid with random variables at each pixel location. We define the 'order' of a neighbourhood based on the (Euclidean) distance of the neighbours from the current pixel. Thus a 1st order neighbourhood contains the horizontal and vertical neighbours of a pixel, the 2nd order adds the diagonal neighbours, and so on (see 4.3(a)). Figures 4.3(b) and 4.3(c) show the cliques for a 1st and 2nd order neighbourhood of this type, respectively. The "dots" represent pixels, and the lines show clique relationships.

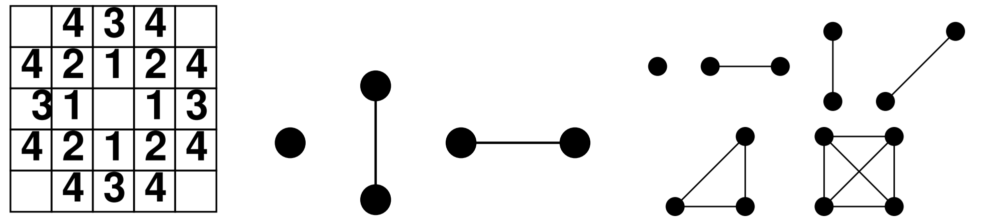
Definition 4.4 (Markov Random Fields) *A random field* X *is called a MRF with respect to the neighbourhood system* $\mathcal{N}$ *if* $\forall s \in S$, *given* $X(\mathcal{N}_s)$, *the random variables* $X(s)$ *and* $X(S \setminus \{\mathcal{N}_s \cup s\})$ *are independent.*□

The key point to a Markovianity property is that it allows localisation of the probability density. Hence it is useful to define the following:

Definition 4.5 (Local Specification) *Consider the density at a particular site in the neighbourhood system:*

$$\pi^s = P(X(s) = x(s)|X(\mathcal{N}_s) = x(\mathcal{N}_s)) \qquad (4.2)$$

The local specification *of the field is the collection* $\{\pi^s\}_{s \in S}$. □



	4	3	4	
4	2	1	2	4
3	1		1	3
4	2	1	2	4
	4	3	4	

(a) neighbourhoods up to 4th order

(b) first order cliques

(c) 2nd order cliques (not all rotations)

Figure 4.3: First through fourth order neighbourhoods of central pixel ○ are shown in (a). Note a neighbourhood of order n includes all pixels labelled $<= n$. The first order cliques are shown in (b), and second order (but not all rotations) in (c).

4.2.2 Gibbs Fields

As will be discussed more in the following, there is another useful way to look at the same fields. While the idea of a Markov property on the graph is a helpful way to think about limiting the conditioning one needs consider, energy methods allow a different approach. In most of the work to follow, we will be focusing on this framework. Originating from statistical physics, a *Gibbs* (sometimes called *Boltzman*) distribution for inverse temperature $\beta = 1/T$ is of the form:

$$\pi_\beta(x) = \frac{e^{-\beta\mathcal{E}(x)}}{Z_\beta}, \tag{4.3}$$

where the partition function, Z_β, is given by

$$Z_\beta = \sum_{z \in \Lambda^S} e^{-\beta\mathcal{E}(z)} \ . \tag{4.4}$$

Here the energy function $\mathcal{E}(x)$ takes a particular form, known as a *Gibbs potential*:

Definition 4.6 (Gibbs Potential) *We call a collection* $\{V_C\}_{C \in S}$ *of functions* $\Lambda^S \to \mathbb{R}$ *a Gibbs potential on* Λ^S *if:*

1. *$V_C \equiv 0$ if C is not a clique.*

2. *$x(C) = x'(C)$ implies $V_C(x) = V_C(x') \qquad \forall x, x' \in \Lambda^S, C \subset S$.*

Under the above conditions, the potential derives an energy function for a Gibbs distribution:

$$\mathcal{E}(x) = \sum_C V_C(x) \tag{4.5}$$

□

As an aside for the moment, note that the normalisation constant, or *partition function* Z_β, is a significant problem in the imaging contexts we are interested in here. Although particularly for some simple physical systems, determining the partition function analytically allows inferring properties of the system, this is not generally possible. With the large configuration spaces we will be considering, brute-force calculation of Z_β is completely intractable. This fundamental difficulty suggests approaches that avoid knowledge of Z_β, for example the Markov Chain Monte Carlo (MCMC) methods that will be discussed later.

The following two theorems show the equivalence of Gibbs fields and Markov fields. First, all Gibbs fields are Markov fields:

Theorem 4.1 *Let X be a random field of the form (4.3) on configuration space Λ^S, with neighbourhood system $\mathcal{N}$, and energy function (4.5). Then X is a Markov field relative to $\mathcal{N}$.* □

Proof See (Bré98, Win91b) ■

The converse was first suggested in [HC71], but not published until much later:

Theorem 4.2 (Hammersly-Clifford) *Let π be the distribution associated with Markov random field with respect to a neighbourhood system $\mathcal{N}$, and sites S (i.e., a graph $(S, \mathcal{N})$). Suppose π satisfies a positivity condition on the marginal distribution π_j at site $j \in S$: $\forall j \in S$*

and $\forall\ y_1, \dots, y_{j-1}, x_j, y_{j+1}, \dots, y_k \in \Lambda$

$$\pi_j(x_j) = 0 \quad \Rightarrow \quad \pi(y_1, \dots, y_{j-1}, x_j, y_{j+1}, \dots, y_k) = 0. \tag{4.6}$$

Then

$$\pi(x) = \frac{e^{-\beta * \mathcal{E}(x)}}{Z_\beta}, \tag{4.7}$$

is a Gibbs field deriving from some Gibbs potential $\{V_C\}$. □

PROOF See (Bré98, Win91b). ■

The forward direction allows efficient simulation of Gibbs/Markov fields as discussed by Metropolis *et al.*[MRR+53] and generalised by Hastings[Has70], and others [GG84, Bar65]. The converse is theoretically important, as it establishes the equivalence (excepting the positivity condition) of these two structures.

Before proceeding much further, it will be informative to briefly consider an important random field model. The Ising model is not particularly important in the context of imaging, but generally and historically it is important. This model was the driving idea behind the development of much of the theory, and also has the advantage of being simple.

4.2.3 The Ising Model

The canonical example of a MRF is the Ising model. In 1925 Ising proposed a model of ferro-magnetism based on a crystal lattice of spin up or spin down dipoles. This model is discussed at length by Kindermann [KS80]. Ising's model was the motivation behind MRF theory; the idea came from attempts to generalise Ising's work. Due to the simplicity of the model, it is well suited to investigation of the concepts of MRF, although Ising did not originally set the model up in this way.

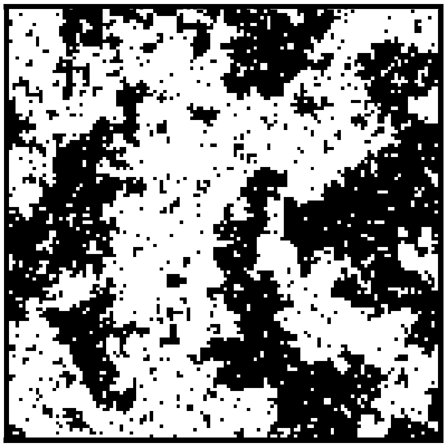


Figure 4.4: Sample from Ising Model at ‘critical temperature’

Example 4.1 (Ising Model) Here is a common energy function for the ferromagnetic model including external field B:

$$\mathcal{E}(x) = -\frac{1}{kT}\left[J \sum_{\langle s,t \rangle} x_s x_t - mB \sum_s x_s \right] . \tag{4.8}$$

Here the notation $< s, t >$ is used to describe a sum over all clique pairs. The simplest case of this model (no external field) has the energy function:

$$\mathcal{E}(x) = -\beta \sum_{<s,t>} x_s x_t \ . \tag{4.9}$$

□

Figure 4.4 shows a sample from the Ising model. The Metropolis sampler (which will be discussed shortly) was used to draw a sample from (4.3), at a particularly interesting temperature known as the ‘critical temperature’, or T_{crit} (more on this to follow, also).

4.2.4 Sampling

Direct sampling from the sorts of two-dimensional fields applicable to images is computationally infeasible, due to the size of the configuration space (recall that the partition function must be calculated over the entirety of this space). This section will briefly discuss methods of indirectly computing samples from *discrete valued MRF's*. The methods of interest are called Markov Chain Monte Carlo (MCMC) algorithms. Hastings [Has70] proposed use of the Metropolis algorithm in a more general setting, and several similar algorithms have been developed, for example Barker [Bar65], and Gibbs [GG84]. The most commonly used in imaging are the Metropolis-Hastings sampler, and the Gibbs sampler, hence they will be be discussed in a bit more detail.

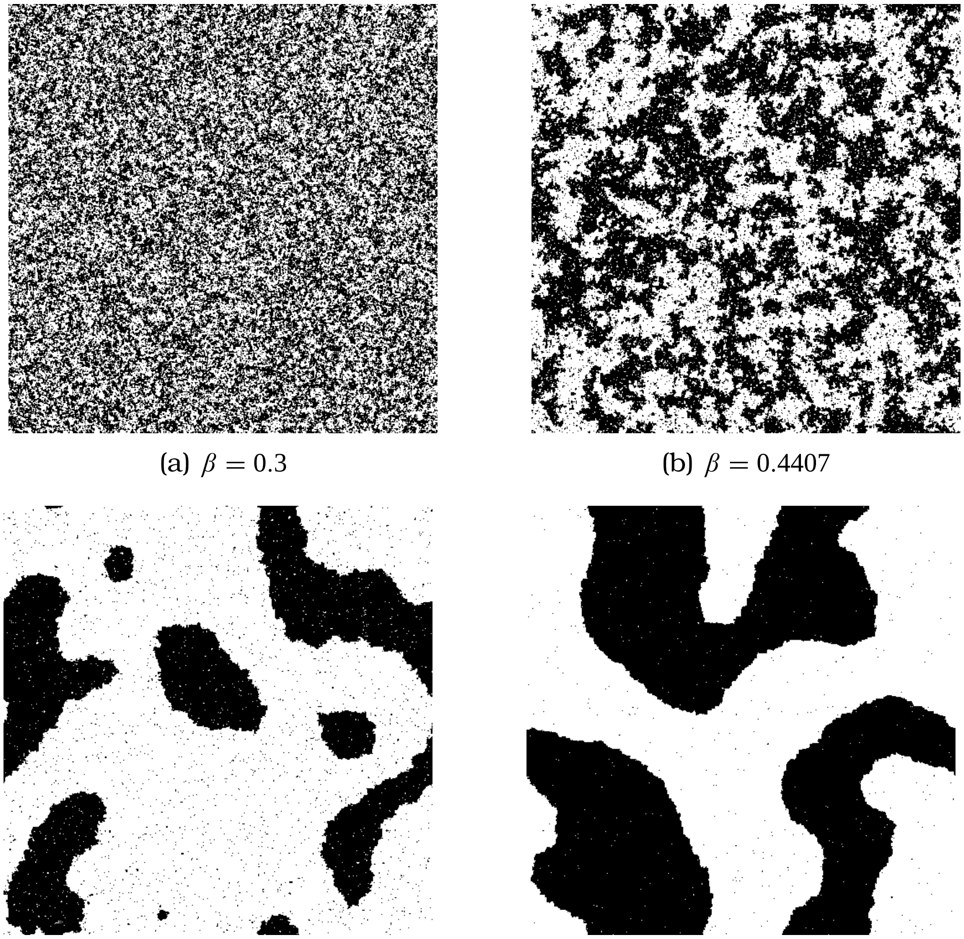
In 1953 the Metropolis sampler revolutionised the study of Gibbs fields, by allowing computer simulation [MRR+53, Has70]. This work was done in the context of statistical physics, and a large literature has followed. Decades later, in 1984, a seminal paper by Geman & Geman was published [GG84]. In this paper, the connection between these statistical physics methods and image processing problems was made. This paper introduced the *Gibbs sampler*, was first to use *simulated annealing* (as will be discussed later) for image processing, and also proposed a compound MRF.

Example 4.2 (Sampling from the Ising Model)

Figure 4.5 shows some example images taken from the Ising model (4.9) (with no external field) to demonstrate that such a simple model can result in complicated structure.

While the Ising model is not particularly useful in the context of image processing, it does provide a simple example of a Markov field which has been extensively studied. It is also one of the rare examples of such a field where the partition function is known explicitly [Win91b].

The transition from the unstructured state in 4.5-a to the complicated structure in 4.5-b is called a phase transition, and is a complicated phenomena in statistical physics (a general introductory description may be found in [KS80]. We will not have to deal *directly* with this complexity in the following work). However, it is the onset



(a) $\beta = 0.3$

(b) $\beta = 0.4407$

(c) $\beta = 0.6$

(d) $\beta = 0.8$

Figure 4.5: Samples from the simple (no external field) Ising model. All examples were iterated for 2000 sweeps. a) shows a fairly high temperature, b) shows the critical temperature for this model with structure at all scales, c) gives a lower temperature d) shows a very low temperature

of such structure that creates the phenomena known as *critical slowing down*. Once past the critical temperature T_{crit}, the convergence of samplers in this system slows down markedly. Since in chapter 5 we will in essence be dealing with convergence in low temperature regimes, this *critical slowing down* is a fundamental problem [KJ97, BGHM95], one that motivates much of the development.

MCMC algorithms

It turns out that there is a way to avoid calculating the normalisation constant Z_β while drawing samples from the Gibbs density (4.3). As might be expected, ratios of $\pi_\beta(x)$ for different configurations x are involved, so that the normalisation constant cancels. The particular techniques used are the construction of appropriate Markov chains, with long-run distributions that match the required $\pi_\beta(x)$. Transition probabilities in the chains are related to the above mentioned ratios, naturally. The second 'MC' in the MCMC sampling algorithms stands for Monte Carlo, reflecting the stochastic nature of the exploration of configuration space that these methods pursue.

In essence, as a high-level view these methods can nominally be broken down into two steps. At each iteration, the system (in our cases of interest, the image) is in a particular state. The sampler will now propose a new state for the system, according to some sort of *proposal rule*. The new state will then be accepted (or not) according to an *acceptance rule*.

To make things more concrete, the details of two samplers follow. For more detailed discussion (and discussion of other samplers), see [Has70, GG84, Bré98, Win91b].

The Gibbs Sampler

Algorithmically, the Gibbs sampler behaves as follows: Take the system in state x_i. A new state x_j is proposed differing from the current state x_i at one site, s. This site is chosen with probability q_s, independently of the past choices. The new state x_j will be:

$$x_j(t) = \begin{cases} x_i(t) & t \neq s \\ \text{sample from } \pi(x_j(t)|x_i(S \setminus s)) & t = s \end{cases} . \tag{4.10}$$

In other words, the new state for x_j at site s is drawn according to the local specification (4.2).

Metropolis Sampler

Suppose the current configuration is x_i. Propose a new configuration x_j uniformly from all configurations that differ from x_i at only one site. Accept this new configuration with probability $\min\{1, \frac{\pi(x_i)q_{i,j}}{\pi(x_j)q_{j,i}}\}$. A common case is a purely random candidate-generation, i.e. $q_{i,j}$ is a constant. Then the acceptance probability is $\min\{1, \frac{\pi(x_i)}{\pi(x_j)}\}$

In practice, sites may be chosen randomly, or pseudo lexicographically if some care is taken to avoid picking sites in the same clique in succession. The Metropolis sampler has been used in the computer experiments reported in the next chapter.

Convergence of MCMC samplers

As noted, essentially the game here is to construct transition properties for an appropriate Markov chain, so that the chain will converge in distribution to (4.3). Each sampler does this in slightly different ways, and so the details of these constructions differ. Details for the Gibbs sampler are given in [GG84], and Brémaud [Bré98] has a nice discussion of a unified approach covering most algorithms, which follows Hastings [Has70].

It is not the MCMC samplers that we will be interested in directly, but rather their application in a technique called 'simulated annealing' that will be important. This quick sketch of MCMC methods is by no means complete, but has provided the necessary background to proceed to simulated annealing.

4.2.5 Simulated Annealing

Simulated annealing [Čer85, KGV83] is an optimisation method which was named for its resemblance to the chemical/metallurgical phenomenon known as annealing.

In essence, the method takes the homogeneous Markov chains defined by the samplers in the previous section, and replaces them with non-homogeneous Markov chains by making the parameter β a function of time (or the index of the sequence of site visitations, to look at it a different way). In particular, while the system may return to the same configuration x, and hence the energy $\mathcal{E}(x)$, the density $\pi_\beta(x)$ will have changed if β has, and thereby the transition probabilities can change.

Algorithmically, the method involves repeatedly sampling from a field while reducing the temperature, i.e. $T = 1/\beta$, of the model according to a *cooling schedule*. Algorithm 4.1 shows this process.

Algorithm 4.1 Simulated Annealing

```
$k \Leftarrow 0$
$X \Leftarrow$ initial state
while $\mathcal{E}(X)$ not converged do
    $\beta \Leftarrow 1/T_k$
    $X \Leftarrow$ apply sampler to $X$
    $k \Leftarrow k + 1$
end while
```

Definition 4.7 (Cooling Schedule) *A* cooling schedule *is a sequence of temperatures* $\{T_n\}$ *such that:* $T_n \to 0$ *as* $n \to \infty$. □

It should also be noted at this point that Algorithm 4.1 is intentionally vague about what 'converged' means, which will be discussed further.

Defining a cooling schedule as an eventually decreasing sequence, allows for many approaches, but does not give much insight into what is typically done. The following two definitions give particularly interesting cases for our purposes.

Definition 4.8 (Logarithmic Cooling)

$$T_k = \frac{A}{\log(k+1)} \qquad A > 0 \textit{ a constant.} \tag{4.11}$$

□

Here A is unknown, and in general dependent on the energy function.

Unfortunately *logarithmic cooling* is completely impractical for large configuration spaces, and/or energy functions that are computationally expensive to evaluate. However, most general analytic results rely on this schedule.

Definition 4.9 (Geometric Cooling)

$$T_k = \alpha^k T_0 \qquad 0 < \alpha < 1 \ . \tag{4.12}$$

□

Again T_0 (the initial temperature) is unknown, and will be chosen for particular energy functions. This method is also called *exponential cooling*, and of course has the property that $T_{k+1} = \alpha T_k$.

In practice many other cooling schedules are used. In this work, we will primarily be concerned with geometric cooling, which has the advantages of being practical in the cases of interest to follow, and easily parameterised. Unfortunately, it has also proven difficult to analyse [Cat92].

Although there are technicalities involving particular sampling algorithms, for the cases considered here we have the result that simulated annealing is a global minimisation algorithm, given particular cooling schedules.

Convergence of simulated annealing

The basic idea behind showing the convergence of simulated annealing is to take the homogeneous Markov Chain of a particular sampling algorithm and change it into a non-homogeneous Markov Chain by introducing a cooling schedule ${T_k}_{k=0}^{\infty}$; the transition probabilities are now a function of this temperature. The particular cooling schedule needed to ensure convergence of this non-homogeneous Markov chain is dependent on the sample. Under appropriate cooling schedule, then, this algorithm will converge *to a uniform distribution over the global minima of the energy function.*

Geman & Geman [GG84] show this for the Gibbs sampler, a cooling schedule satisfying

$$T_k \geq \frac{N\Delta}{\log k} \quad , \tag{4.13}$$

is sufficient. Here Δ is the sup of all possible energy changes by the sampler, and N is the size of the lattice. This is an impractically slow schedule.

Hajek[Haj88] has shown for the Metropolis sampler a necessary and sufficient condition for convergence of annealing is the existence of a constant γ such that the logarithmic cooling schedule

$$T_k = \frac{a}{\log(k+1)} \quad \text{converges if and only if} \quad a \geq \gamma \quad . \tag{4.14}$$

This strong result may be reassuring from the point of view of the 'no free lunch' theorem, but it also indicates why using this approach in practice is difficult.

So what is going on here? Intuitively, it is helpful to reflect on (4.3), and imagine a complicated 1-d energy function. Clearly as $T \Rightarrow \infty$ the shape of the energy function becomes irrelevant. So at high temperatures, the sampler will uniformly explore the configuration space. As the system cools, the structure of the energy function is revealed, with local minima, and (generally) basins around them. If the system finds itself in a local minima, in order to escape it will have to 'jump' out by making a (or several) energetically non-favourable move(s). This can only happen with some probability given by the transition probabilities in the Markov chain. For higher temperatures, this is easy (high probability). However, as temperature is lowered, it is more and more likely that the system will fail to 'jump out' of any particular basin on a given attempt. Thus it can become stuck if the probability of escape becomes low enough.

So this describes how it may reach a particular local minima, but not how global optimisation is achieved. It turns out that if you cool slowly enough, only the deepest energy wells can trap the system. For this to happen, the system must be exploring the configuration space in such a way that as the temperature cools below the level that escaping the deepest wells is likely, the system is (with probability 1) in one of those wells.

This is why in general very slow cooling will be needed to guarantee convergence to global minima. As noted though, this may not be practical. Indeed, Brémaud [Bré98] comments: "The results... are of theoretical and qualitative interest only. Practical algorithms use faster than logarithmic schedules..."

For this reason, many approaches to faster cooling have been devised [Bes86, HRSV86, Haj88]. As previously noted, in the applications discussing in chapter 5, we will primarily consider geometric cooling schedules.

4.3 MRF's and images

Now that the foundations of a MRF approach have been sketched, it is time to ask: what about images? The seminal paper by Geman & Geman [GG84] introduced these techniques to the area of imaging, and the following years brought various refinements (for an overview discussion see [Bré98, Win91b]). It has been noted previously that the Ising model, while simple to apply to (binary) images, is not particularly useful for many imaging contexts. The simplest (in some sense) MRF that has seen widespread use in imaging is the Gaussian Markov random fields (GMRFs). The next few sections will offer some commentary on the GMRF, as it will be referred to later on. There is much literature about MRFs in areas such as texture analysis, segmentation, and denoising. However, we will focus rather narrowly toward issues of interest to a motivating application: reconstruction of porous media images (to be discussed in the next chapter).

4.3.1 The Gaussian Markov Random Field (GMRF)

A more specialised approach is possible if **x** is Gaussian. These models may be related to autoregressive models [CHK83], which are popular in image processing. It turns out that these GMRF's can be re-cast in terms of expectations rather than probability density functions [CK82], [ZFW00]. This results is estimates of the form

$$\hat{x}_{i,j} = E\left[x_{i,j}|x_{k,l},\ (k,l) \in N\right] \ . \tag{4.15}$$

For Gaussian x the expectations are linear so we may re-write this as

$$\hat{x}_{i,j} = \sum_{(k,l)\in N} a_{i,j,k,l}\ x_{k,l} \ . \tag{4.16}$$

The coefficients simplify if the field is stationary, then $a_{i,j,k,l}$ becomes $a_{i-k,j-l}$. This notation follows [ZFW00]. Approaches to the GMRF are discussed by R. Chellappa *et al.* [CJ91], [CHK83], [CK82], and also by H. Derin and P. Kelly [DK89].

It is worth noting that equation 4.16 could be misleading, as it is similar to the application of a filter (with the middle pixel "left out"). This is due to the special constraints (Gaussian, toroidal, stationary), and does not imply that neighbourhood size constrains modelled feature size (however, they are related [Win91a]).

4.3.2 Toroidally Stationary GMRFs

This particular restriction of a GMRF is quite useful. If an NxM field is toroidally stationary (i.e., periodic) then its correlation structure may be represented as

$$\Lambda_{i,j} = E\left[x_{0,0}x_{i,j}\right] 1 \leq i \leq N,\ 1 \leq j \leq M \ . \tag{4.17}$$

This is noteworthy because this matrix will be diagonalised by the FFT [ZFW00]. R. Chellappa *et al.*[CK82] describe the model in more detail, along with discussion of some important implications for the toroidally stationary case; they describe the necessity for it to be positive-definite in order to retain the Markovian property, and how this leads to a FFT based algorithm [CHK83]. This allows easy computation of sample paths and estimates [ZFW00]. Given an array, q, of independent, unit variance Gaussian samples, then a sample path s is found:

$$s = \mathrm{IFFT}(\sqrt{\mathrm{FFT}(\Lambda)} \cdot \mathrm{FFT}(q)) \tag{4.18}$$

where $sqrt$ and $\cdot$ are both taken element-by-element, and FFT, IFFT represent the two-dimensional fast Fourier transform and inverse.

Similarly, given a set of observations $y = x + v$, where x is the field variable and v represents a noise process with $cov(v) = \sigma^2 I$ then the least squares estimate for x is:

$$\hat{x} = \mathrm{IFFT}(\mathrm{FFT}(\Lambda) \cdot \mathrm{FFT}(y)/\mathrm{FFT}(\Lambda + \sigma^2)) \ . \tag{4.19}$$

One other important relationship lies between the correlation structure Λ and the matrix of "neighbourhood weights", the $a_{i-k,j-l}$ of equation 4.16. We define this matrix G as

$$G = \mathrm{IFFT}(1/\mathrm{FFT}(\Lambda)) \ . \tag{4.20}$$

This method will be used in §5.4.2 to construct synthetic data sets with particular characteristics.

4.3.3 Other Specialised GMRFs

From a computational point of view, the "causal GMRF" is interesting because it allows simple, efficient sampling and estimation algorithms [ZFW00]. A causal MRF enforces a "time" ordering on the image, resulting in a field similar to a one-dimensional MRF. The causal MRF is of limited utility, however, since the estimates are poor for many classes of texture [DK89, ZFW00, Mum95]. For this reason, in many applications an "acausal" GMRF will be used (these comments are not restricted to the Gaussian case, acausal MRFs can model a superset of causal MRFs [DK89]). Here Derin and Kelly enumerate a large number of Markov-type image models, and discuss the connections between them [DK89]. They discuss the relation of GMRF to other Markov-type models, along with the idea of wide sense Markov (WSM) models (as opposed to strict sense), of which this GMRF is one.

4.3.4 Other MRF models

General discussion of sampling from other MRF models is beyond the scope of this report. The next chapter will dedicate some discussion to a very specific application domain. However, the literature on MRFs is quite broad, and contains many somewhat specialised approaches. As noted previously, examples of the application of MRF to texture models may be found in Geman and Geman [GG84, CJ91, Gem88], and some detailed discussion of sampling methods is found in [Win91b, Bré98]. It is also interesting to note the connections between MRF and standard image analysis techniques [Win91b, DK89, Gem88]. P. Winder, in his Ph.D thesis, discusses Markov-model texture analysis in depth [Win91a].

Having provided the fundamentals of Gibbs/Markov random field approaches in imaging, we will proceed in the following chapter to a particular motivating problem and the research arising from it.

CHAPTER 5

Reconstructing Porous Media

In this chapter, we will focus on the particular application of sampling binary porous media images. The restriction to binary images has the advantage of reducing the size of the configuration space. However, the complexity of physical data (i.e. the complex structures of the porous media) provides more than enough difficulty in these sorts of approaches. Before going further, Figure 5.1 shows a couple of examples of the sort of image we will be considering for the entirety of this chapter.

5.1 Description of the application

Why the interest in simulated annealing?

Porous media researchers would like to be able to synthesise large, preferably three-dimensional, binary images with particular statistical properties. This is a very difficult problem, not least because of the wide range (as much as 8 or 9 orders of magnitude!) of length scales that are significant for their applications.

Of course this many orders of magnitude is not going to be rep-

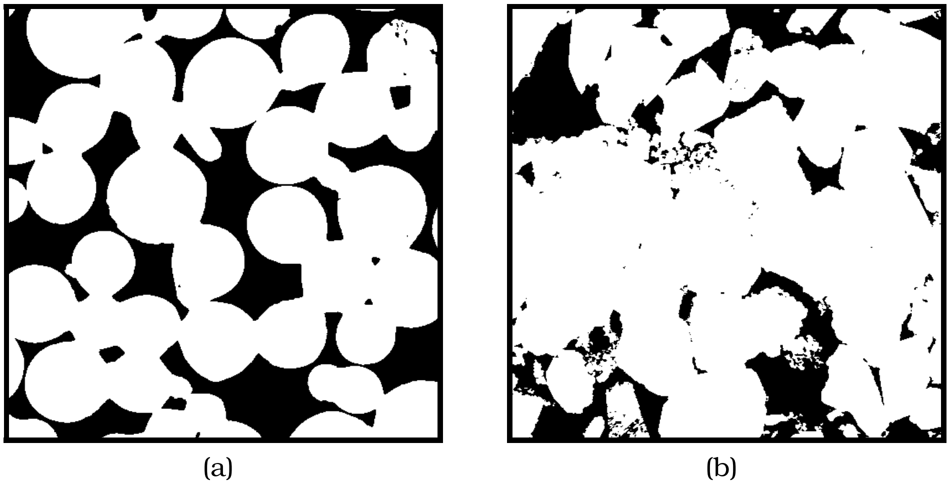


Figure 5.1: Examples of binary porous media images a) sintered glass beads b) Berea sandstone

resentable in a computers memory any time, but certainly today researchers would like to be able to synthesise data at 512^3 pixels, or 1024^3 pixels. Stochastic methods have, to date, failed to provide such samples due to the large computational costs involved. Researchers in this field continue to try a large range of heuristic methods to attempt to get around that problem [EHF+05].

Researchers are interested in macroscopic properties of materials, whether they be mechanical, fluid transport, capillary action, etc. However, the problem comes down to this: merely knowing the volume fraction of the phases (e.g. density/pore) making up the sample will not be enough. The geometry and topology of the surfaces inside (i.e. the pore/density boundaries) are crucial.

Hence what is desired is a process by which, from limited available morphological information, a stochastic reconstruction will reproduce a good approximation of key properties of the material. There are essentially two approaches to stochastic reconstruction that have been pursued in this literature recently. The first method is to use the GMRF, samples from which are truncated to create the binary pore/density image [AJQ90, Ber87, RK96, Lev98]. The second method is based on Simulated Annealing (SA) [YT98, RU01, TTI02].

The first, GMRF based, approach has the immediate advantage that samples may be efficiently generated (see §4.3.2). However, this benefit comes at a cost: the approach cannot impose all the constraints one might wish. While first- and second-order constraints are possible (volume fraction and two-point correlation, respectively) are possible, this is not in general sufficient for reproducing the desired morphology [Lev98, Rob97]. This is a serious drawback.

SA, on the other hand, offers more freedom, as it is essentially model independent (at least in theory). This too, however, has a drawback. The computational costs involved have proven to be impractical so far.

Our motivation is to provide an improvement on the second approach. The first part of this chapter will mostly concentrate on the computational advantages of our method, but later discussion will turn to modelling issues.

In an important paper, Gidas [Gid89] introduces the idea of renormalisation to image processing contexts, and indeed addresses simulated annealing for the restoration of images in such a context. The current work is naturally related, but primarily differs in two important aspects. Firstly, Gidas discusses estimation, for example the denoising of an image. In such a problem, one wishes to construct an energy surface with the nominally *correct* image as a global minimum. In practice the problem is conditioned by the fact that in some sense we expect the correct image to be nearby in configuration space. In pure synthesis, by comparison, we are interested in exploring all minima of a complicated energy surface. Secondly, we are not only considering proper renormalizations of our configuration space, but rather we will be interested in the characteristics of a more general hierarchical approach.

In this chapter, we propose a Hierarchical Simulated Annealing (HSA) algorithm for the reconstruction of random microstructures from limited morphological information. The initial focus of this work is on the computational benefits of this approach. This will be followed by some discussion of modelling issues and opportunities in such a hierarchical scheme.

5.2 Difficulties with this approach

There are several problems with simulated annealing in this context. As mentioned, simulated annealing has previously been used as a method of constructing synthetic porous media with particular properties in this context [YT98, RU01, TTI02]. However, while initially enthusiastic, porous media researchers have largely given up on the approach for practical reasons. That is, the computational costs, which we address in this thesis, have been too high. See, for a recent example, [EHF+05].

5.2.1 Lack of Theoretical Support

There is a fairly fundamental disconnect between theory and practice in the reconstruction of porous media as found in the literature, but the area is not unique that way — researchers using SA for other purposes have encountered similar difficulties.

The first issue is that while there is a general result (i.e. for any appropriate energy function) for convergence of simulated annealing *with a sufficiently slow cooling schedule*, in many problem domains (including ours) such cooling is completely impractical. This problem drives researchers in the area to use 'fast cooling' techniques. Although the result gives us uniform convergence over the set of global minima of an energy surface *when correctly annealed with a logarithmic cooling schedule*, it has nothing to say about performance of faster cooling schedules. Thus while practicality has driven many to use such approaches (with such names as 'quenching', 'tempering' and other suggestive variations on the theme of metallurgical processes) the tendency is, except in specific circumstances, to lose the convergence (cf discussion of Hajeks result in 4.14).

There is a second problem that follows from this. In the context of the porous media 'sampling' application described here, this leads to a dishonesty if care is not taken. When speaking of 'sampling' from the Gibbs distribution, we are assuming we have actually reached the equilibrium distribution of the underlying MCMC chain, presumably via simulated annealing with sufficiently slow cooling. In practice, however, what is probably happening is that we are using the simulated annealing algorithm with a faster cooling schedule, as an optimisation procedure. This is a natural role for simulated annealing, of course, and in this practice will often allow low energy states of the system to be found. Calling these states, however, 'samples' from the underlying Gibbs distribution is simply incorrect, and calling them 'approximate samples' is problematic because there is little theoretical support to describe the distribution of such realisations.

5.2.2 Practical Difficulties

While in general, as previously noted, there are only convergence results for *logarithmic* cooling, in practice for this application the sheer expense of annealing dictates that other schedules are used. For example as previously noted a geometric schedule such as:

$$T_{n+1} = \alpha^n T_0, \qquad 0 < \alpha < 1 \tag{5.1}$$

is common.

The main practical difficulty is that, even with greatly accelerated cooling, the process may be far too computationally burdensome to be practical for systems large enough to be of interest.

It is this last point that had initially motivated the following work. Within the porous media literature exist several attempts at reconstruction which are essentially frustrated by the computational complexity of the method[1] (see, for example [TTI02, LIC00b, TTI02, Tor01, YT98]). Knowing that it is the sheer size of the configuration spaces involved and the computational complexity of running an MCMC sampler on them that is the problem, it is natural to ask if there is a multiscale approach that can help. Such an approach is developed in the following section.

5.3 Hierarchical Simulated Annealing

Recall that we will model our porous media images as realisations of a Markov/Gibbs random field. Hence for synthesis of images from this (prior) model, we wish to sample (but note above discussion) a density:

$$\pi_\beta(x) = \frac{e^{-\beta\mathcal{E}(x)}}{Z_\beta} \quad . \tag{5.2}$$

Here x denotes the image (i.e. current configuration), where $\beta = 1/T$ is the inverse temperature parameter, and $\mathcal{E}(x)$ is an energy function which will be constructed to represent an energy surface in which the low energy states have particular properties (more on this later).

However, also recall by the Hammersly-Clifford theorem that we may also consider this as a Markov Field. Intuitively, this may be an easier way to see where the difficulties stem from, holding in mind as an example the neighbourhood structures such as described in Figure 4.3. Note that because of the nature of the Markovianity condition (essentially the decoupling of the state of a pixel from all but a few of the other pixels) state changes outside a pixel's neighbourhood cannot effect it. Hence for any information about large scale structures to be propagated in the image, it must make its way around by way of a (potentially very long) series of local interactions. In terms of the underlying MCMC algorithm, large scale structures may only be constructed by a large number of (perhaps low probability) state

[1] Modelling issues also play a role.

transitions. This is reflected by the phenomenon known as *critical slowing down*.

So we see that the difficulty arises when the scale of structures of interest (in the sense of the energy surface favouring them, if not the path *to* them) lies outside the neighbourhood. Given this insight, it is natural to ask if you can rescale somehow in order to avoid this. As an illustration, consider Figure 5.2 showing various rescalings of two porous media images. Since we are rescaling binary images, we cannot average the results and still have binary images. The method of rescaling used is often-called *majority-vote*. This approach is a strictly dyadic rescaling, in which each pixel in the rescale image is the 'parent' of four neighbouring 'child' pixels in the original image. The value of the pixel depends on the count c of white children:

$$\text{pixel}(c) = \begin{cases} 1, & c \in \{3,4\} \\ 0, & c \in \{0,1\} \\ \text{probability } \frac{1}{2} \text{ of either value} & c = 2\ , \end{cases} \tag{5.3}$$

where black = 0 and white = 1 pixel values. This is the form of rescaling that will be used unless otherwise noted.

Figure 5.2 illustrates the way local (to a pixel) neighbourhoods local look quite different at different scales. Consider the relative likelihood of homogeneous white (or black) regions at 512 × 512 scale, compared with 64 × 64. For example, the neighbourhood local to a given pixel is much more often homogeneous in the high resolution images (a) and (b), than in the coarser (e) and (f) (where rescaling has made individual pixels easily discernible).

As another example, consider the (GMRF sample) image in Figure 5.3 showing one image from a training set and three successive samplings of this image. We can empirically examine the localised distribution of pixels by defining a neighbourhood structure and keeping track of all possible configurations within this neighbourhood. Figure 5.4 shows up to third order neighbourhoods and

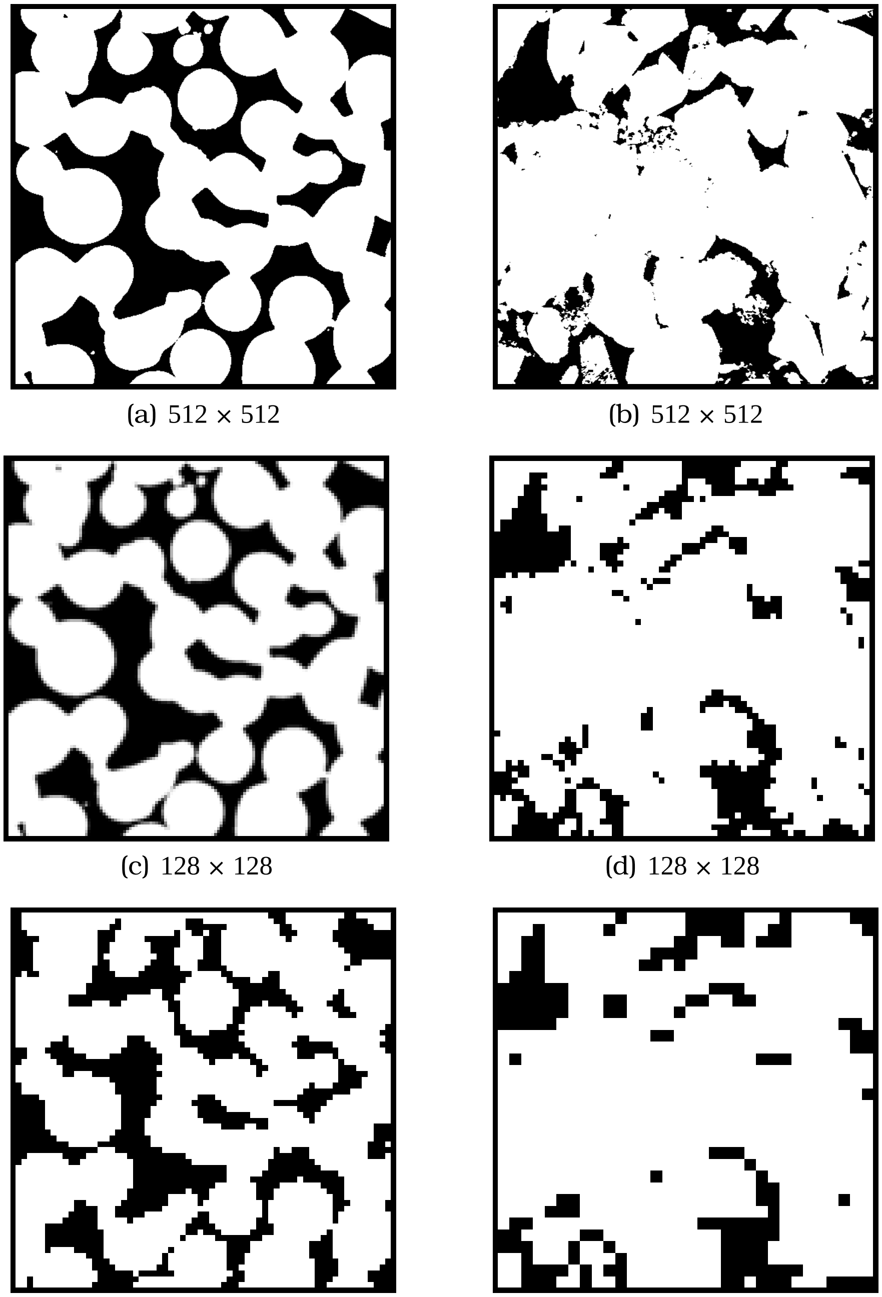


Figure 5.2: Lightly fused glass spheres (left column) and Berea sandstone (right column) at full resolution and two subsamplings.

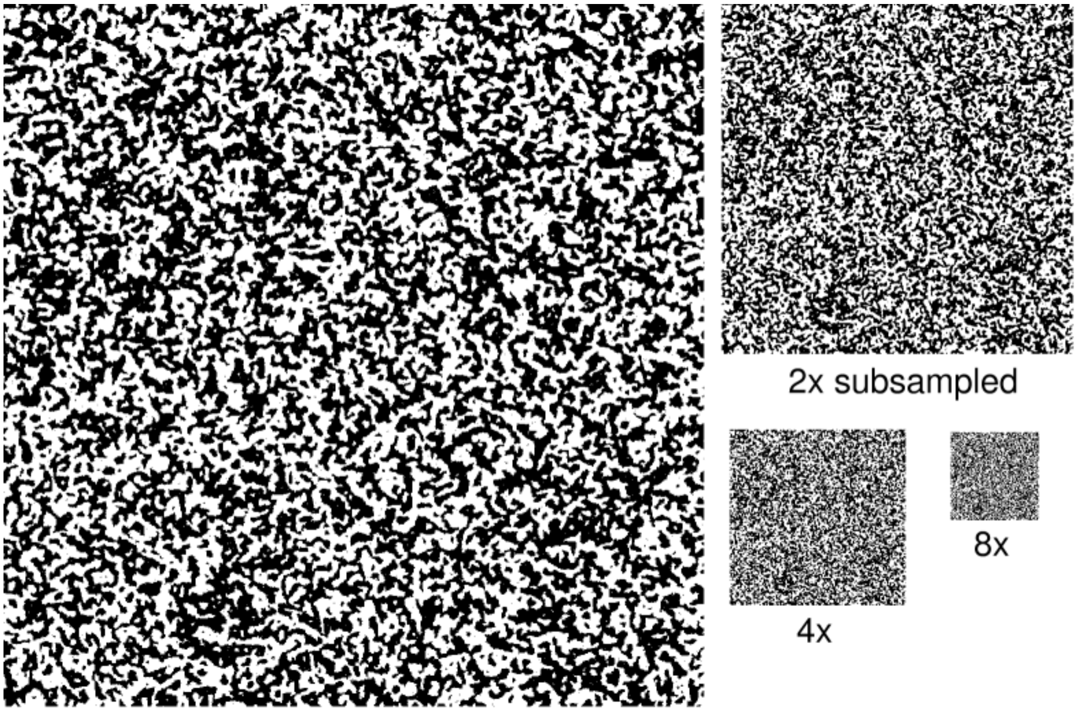


Figure 5.3: Image at several scales: original 1024x1024 image and 3 subsamplings

●	●	3	●	●
●	2	1	2	●
3	1	○	1	3
●	2	1	2	●
●	●	3	●	●

(a)

●	●	9	●	●
●	8	1	5	●
12	4	○	2	10
●	7	3	6	●
●	●	11	●	●

(b)

Figure 5.4: (a) Neighbourhoods and (b) bit indices up to 3rd order.

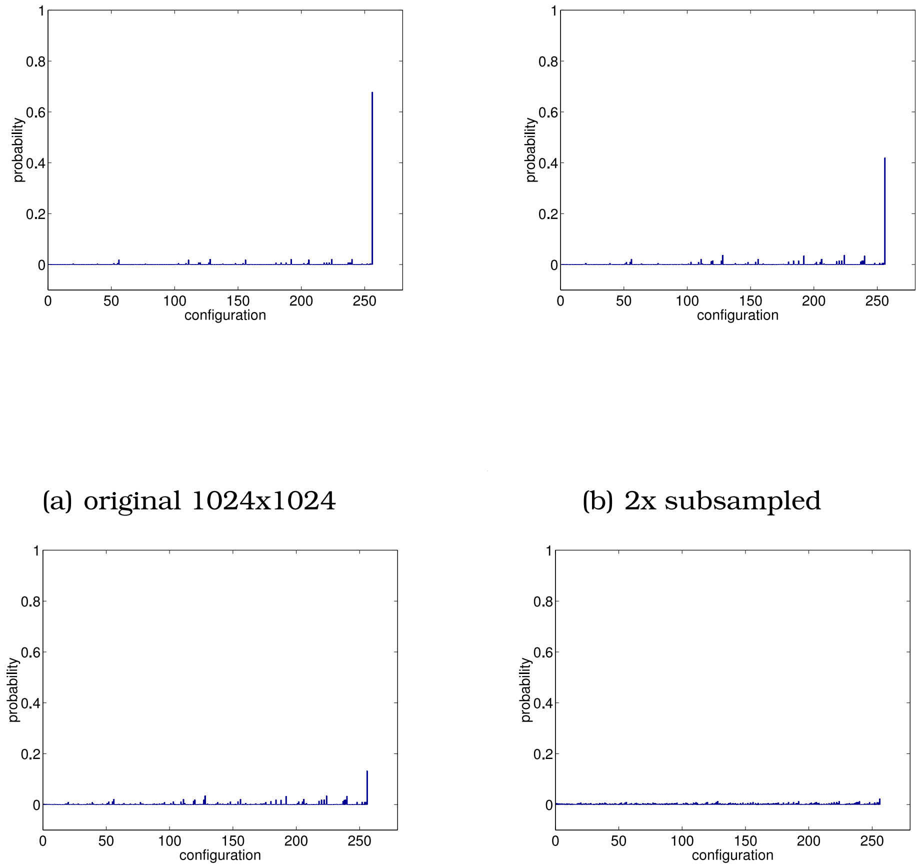
corresponding *bit-indices* to encoded configurations as the binary expansion of an integer. In this case, we have not indexed the central pixel, so statistics for white and black central pixel will be collected separately. Of course the ordering of such an expansion is arbitrary, but so long as we are consistent, we can usefully compare measured statistics of such configurations.

Figure 5.5, then, shows histograms (for the white central pixel case only) measured on a training set of 500 images (of which Figure 5.3 is one example). In these figures, note that the far right hand column of the histogram represents the homogeneous configuration of white pixels. As the subsampling becomes coarser, mass in the probability mass function (pmf) is spread more uniformly as the localised neighbourhoods start to become more often inhomogeneous. This sort of localised histogram will be revisited in §5.5.3 as a modelling approach.

It is also informative to look at the Ising model (4.9). Figure 5.6 shows the result of many runs for various values of the coupling constant $\beta = 1/T$, plotted against the *relaxation time* of structures of increasing size. In particular, each plotted line tracks the relaxation time of structures increasing in size by a factor of $\sqrt{2}$ of the plot below it. Here the correlation statistics are measured over time and the average decorrelation time is plotted for a particular scale and Ising parameter.

We propose attacking this problem by use of a hierarchy of scales, in a very natural way. The construction of this hierarchy comes from two key ideas. Firstly, there is a natural way to rescale our problem. Secondly, there is a relationship between temperature and characteristic scales of structure in the field.

To expand upon the latter point: Clearly there is no absolute stability of structures under MCMC sampling. However, structures of particular scales are metastable for particular temperature ranges. This is strongly related to the idea of *critical slowing down*, and its relation to the local nature of interactions. To wit, construction *or de-*



(a) original 1024x1024

(b) 2x subsampled

(c) 4x subsamples

(d) 8x subsampled

Figure 5.5: Trained histograms:white central pixel histograms

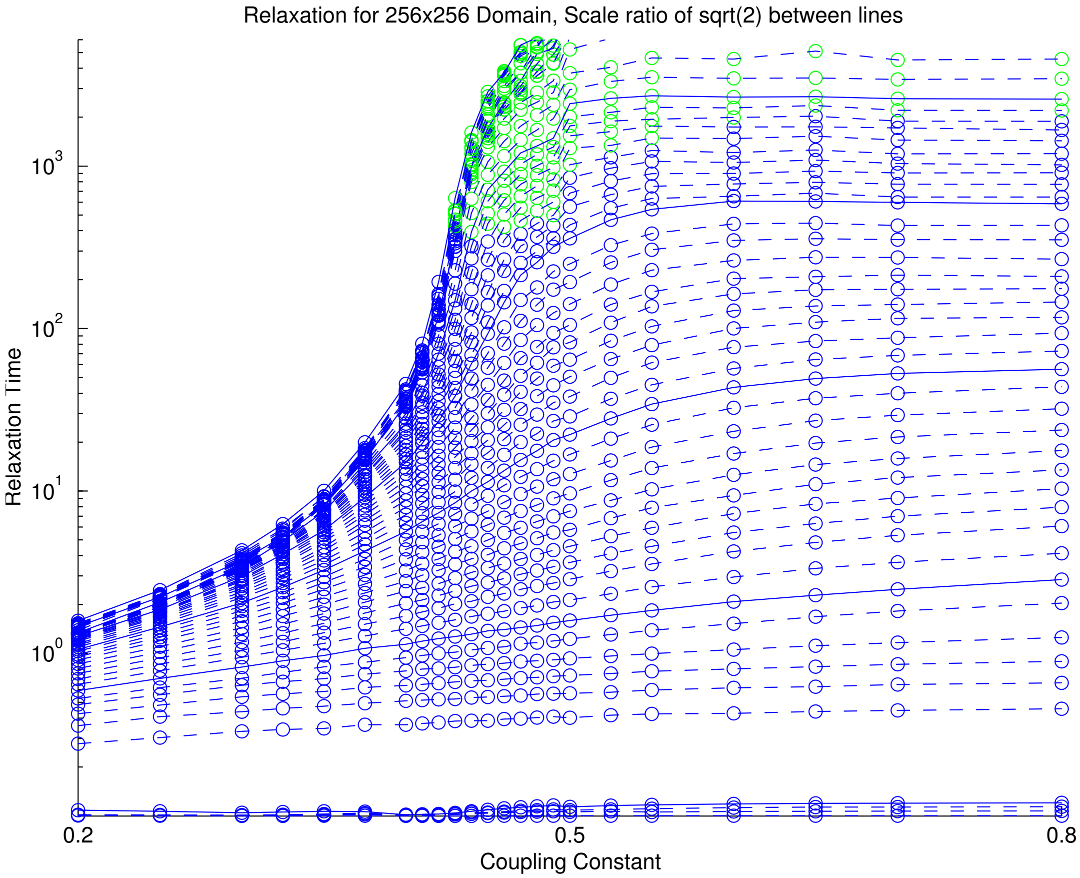


Figure 5.6: Ising model relaxation times *Figure provided by P. Fieguth*

construction of structures beyond the local neighbourhood structure of the field can require energetically non-favourable events, which occur only with a certain probability (i.e. transition probabilities in the Markov chain). The larger the extent of these structures, the more consecutive such events may be needed, leading (roughly speaking) to an ever decreasing product of probabilities. Hence large structures take a long time to build (but are more easily destroyed by multiple local interactions). A high enough value of the temperature parameter will simply destroy all structure in the field. Equally clearly, very low temperatures result in near gradient-descent behaviour, which will freeze structures in a local minimum energy configuration. As it turns out, analysing the effect of various cooling schedules is difficult [Bré98, GG84, SH87].

In considering a hierarchical approach, we want to concentrate on particular scales of structure at particular levels of the hierarchy. For appropriate temperature ranges we expect the following *relative to a particular scale*:

- Larger scale is ‘frozen’ (metastable)

- Current (or medium) scale tends to be ‘slushy’ at characteristic temperatures

- Smaller scale is rapidly changing

which is borne out empirically.

Taken together, these ideas suggest a natural hierarchical approach. To take advantage of these observations, we construct a hierarchical approach that will concentrate, at each level of the hierarchy, only on the medium scale structure, leaving the large scale structure intact (or at least not vary it inconsistently). We are not concerned about small scale structure as it is transient. Small structure becomes single or sub-pixel, while the current scale of interest is practically attainable with local interactions, not requiring large numbers of energetically unfavourable events. Algorithm 5.1 shows this process.

Algorithm 5.1 Hierarchical Annealing

```
$k \Leftarrow 0$
$X \Leftarrow$ initial state at coarse resolution
**for** scale $s$ from coarsest to finest **do**
  **while** $\mathcal{E}^s(X_s)$ not converged **do**
    $\beta \Leftarrow 1/T_k$
    $X_s \Leftarrow$ apply sampler to $X_s$
    $k \Leftarrow k + 1$
  **end while**
  $X_{s-1} \Leftarrow P_{s-1}(X_s)$    {project to next finer resolution}
**end for**
```

5.4 Computational Benefits

In this section we will discuss the benefits of HSA in reconstructing porous media. Before doing so, a description of the particular energy models considered is needed.

5.4.1 Models/Energy Functions

The purpose of this work is to describe the computational benefits made possible by using a hierarchical approach to simulated annealing. As such, discussion of the relative merits of particular models is eschewed. Rather, we present results on common models found in the appropriate literature, with little comment on efficacy or verification.

In particular, there are several components of energy functions that have been used [TTI02, LIC00b, Tor01] in related approaches, and we demonstrate how the increased performance of the HSA approach improves results with these models.

In general, our energy/cost functions will be made of one or more components. In the hierarchical case, these energy functions will exist for every scale s. Since we can look at separate scales, and also mixtures of these models, the general formulation will be

$$\mathcal{E}^s(x) = \sum_i c_i \mathcal{E}_i^s(x) \qquad c_i > 0 \ \forall i \quad . \tag{5.4}$$

Here for each scale s the energy $\mathcal{E}^s$ of a particular image x is given as the weighted (by coefficients c_i) sum of component energies $\mathcal{E}_i^s$.

Particular components of interest would be [YT98, TTI02, Tor01]:

- one-point correlation function
- two-point correlation function
- chord-length distribution
- lineal path distribution
- “pore size distribution”.

In this work we use a method of targeting of mean distributions such as described in [YT98, TTI02]. Note that in principle a modelling approach may seek to address each scale in (5.4) separately. This hierarchy allows approaches that are not possible with a single scale algorithm [AFV04a].

Quantities of particular interest in the study of porous media are the above mentioned correlation functions, as well as the distribution of chordlengths [TTI02]. Notationally, let us take $\hat{\cdot}$ to denote trained/target values. Furthermore, denote the lattice size of the image at a particular scale s as $\mathcal{O}(s)$ (then for initial lattice of $N \times N$ $\mathcal{O}(s) = N/2^s$).

We are considering binary dense/pore structures [TTI02] in an image. We can denote $I^s(x)$ for the index function for our binary image at scale s, (yielding 0 for pore, and 1 for density). If we let $\langle \cdot \rangle$

denote a spatial average over the image, then the average density ϕ (or "one-point" correlation) can be denoted as

$$S^s(r) = \langle I^s(x+r) \rangle = \phi^s \ . \tag{5.5}$$

Similarly, two-point correlation:

$$S^s(r_1, r_2) = \langle I^s(x+r_1) I^s(x+r_2) \rangle \ . \tag{5.6}$$

One-point energy function: With the above notation, $\hat{\phi}^s$ denotes target values for the image density at scale s and we can express an energy function for this scale based on one-point correlation as

$$\mathcal{E}_1^s = \|\hat{\phi}^s - \phi^s\| \ . \tag{5.7}$$

Two-point energy function: If we restrict ourselves to the horizontal and vertical directions (to reduce computations), two-point correlation gives us

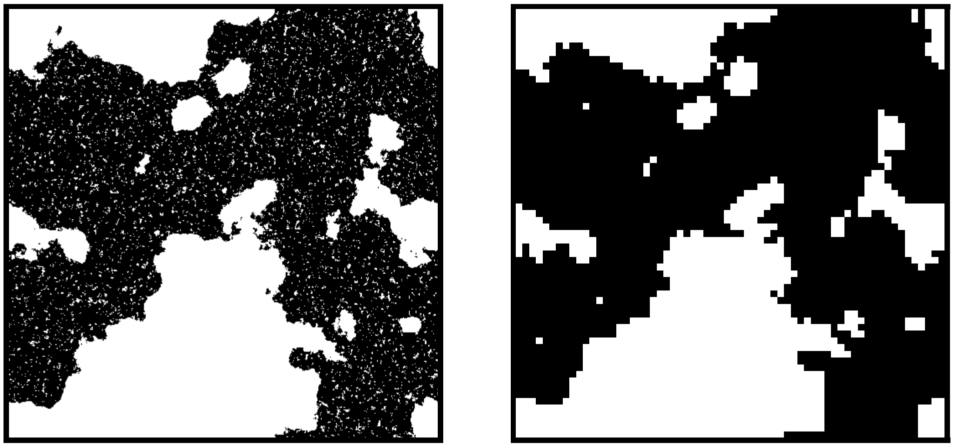
$$\mathcal{E}_2^s = \sum_{r=1}^{\mathcal{O}(s)/2} \|\hat{S}^s(0,r) - S^s(0,r)\| + \|\hat{S}^s(r,0) - S^s(r,0)\| \ . \tag{5.8}$$

Chordlength energy function: If we again restrict ourselves to the horizontal and vertical directions again, this is essentially the distribution of length of contiguous "runs" of density pixels in these directions. Denoting these probability mass functions as p_C^h and p_C^v for the horizontal and vertical directions, respectively, we may construct an energy function:

$$\mathcal{E}_3^s = \sum_{n=1}^{\mathcal{O}(s)} \|\hat{p}_C^h(n) - p_C^h(n)\| + \|\hat{p}_C^v(n) - p_C^v(n)\| \ . \tag{5.9}$$

Here the sample pmf's are estimated by histograms from the image data.

Note that the norm used in the various energy functions is not specified — there are several possibilities in most cases. In the numerical experiments reported here, the l_2-norm is used, but the best choice is not clear. One could argue that the correct way to view the difference in probability distributions, at least, should be via the Kullback-Lieber divergence, but for optimisation this is not the clear choice.



(a) 512 × 512

(b) 64 × 64

Figure 5.7: GMRF data generated with two characteristic scales, as shown in a). A strong bias toward black pixels in the shorter length scale GMRF has the effect that after several rescalings, the coarse resolution images such as b) are much simpler.

5.4.2 Experiments

Synthetic Data

Physical porous media samples tend to have complicated morphology, and it can be difficult to evaluate the performance of a sampling algorithm (for recent approaches, see for example, [AKR04]). In addition to such data, then, it is informative to construct synthetic data to emphasise the benefits of a hierarchical approach.

First, we consider a set of 300 images generated by means of two GMRF's (for ease of sampling), and exhibit two characteristic length scales. Figure 5.7 shows an example of this data set,

This data allows us to concentrate on two characteristic length scales and on a morphology simpler than may be found in some physical samples. This set is particularly illustrative of the difficulty in convergence of *flat annealing*. Intuitively, this is easy to see in the evaluation of chord-length distributions. At the highest resolution, there are small isolated chords, some small chords on the edges of large features, and longer chords comprising large features. In the

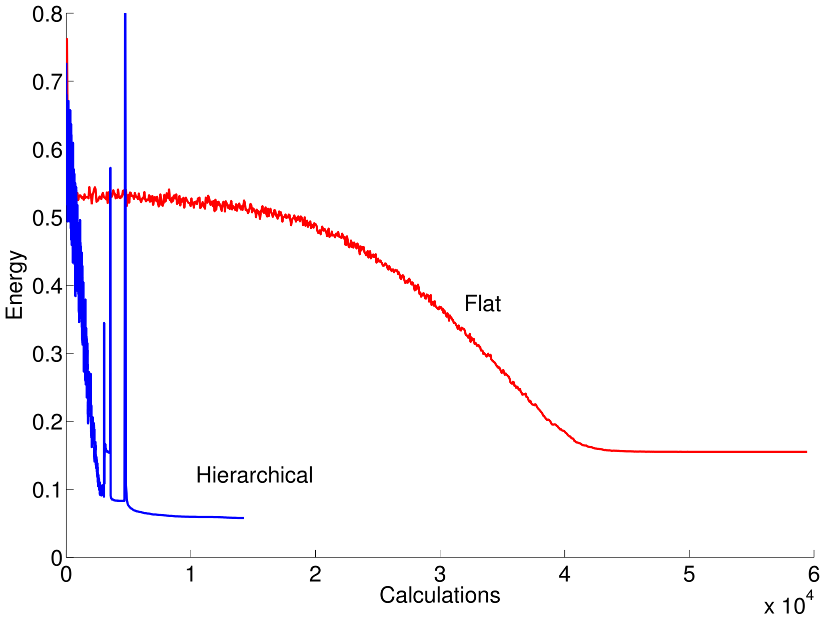


Figure 5.8: Convergence flat (red) and hierarchical (blue) annealing for GMRF data set. Note spikes in HSA profile due to projection.

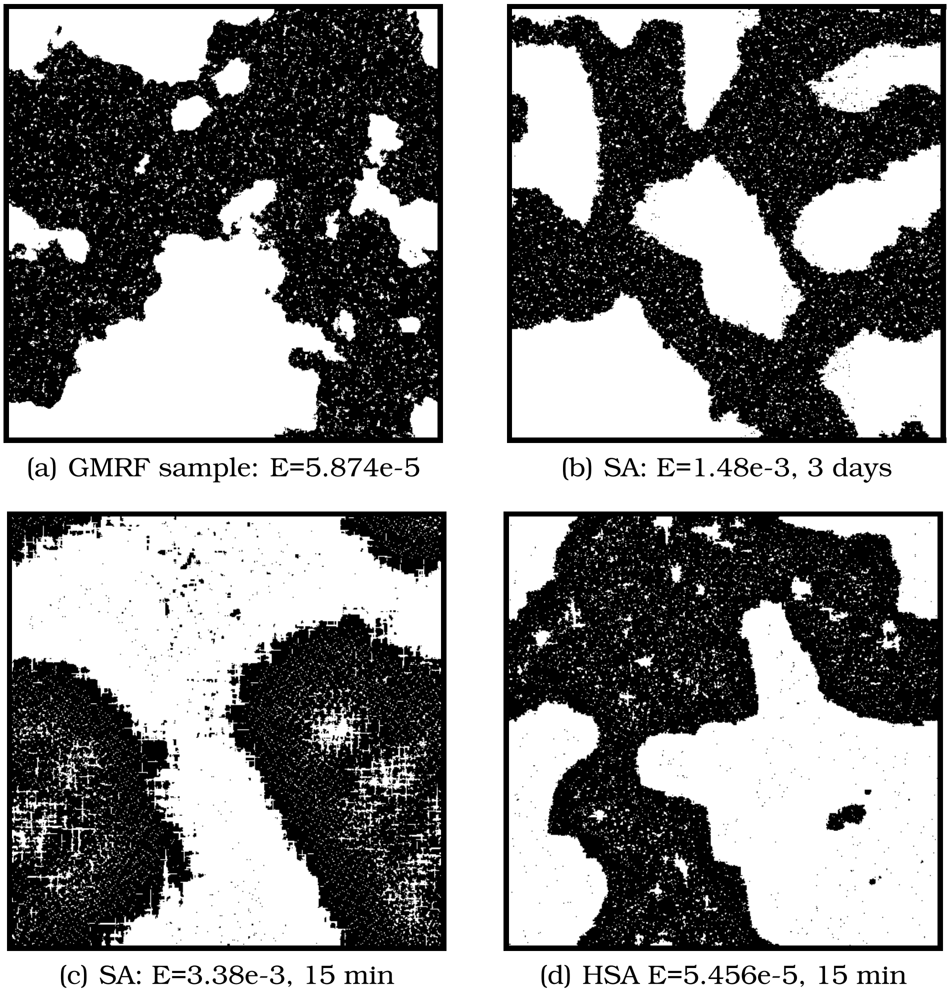
annealing algorithm, moving between isolated small chords through isolated medium length chords to eventually collate into larger structures is not energetically favourable, so we have *critical slowing down* as structure begins to emerge. By comparison, coarse levels of the hierarchical annealing do not have any small isolated chords, so we may expect convergence to be quicker. Indeed, empirical results support this conjecture. Figure 5.8 compares flat and hierarchical sampling runs on a log-log plot of energy vs. computations. The energy 'spikes' in the hierarchical annealing curve are due to projection. Immediately after projection, local configurations are high energy due to artifacts of the projection. Since these high energies are due to local configurations, they are easily remedied and energy levels immediately drop as the sampling algorithm progresses.

Figure 5.9 shows 512×512 images sampled from this model by three methods: the best 'flat' SA result allowing about 3 days computation, the best HSA result, taking about 15 minutes, and the best SA we could do (i.e. with a different cooling schedule) in the same number of computations as the HSA sample. As can be seen here, and in a 'zoom' view given by Figure 5.10, the complex morphology is not represented in the SA samples. The long-run result lacks medium size structures, and the fast SA sample is very poor. The HSA result, while clearly less than perfect from a modelling point of view, shows both visually and by final energy a much better result.

Computational Benefits

Separate from issues of convergence (analysis of which is difficult for non-logarithmic cooling schedules, and more so where multiple scales are concerned), there are clear sources of expected computational benefit from the HSA approach. At each level of the hierarchy, we are running a Gibbs-type sampler in a smaller configuration space than at the following levels. In the cases described here, each level in the hierarchy reduces the image size by a factor of four (in 3D the gain is much larger, of course), so we have a geometric reduction in the size of the configuration space. Beyond this simple reduction, there are data-dependent benefits. Since large scale structures are built by local interactions, the probability of constructing something when the intermediate steps are not energetically favourable is reduced as the the number of steps increases (i.e. the product of the probabilities of individual steps). In the hierarchical approach, large structures are put together *with fewer intermediate steps* at a coarser resolution. As this is data dependent, it is difficult to quantify, however the less probable the intermediate steps are (in some sense, the more *critical slowing down* experienced) the more it will benefit from a hierarchy.

Evidence of both of these effects was already presented in Figure 5.8 for our synthetic data. Figure 5.11 shows the results of many hierarchical runs with a simple parameterisation on the cooling



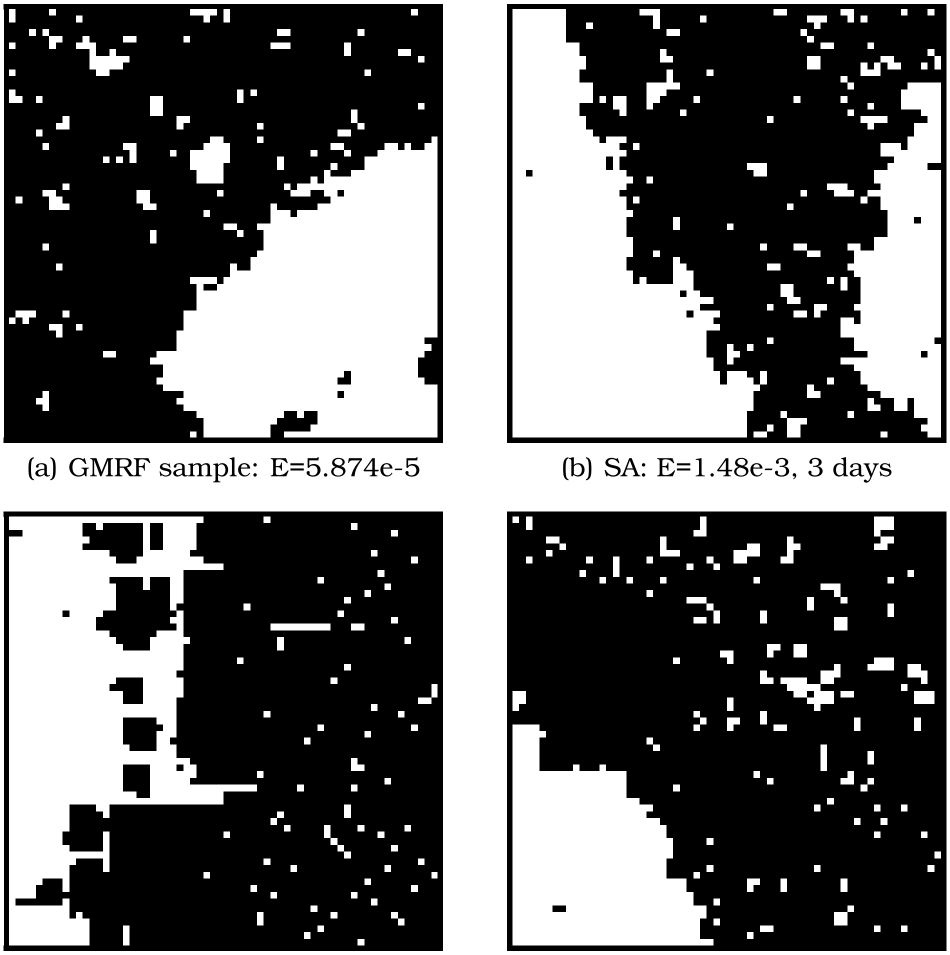
(a) GMRF sample: E=5.874e-5

(b) SA: E=1.48e-3, 3 days

(c) SA: E=3.38e-3, 15 min

(d) HSA E=5.456e-5, 15 min

Figure 5.9: GRF training data. Each panel gives the final energy, E, and approximate runtime (on a 3Ghz Pentium IV class machine). Shown are a) example training data, b) 'best case' sample from SA with very slow geometric cooling c) sample from SA with equivalent computation to the hierarchical case, and finally d) sample from HSA method. Model used was twopoint probability and both white and black phase chord-length distributions. Images are 512×512 pixels.



(a) GMRF sample: E=5.874e-5

(b) SA: E=1.48e-3, 3 days

(c) SA: E=3.38e-3, 15 min

(d) HSA E=5.456e-5, 15 min

Figure 5.10: Zoom views of images shown in Fig 5.9. Here we see up close the sort of artifacts rapid cooling creates, and more subtle differences between the HSA and SA results, when compared to the GMRF sample.

schedule [AFV04b]. In particular, the parameterisation was formed of the cooling schedule parameter α, the number of sweeps at a given scale, and the temperature at which the following scale was initialised. After completing sweeps at a give scale, the current state (image) was projected to the next finer resolution (in the usual way, i.e. one white pixel in the current scale becomes four white pixels over the same spatial area in the higher resolution, and likewise for black pixels). The initial temperature for the new resolution was then set (according to the current parameterisation), and the process begins again.

In these simulations, the data set was Berea sandstone, and the model was combined twopoint correlation and chord-length distributions. It is important to note that the parameterisation allowed for poor choices of cooling schedule (i.e., that would allow high enough temperatures to tear apart structure after projecting to a high resolution), and these paths are especially evident at the top-right of the ensemble. Many parameterisations allow for very good performance, as evidenced by Figure 5.11. This figure compares several HSA results to geometric cooling schedules (5.1) for flat simulated annealing at the highest resolution but with various values for the parameter α (cooling rate). As expected, increasing α improves the final energy at the cost of more computations. Both figures are presented with log-log scales, so demonstrate a computation vs. energy gain of more than an order of magnitude.

As noted in the previous section, an interesting demonstration of the slow convergence of some models is an energy function made up of both positive (white) and negative (black) chord-length distributions. When attempting to build large scale structures, such as in our GMRF data set, there is very little energy difference made by any intermediate step. In some sense the white and black chord distributions are at odds, hence very slow convergence results. While this is definitely *not* a good model to capture the morphology in question, it is interesting that even allowing many days, flat annealing will not

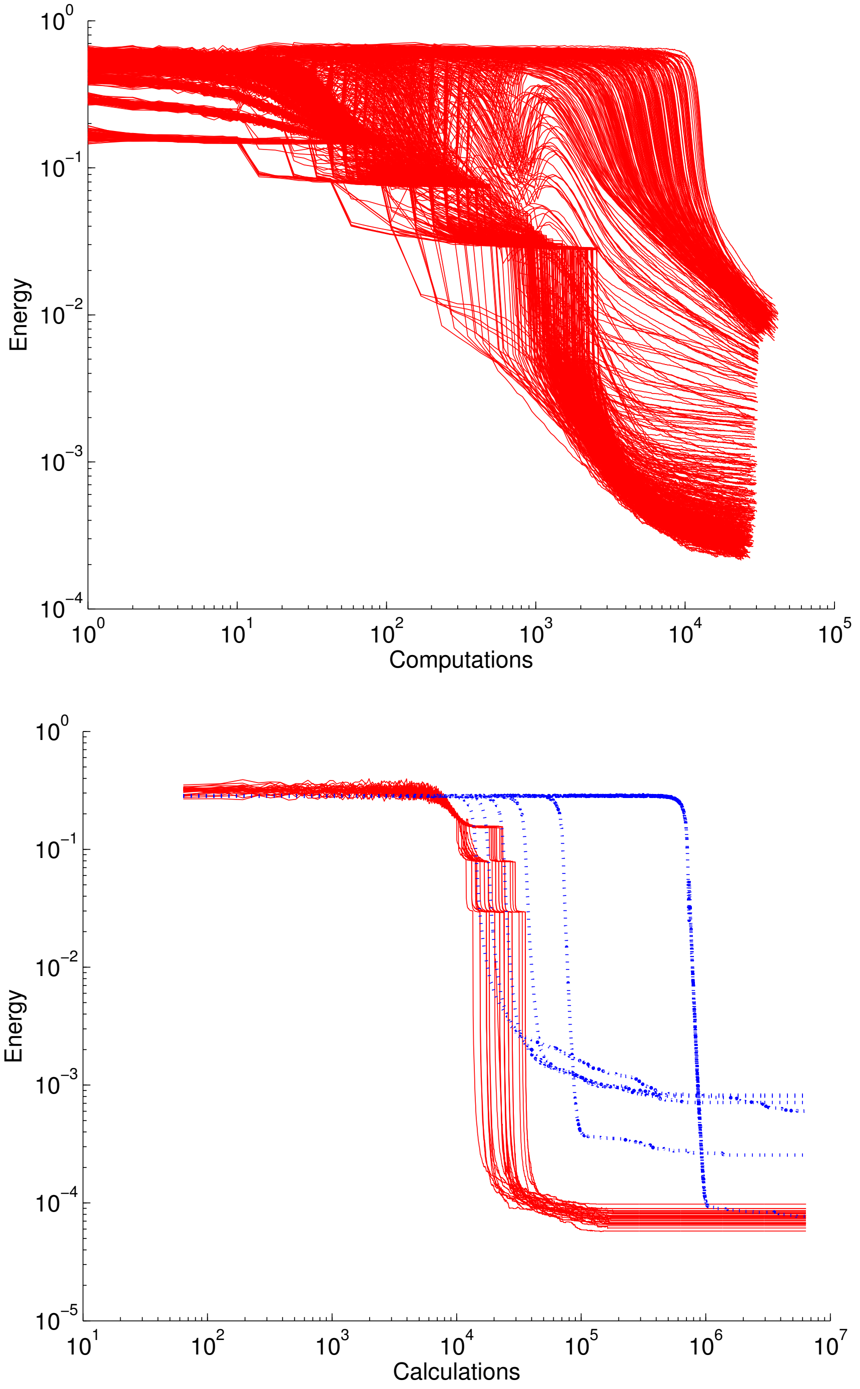


Figure 5.11: Energy vs. Computation: Top figure shows results for many parameterisations of a simple chord-length model. Lower figure gives several (good) hierarchical annealing runs with annealing (dashed lines) for various cooling schedules (increasing α trades more computations for better final energy).

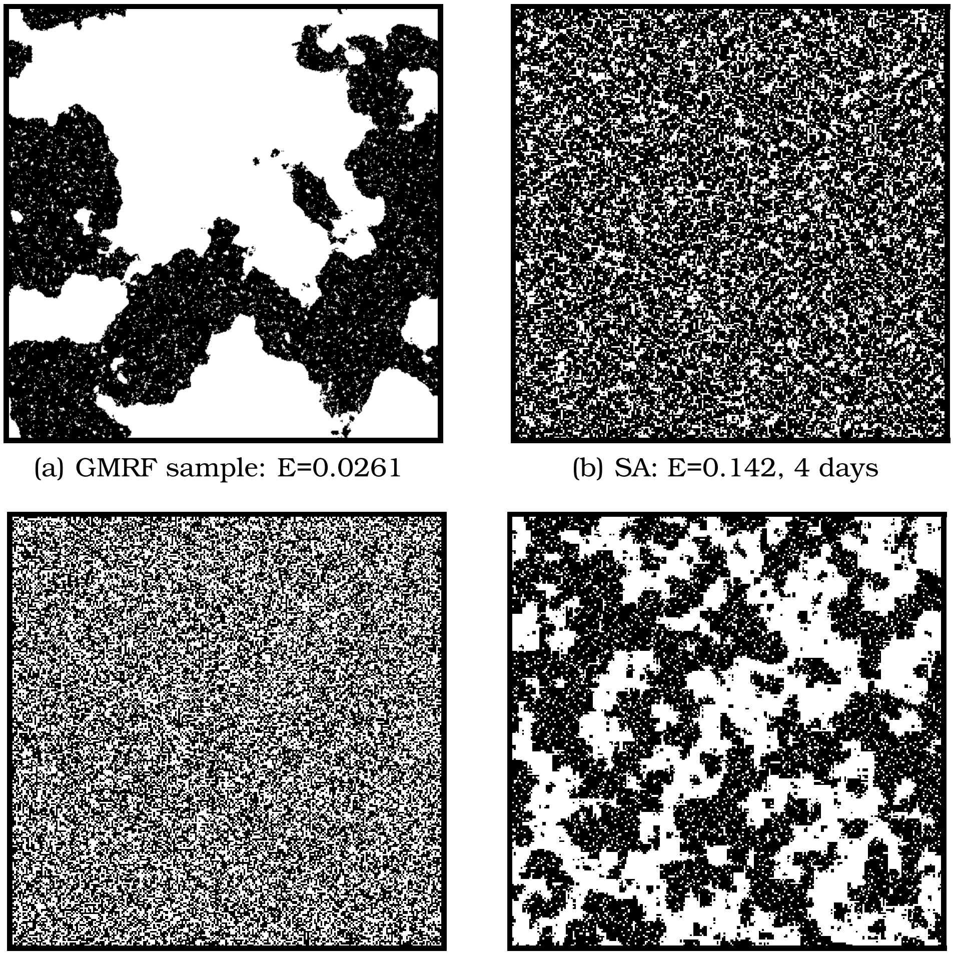
converge at all usefully, while hierarchical-annealing generates large scale structure in a few minutes. Figure 5.12 demonstrates this.

Physical Data

While empirical results on the synthetic data described in the previous section are valuable, it is of interest to apply the approach to real data. It is worth re-iterating that the purpose of this current work is *not* to address modelling issues in the problem domain. For this reason validation of resultant images is difficult, at least in an absolute sense. While subtle issues of validation are beyond the scope of the paper, in comparison to flat annealing, improvements can be *extremely clear*.

Figure 5.13 shows results of the same computational methods as Figure 5.12 applied to a data set of lightly fused glass beads. Note again the complete failure of convergence for the flat annealing cases. In comparison, Figure 5.14 shows similar samples, but for a more realistic model, of chord-length distribution and twopoint correlation. This is *still* not a sufficient model to capture the morphology of the training set. We can see, though, that the samples are drawn from a class of images with chord-length distributions that are strongly in agreement with the mean chord-length of the training set, as Figure 5.15 demonstrates.

The process under the HSA algorithm is converging to something with reasonably low energy under this model. Of course, this in no way addresses the question of whether or not the mean distribution is the correct way to model these images.



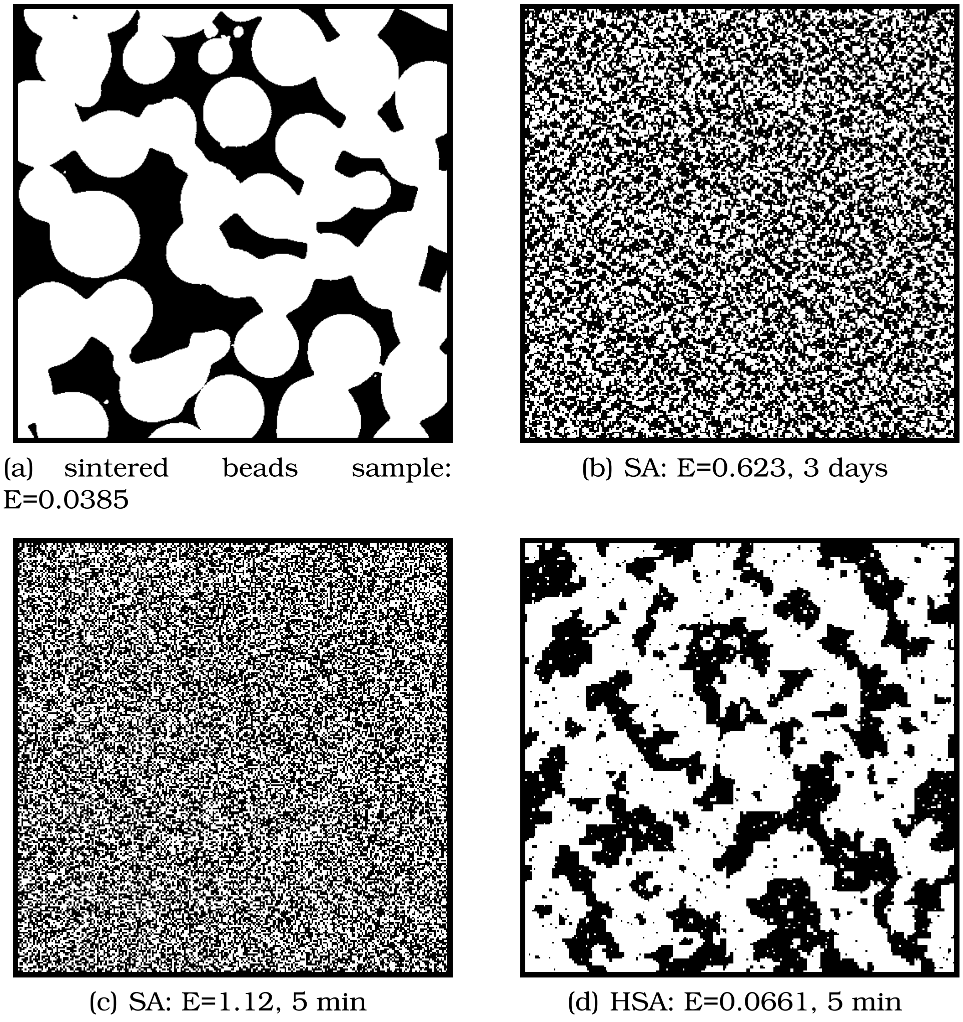
(a) GMRF sample: E=0.0261

(b) SA: E=0.142, 4 days

(c) SA: E=0.510, 15 min

(d) HSA E=0.0261. 15 min

Figure 5.12: Similar to Figure 5.9, these panels show results for the (more difficult) case where the energy function is a mix only of white and black phase chordlength.



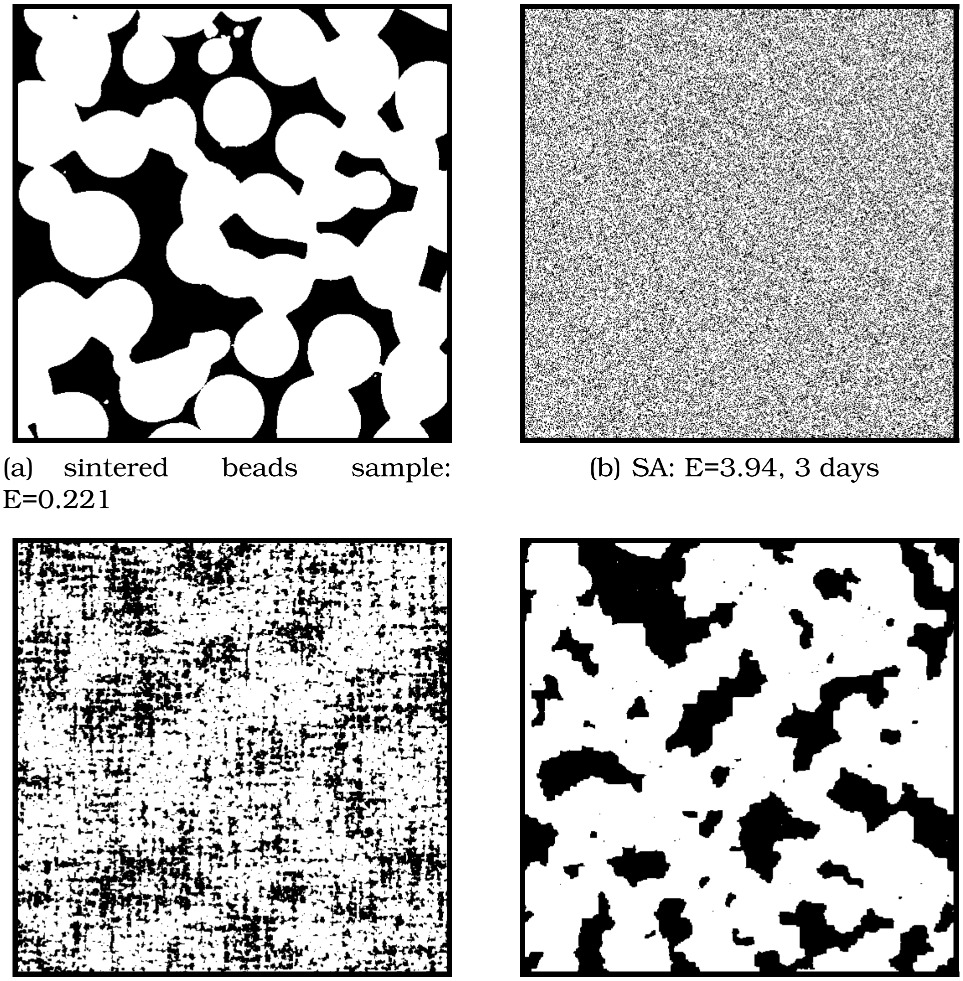
(a) sintered beads sample: E=0.0385

(b) SA: E=0.623, 3 days

(c) SA: E=1.12, 5 min

(d) HSA: E=0.0661, 5 min

Figure 5.13: Sintered glass beads data set. Again, each panel gives the final energy, E, and approximate runtime. Shown are a) example training data, b) sample from SA with very slow geometric cooling c) sample from SA with equivalent computation to the hierarchical case but with enforced volume fraction (to the mean of the training set), and finally d) sample from HSA method described herein. Model is (cf Figure 5.12) both white and black phase chord-length distributions. Images are 256×256 pixels.



(a) sintered beads sample: E=0.221

(b) SA: E=3.94, 3 days

(c) SA: E=11.0, 15 min

(d) HSA: E=0.483, 15 min

Figure 5.14: Sintered beads data set with a slightly more realistic energy function, the mixture of twopoint probability and chordlength distribution. In this case the images are 512 × 512.

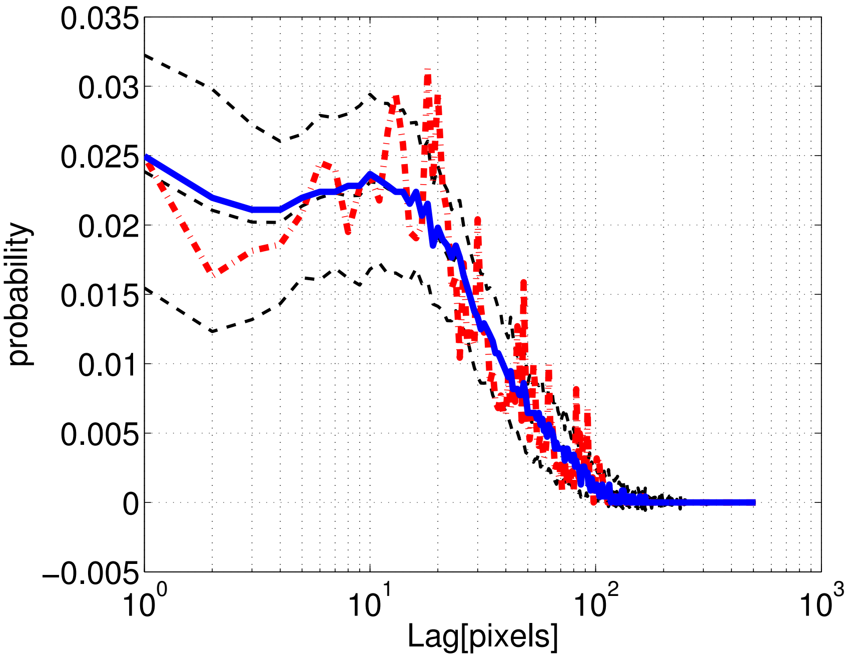


Figure 5.15: Chordlength distributions for spheres data set (shown in Fig 5.14). The dashed lines give mean chord-length distribution over training set, and one standard deviation. The solid line is final distribution of an HSA sample, while dot-dash line shows a typical sample from the training set.

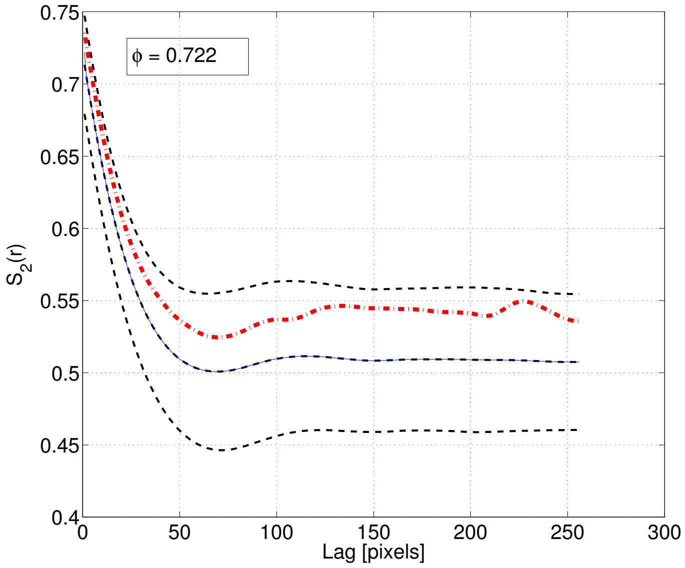
5.5 Modelling Issues

In this section several issues with the construction of energy functions, and the use of simulated annealing, will be discussed. Problems of both a theoretical and practical nature will be described. There are two key issues.

5.5.1 Standard Methods

The choice of energy functions in this context has largely been driven by a desire to match certain statistics in the reconstruction of porous media [YT98, RU01, TTI02, OI04, RIH$^+$04, LIC00a]. Hence, in order to achieve this the tendency has been to construct an energy function based on targeting the mean value (over some training set) of the particular distribution of interest. Indeed, with HSA we can achieve this quite well, at least in the case of energy functions that target only, say twopoint probability functions. Figure 5.16 shows an example: Here we have constructed an energy function based on ℓ^2 error relative to the (pointwise) mean twopoint probability function of the training set. The reconstructed image has extremely low error (on the order of 10^{-6}, essentially the histogram accuracy given the image size). On the plots the HSA result is indistinguishable from the mean value except for the very shortest lags, and then only in the log-log plot! This discrepancy is because the target volume fraction of $\phi = 0.722$ differs from the reconstructions value of 0.718. On the other hand, the dashed line gives a typical result from one of the training images.

While reaching such low errors, much better than one standard deviation, demonstrates the success of this method, Figure 5.16 is troubling. From a somewhat philosophical point of view, the concern is that our reconstruction process could not conceivably result in one of the training images. By construction, we will have the situation that because of the variance of the data set, each of our training images will have relatively large error compared to reconstructions



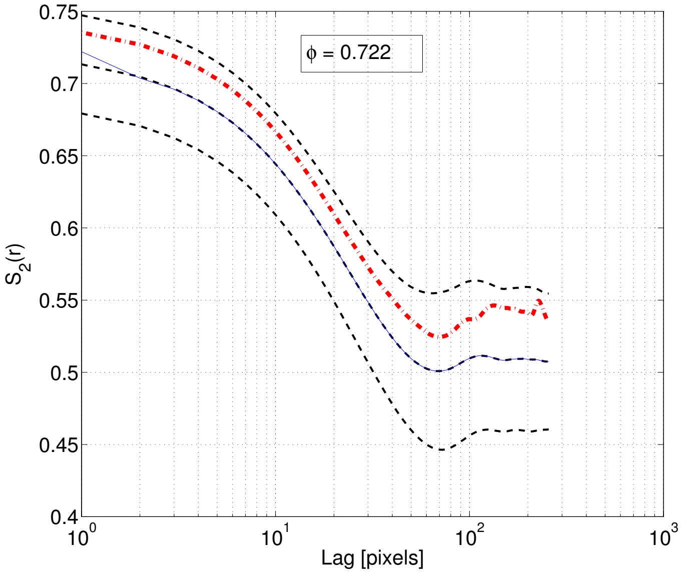


Figure 5.16: Twopoint probability functions for spheres data set (shown in Fig 5.14). The dashed line is (pointwise) mean value over training set, and dotted lines give one standard deviation. The solid line is final distribution of an HSA sample, while dot-dash line shows a typical sample from the training set.

(at least if the annealing process is working).

In practice, energy functions are more complicated and have two or more components, which, again in practice, tend to result in more error as the various components “fight” in the optimisation process. One could consider this to exacerbate the problem of over-fitting to mean values, but it doesn’t really.

Perhaps more troubling though, is that the pointwise averaging tends to result in a much smother curve than any of the training data. So not only are we optimising to reach a class of images that doesn’t included the training data, the twopoint correlation is characteristically different from all the training data!

The same issues exist with chord-length, as shown in Figure 5.15. Indeed, it is clear that this is a general weakness of the approach.

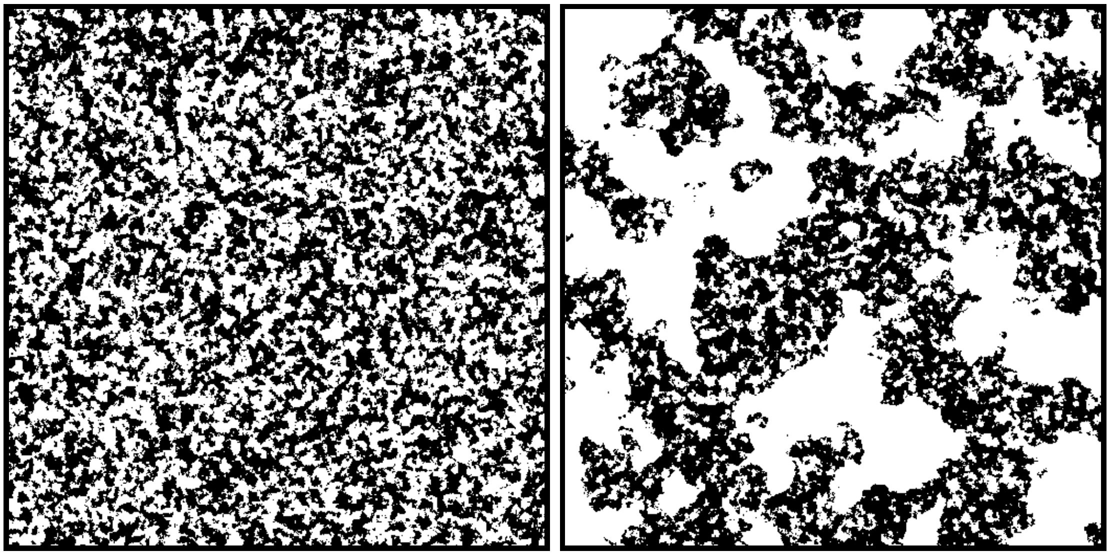
In a perfect world, what we would like to be able to do is to sample from the joint distribution over appropriate spaces of functions describing these various distributions. Once such a sample had been drawn, the HSA process could be used to optimise toward a reconstruction that then would, assuming convergence to reasonably small error, have the appropriate statistical properties.

It is clear that a naive approach to such a method is intractable, since the amount of data needed to get a good estimate of the joint distribution won’t be available, even if other problems are solvable. However, the flavour of this idea may be a way forward, and perhaps deserves more thought.

Failing such a method, it is clear that using descriptors such as twopoint correlation and chord-length distributions to *validate* (at least in some sense) the reconstructions arrived at by other means is sensible. This leads to the question of whether or not we can improve on these energy functions. A broad treatment of that question would necessarily require deep reflection on the nature of porous media, and what sorts of physical properties are desired, etc. Although that could be the subject of quite a different thesis, it is clearly beyond the scope of this one. However, there are some fundamental issues that may be addressed.

5.5.2 Discrimination

Returning to synthetic data generated by GMRF methods (as described in §5.4.2) we may construct a data set with two characteristic length scales of structure, as shown in Figure 5.17. An important point is that the small scale structure is statistically identical in the two images. Now if we measure the chord-length distribution for white phase pixels in the horizontal direction, we have the results shown in Figure 5.18 With linear scaling (the left hand figure) we see that it is difficult to discriminate between the two cases! When plotting on a log-log scale (right-hand figure) things become at least visible.



(a) single scale

(b) two scale

Figure 5.17: (a) single- and (b) dual-scale samples generated by GMRF

There are two crucial difficulties here. The first is that, while there is some difference at long lags (as might be expected), it doesn't account for much mass. Hence any sort of divergence on these distri-

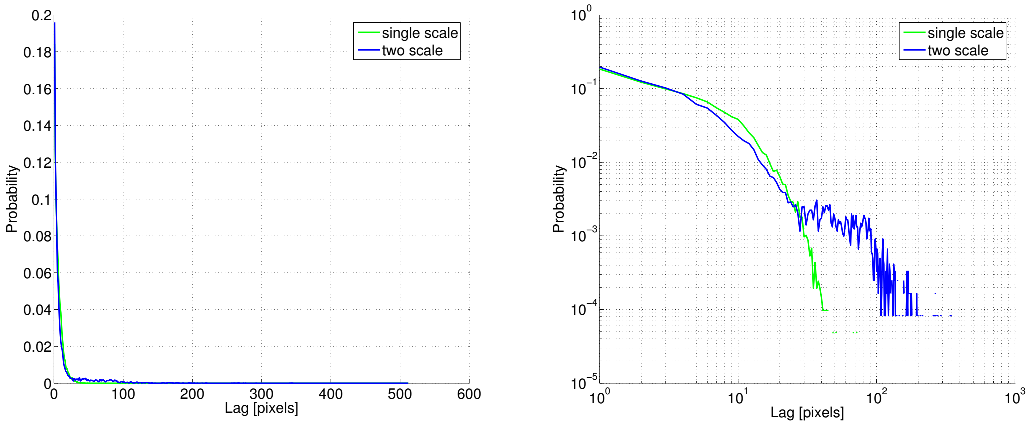


Figure 5.18: Chordlength distribution compared for single- and dual-scale samples generated by GMRF

butions to form an energy function will not emphasise the difference. One may counter this with the observation that for particular samples then, a 'customised' energy function could be constructed, which is true. However, throwing away the general nature of the method doesn't solve the more fundamental issue.

The second problem then, is this. Nearly all of the difference between these two distributions lies at the longer lags. This is intuitive, due to the way the data set was constructed. However, *any* changes made by a simulated annealing method must be made by a localised process operating on individual sites – in this case 'flipping' the pixel values. So in a very real sense, by constructing this energy function, we are working "against the grain" of the process. Given enough time, a simulated annealing algorithm will be able to find its way around such an complex and difficult configuration space — but that is computation time we won't have.

Even worse, we can construct data sets that are hardly discriminated at all by this energy function and yet are quite different.

Biased GMRF data

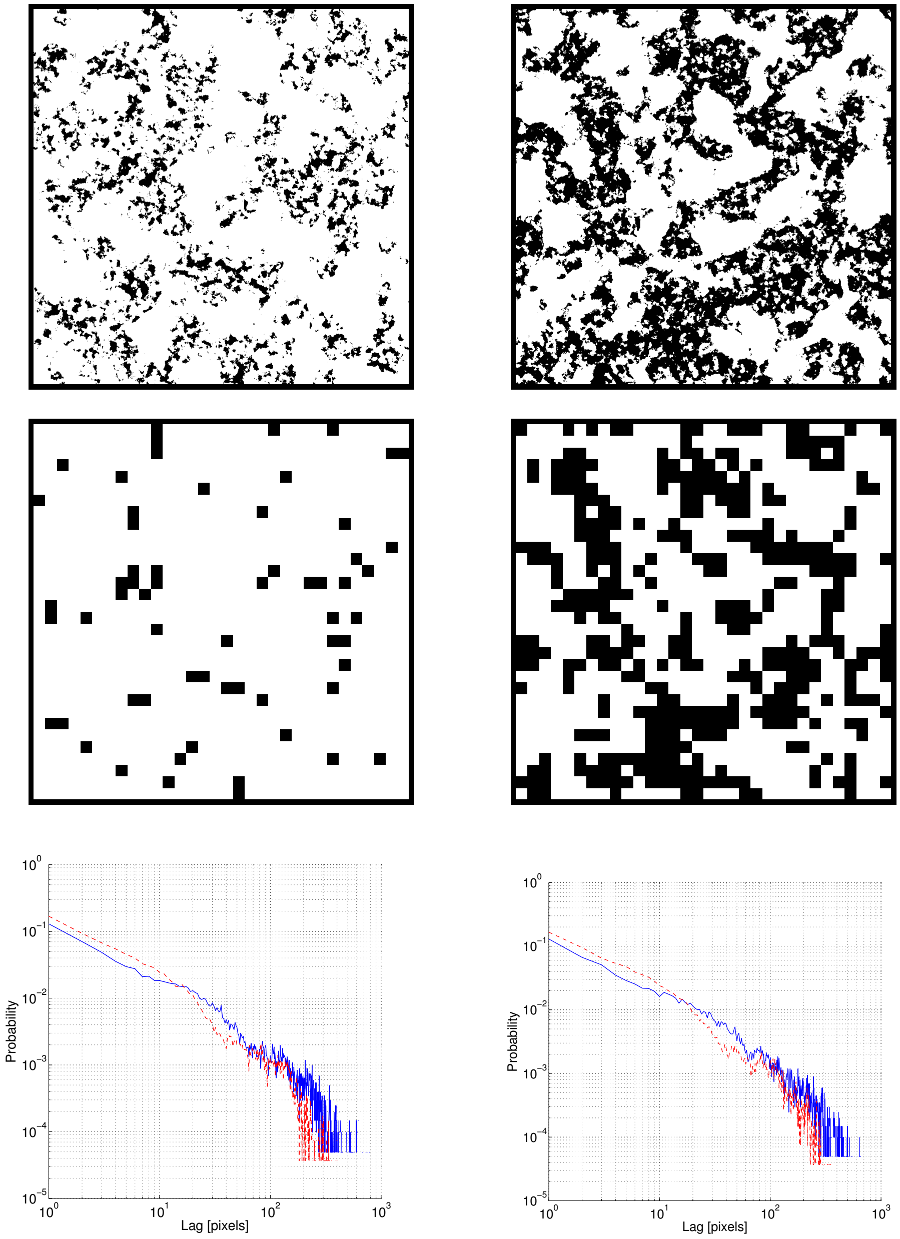
Consider the case of two synthetic data sets constructed analogously to the one described in §5.4.2. Again there are two fields, representing two 'natural scales' of structure in the image. The large scale structure is identical (in the sense of being generated by the same field), but the density of the small scale structure is slightly biased in positive or negative direction, respectively, for the two sets. Recall that the images are generated by thresholding Gaussian Markov random fields (to result in binary images) so simply moving the threshold will adjust the white/black proportions.

Figure 5.19 shows examples. The left column of images is positively biased data, and the right negatively biased. The first row contains 512×512 examples of the data sets. The second row shows rescaled versions of these, at 32×32. The third and final row shows the comparison of chordlength distribution in both black and white phase, for both cases. Clearly it is difficult to discriminate based on these distributions (note again the log-log scale of these figures) but the images, and particularly the rescaled images, are dramatically different! So clearly models like this have difficulty discriminating some features. One thing we would like, then, is a model that addresses this problem.

5.5.3 Local Histograms

One approach is to revisit the 'local neighbourhood distribution' discussed previously. In an analogous way to what we have been doing with chordlength, etc. we can build an energy function based on these distributions. This approach is discussed in [AFV03a, AFV03b].

Recall from §5.3(see also Figure 5.5) the investigation of the distribution of pixels in a local neighbourhood. Naturally, we can build an energy function similar to those seen previously by using histogram estimates of the probability distribution of local neighbourhood configurations. To recap: use bit indices such as (again) shown in Fig-



(a) black phase

(b) white phase

Figure 5.19: Two GMRF samples biased in opposite directions, with rescaled versions, and chordlength distributions (for the full resolution). The lower plots compare chordlength distributions for positively-biased (left column on the images) being the dashed line and negatively-biased being the solid line. This is done for chordlengths in both white phase (a) and black phase (b).

ure 5.20.

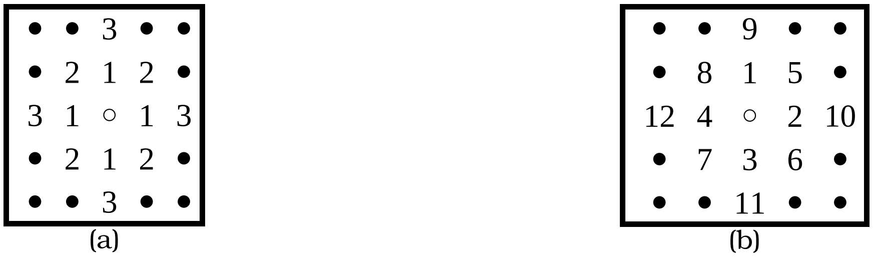


Figure 5.20: (a) Neighbourhoods and (b) bit indices up to 3rd order.

If we let $b = |\mathcal{N}_k|$ be the number of pixels in this neighbourhood, then because each pixel is binary, the size of the set of all possible configurations is 2^b. For reasonably small k, it is computationally feasible to count the instances of each local configuration in an image. For this purpose, a natural bijective mapping $\mathcal{F} : \mathcal{N}_k \to 0 \ldots 2^b - 1$ is arrived at by labelling each pixel in the neighbourhood uniquely from 1 to b, and treating the m^{th} pixel state as the state of the m^{th} bit in a b bit binary representation of an integer in $0 \ldots 2^b - 1$.

For some class of bitmap images, we can consider the global distribution of local configurations. Designate target pmf's for the two cases of white central pixel and black central pixel:

$$p_w[n],\ \ p_b[n] \qquad n = 0 \ldots 2^b - 1. \tag{5.10}$$

Given these target distributions, maintain sample statistics for the current configuration x in the form of histograms:

$$h_x^w[n],\ \ h_x^b[n] \qquad n = 0 \ldots N - 1, \tag{5.11}$$

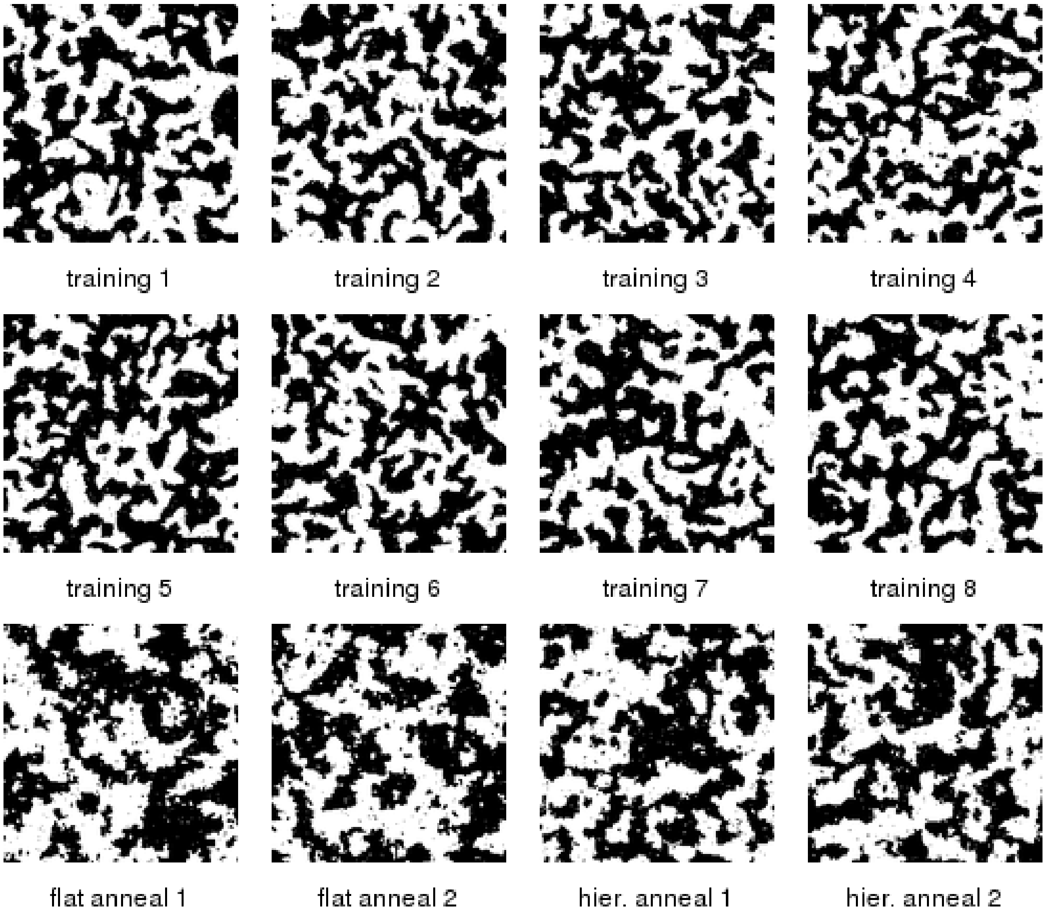
with total counts C_x^w, C_x^b respectively. Thus the sample probability of configuration k for white central pixel is $\frac{h_x^w[k]}{C_x^w}$.

Define an energy function as a (possibly weighted) sum of errors with respect to the target pmf's,

$$\mathcal{E}(x) = \sum_{n=0}^{N-1}\left[\alpha_n\left(p_w[n]-\frac{h_x^w[n]}{C_x^w}\right)^2+\beta_n\left(p_b[n]-\frac{h_x^b[n]}{C_x^b}\right)^2\right]. \tag{5.12}$$

The less general case of $\alpha_n = \beta_n = 1$ seems to be effective in practice, but for some classes of images it may improve convergence rates to weight this way.

Figure 5.21 shows some illustrative preliminary results, on an 'easy' synthetic data set. Essentially what this figure shows is that


training 1

training 2

training 3

training 4

training 5

training 6

training 7

training 8

flat anneal 1

flat anneal 2

hier. anneal 1

hier. anneal 2

Figure 5.21: Examples of flat and hierarchical annealing

the model captures roughly some morphological features of the training data, and (empirically) converges to the same sorts of images (in a loose sense) in both SA (here labelled ’flat’) and HSA cases.

This experiment providing a sort of ‘sanity check’, let’s consider the multiscale behaviour of these histograms on the same data as shown in 5.19

Figures 5.22 shows a sequence of rescalings of the positively biased case. The leftmost column shows a representative image at several scales. The next two columns show the chordlength (white phase) and local histogram (white central pixel) for each scale, (taken from the entire data set). Figure 5.23 shows similar figures, but for the negatively biased case.

Similarly, Figures 5.25 and 5.25 show the black phase chordlength and black central pixel case.

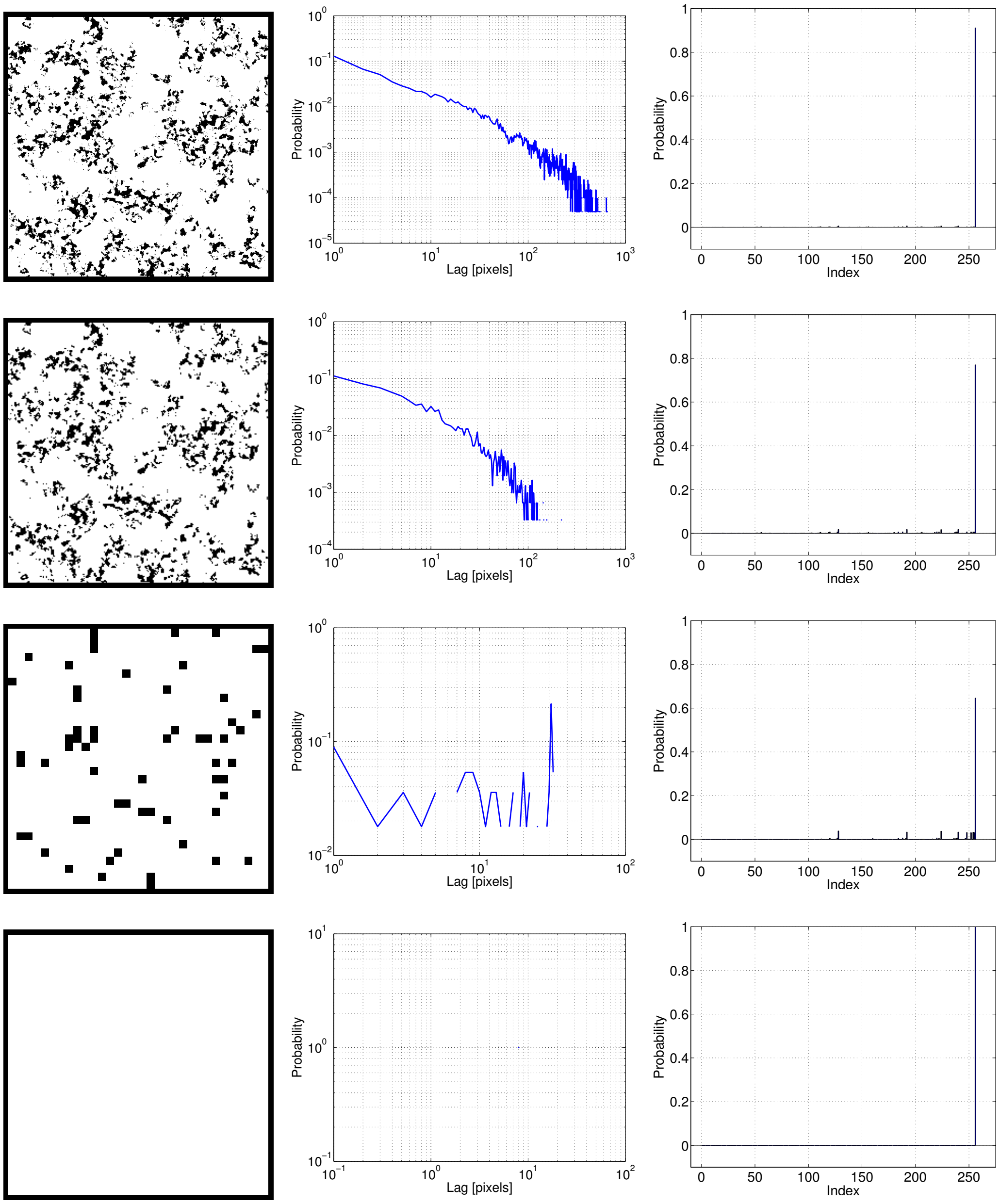


Figure 5.22: Positively biased sample with white phase chordlength distribution and local histogram.

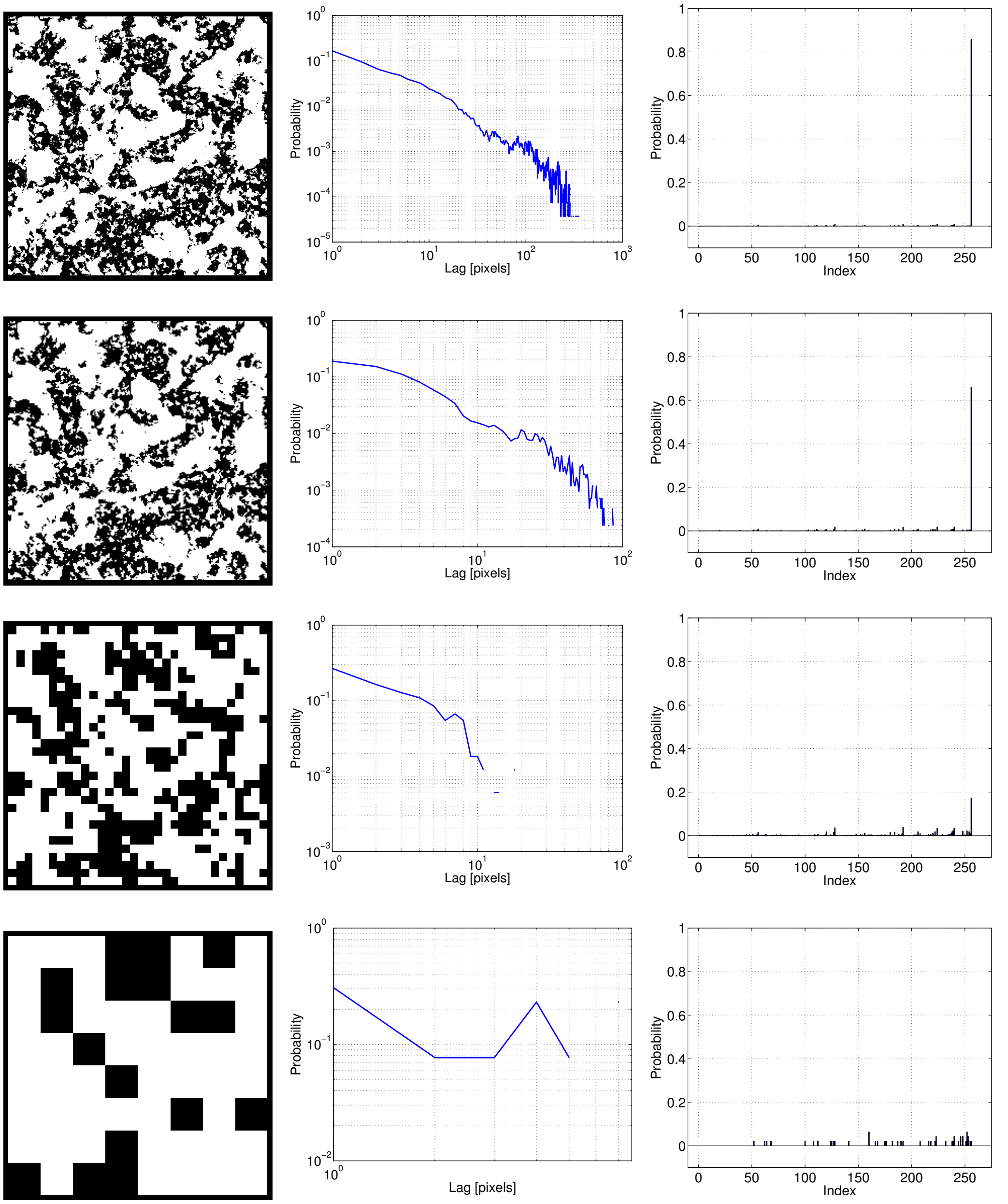


Figure 5.23: Negatively biased sample with white phase chordlength distribution and local histogram.

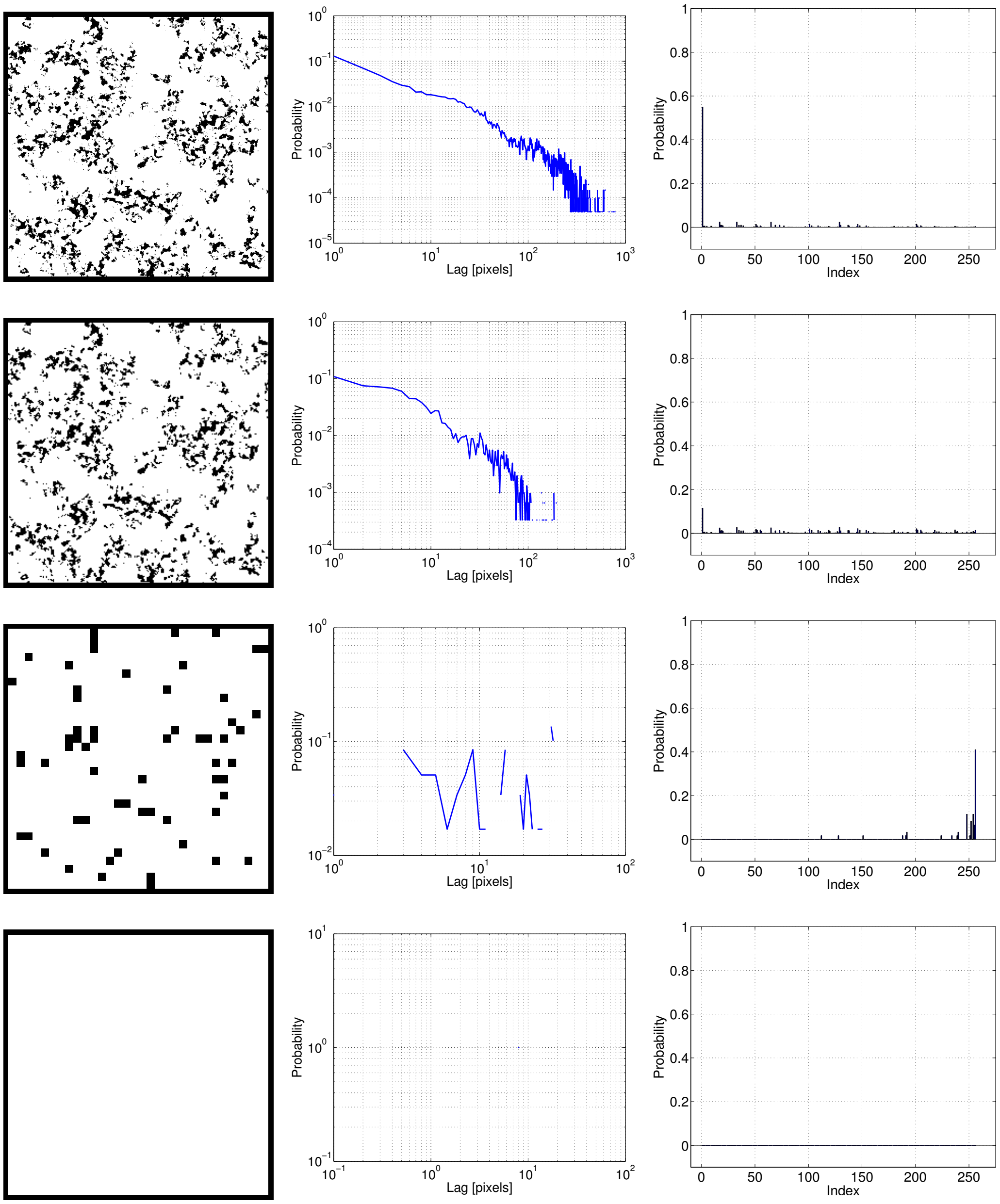


Figure 5.24: Positively biased sample with black phase chordlength distribution and local histograms.

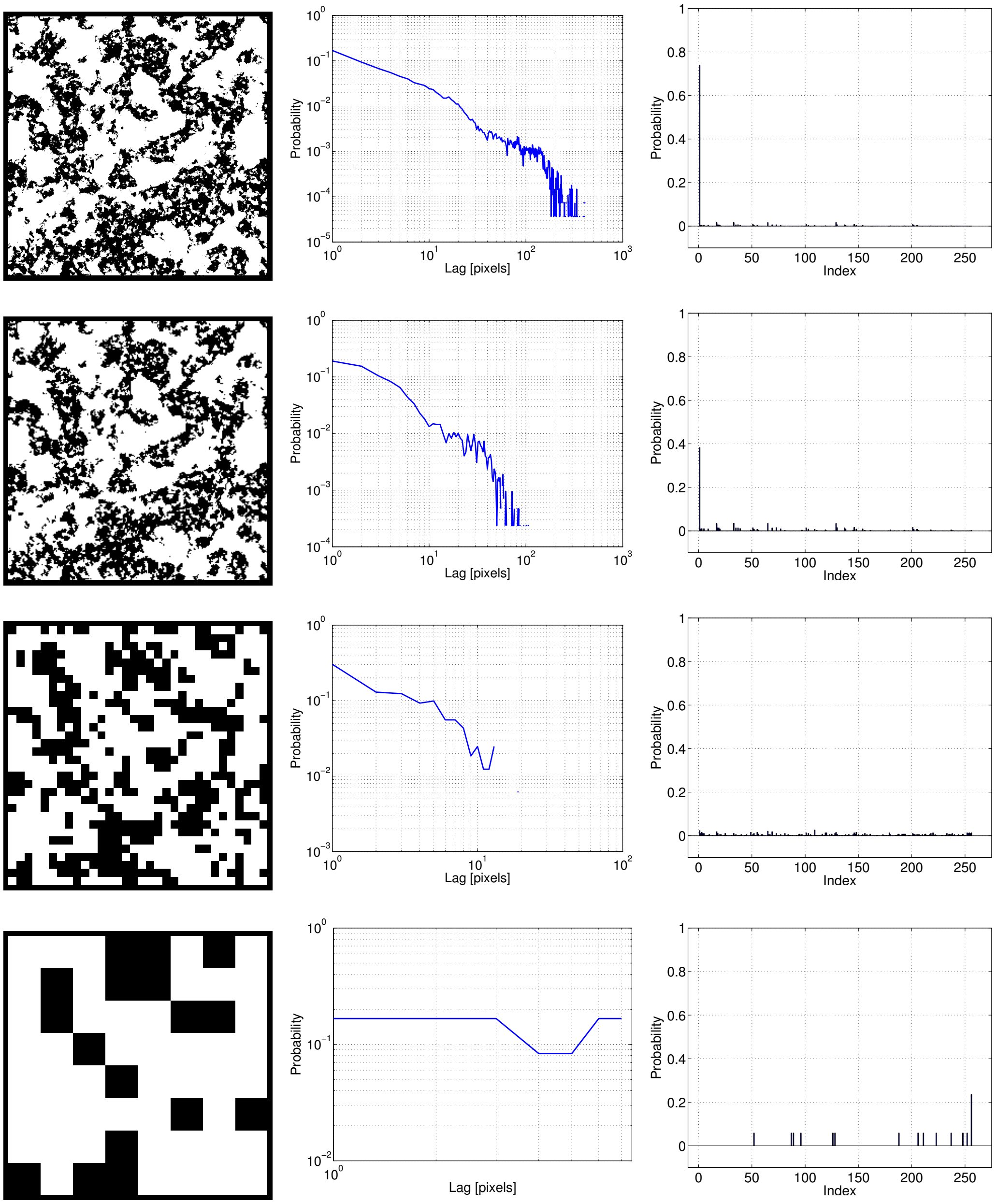
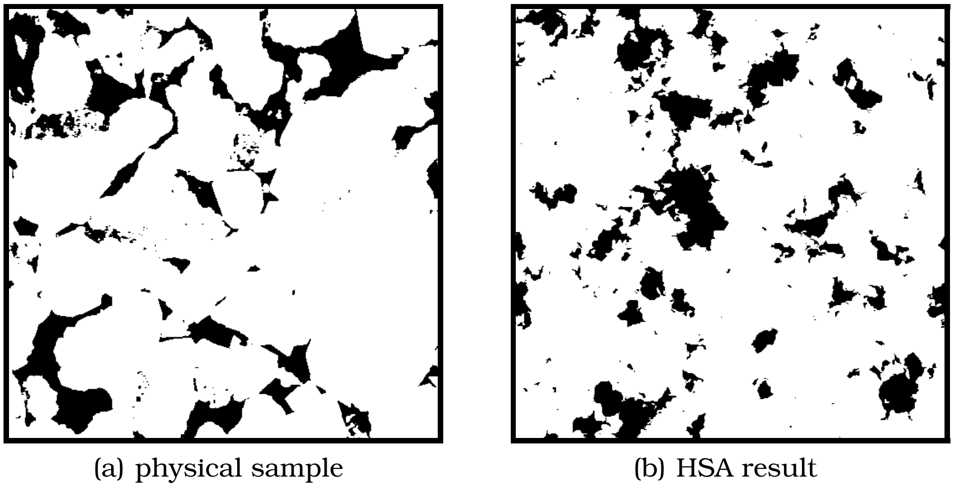


Figure 5.25: Negatively biased sample with black phase chordlength distribution and local histogram.


(a) physical sample

(b) HSA result

Figure 5.26: Training data (a) and a synthetic sample (b) of Berea Sandstone.

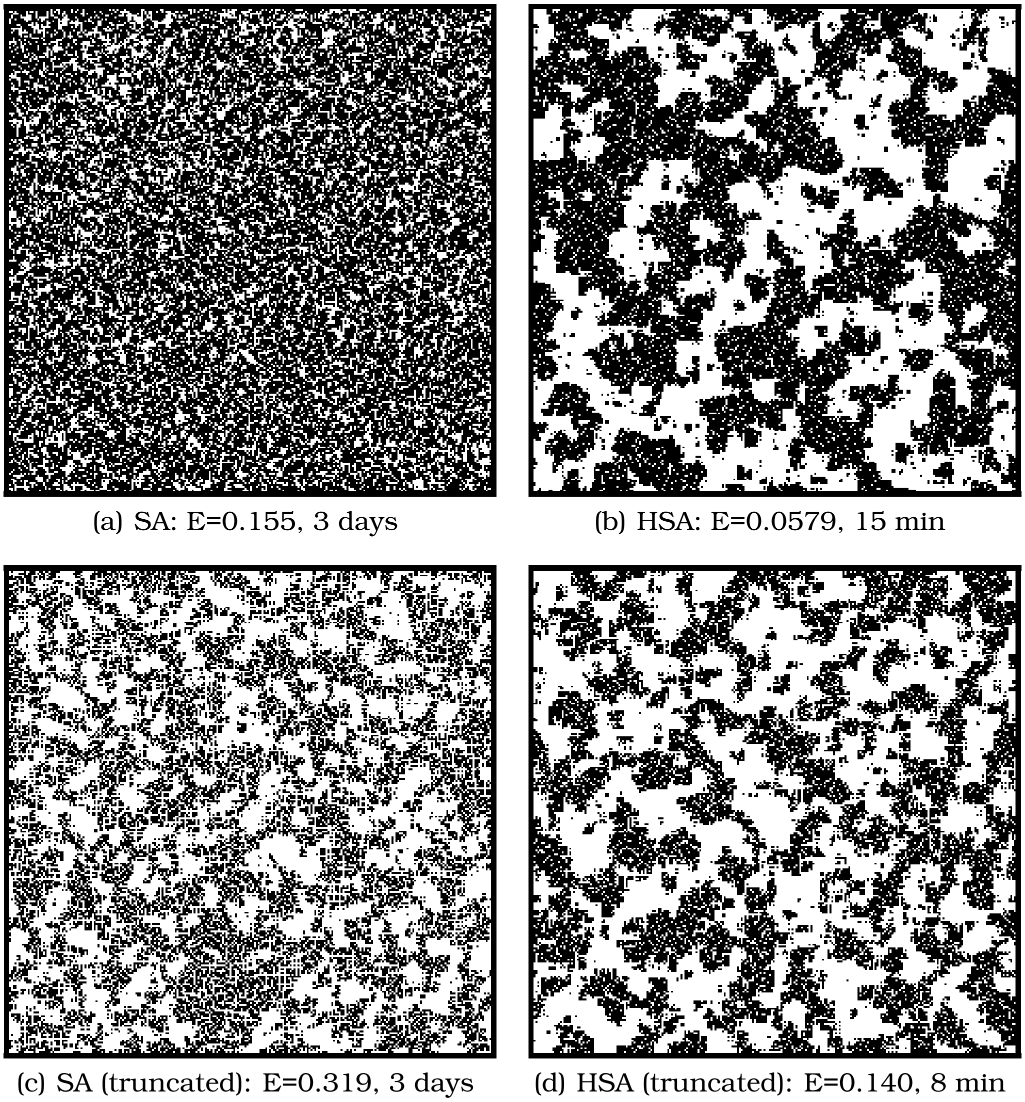
It is interesting to note from these the evolving shape of the histograms with scale. The coarse image rows show some of the difficulty of lack of data for estimating some of these distributions at coarse scales.

These results are typical of experiments that have been run. This type of empirical measurement suggests that a hierarchical approach can address the discrimination problem previously encountered.

Indeed, these suggestions seem to work quite well. Figure 5.26 shows a comparison of a physical sample of Berea sandstone, and a reconstructed result using the methods of this chapter and the local histogram based energy function.

Additionally, these figures support a fairly natural idea: if we are using something like chordlength that is taken as a function of distance, but doing this at multiple scales, there is redundant effort. Thus it should be beneficial to use a ‘truncated chordlength’. In other words, at any particular scale the energy function will only concern itself with relatively short lags,(i.e. number of pixels away from the current site).

As a final example, we have taken a high-resolution image of



(a) SA: E=0.155, 3 days

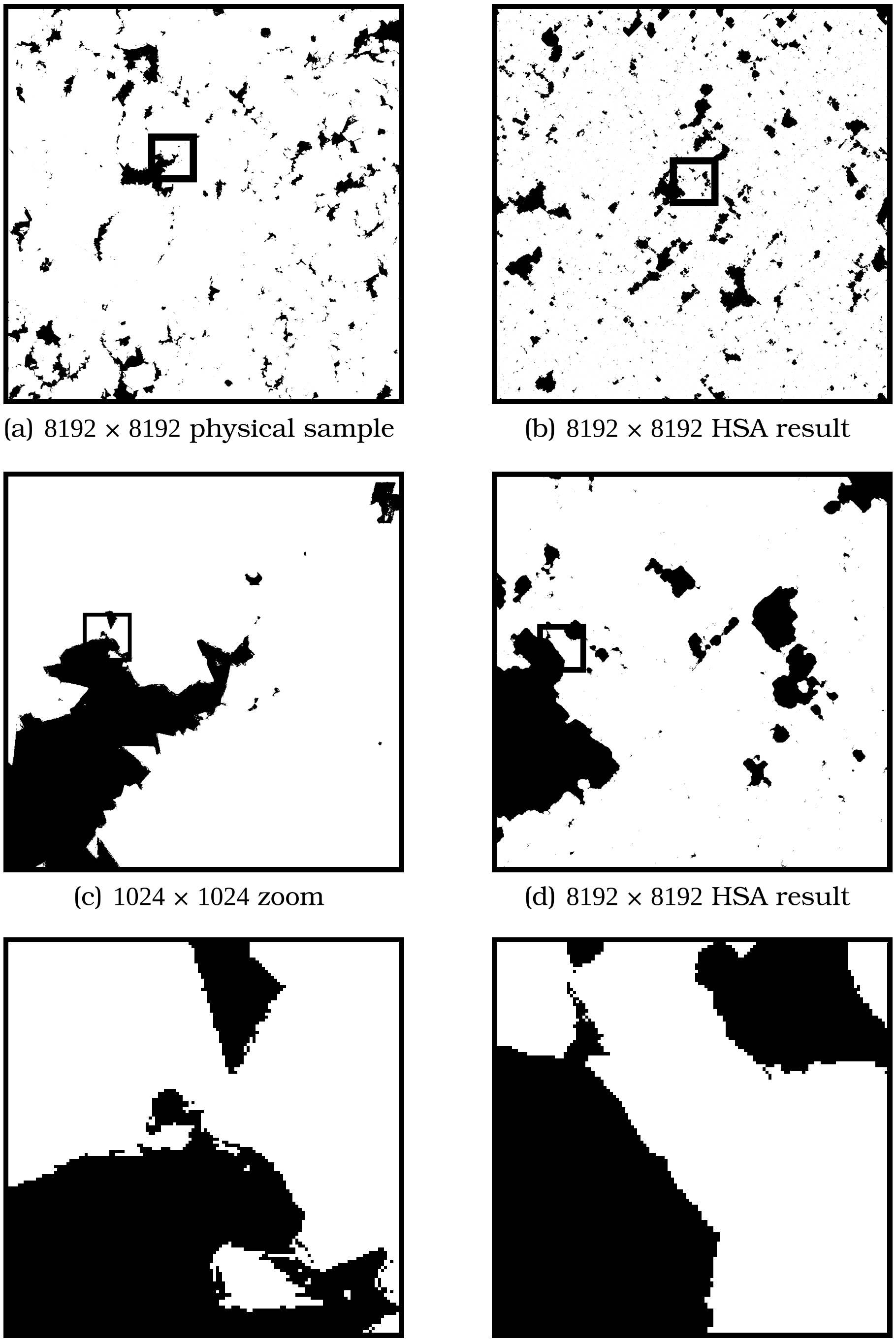
(b) HSA: E=0.0579, 15 min

(c) SA (truncated): E=0.319, 3 days

(d) HSA (truncated): E=0.140, 8 min

Figure 5.27: Results of truncating the chordlength distributions in calculating energy function. Top row shows (a) SA and (b) HSA results using complete chordlength distributions. Bottom row gives (c) SA and (d) HSA reconstruction only considering the first 10 pixels of lag. This has clearly improved the SA results (although convergence remains slow). HSA can be done *very* quickly with this energy function, as shown, but does not reach quite as low a temperature.

vuggy carbonate rock (1.87 micron/pixel) showing structure at multiple scales, and attempted to reconstruct it using the HSA technique with an energy function based on local histograms and chordlength. Figure 5.28 gives this original image and the reconstruction at several scales, while Figure 5.29 shows very good agreement in both chordlength and auto-correlation (which is not directly part of the energy function)



(a) 8192 × 8192 physical sample

(b) 8192 × 8192 HSA result

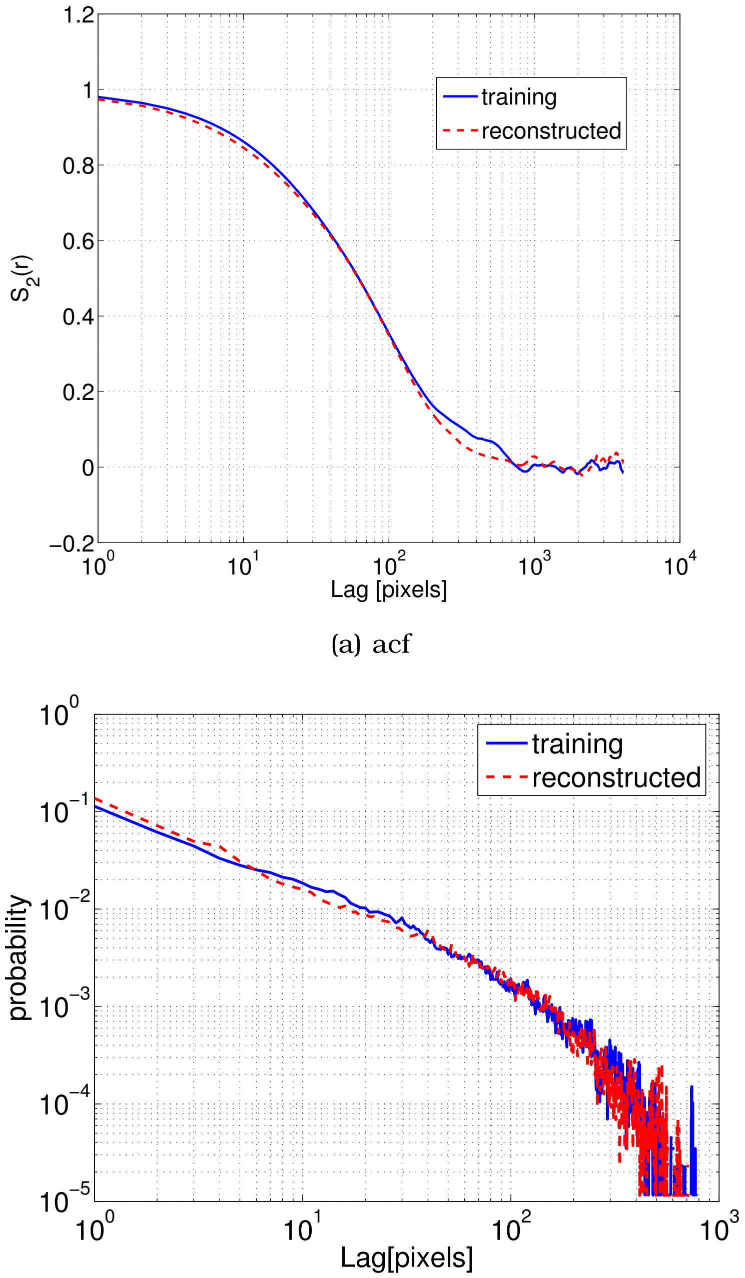
(c) 1024 × 1024 zoom

(d) 8192 × 8192 HSA result

(e) 128 × 128 zoom

(f) 8192 × 8192 HSA result

Figure 5.28: High-resolution image of vuggy carbonate rock (1.87 micron/pixel) showing structure at multiple scales and HSA sample. The reconstruction took approximately 4 days, with a final energy of E=2.85e-4 (the original image has energy of 4.44e-6)



(a) acf

(b) chordlength

Figure 5.29: Comparison of the ACFs (a) and chordlength distributions (b) of images given in Figure 5.28

5.6 Remarks

In this chapter, we have described the application area of reconstruction of porous media images, an area with many technical difficulties and broad application. In particular, the approach of ‘simulated annealing as sampling’ has been investigated. A hierarchical algorithm has been proposed, and empirically shown to improve the computational costs of such approaches. This setting also allows for novel modelling approaches. Some difficulties with modelling approaches used in the porous media literature have been investigated, and benefits of this HSA method described in addressing them. A particular, novel, ‘local histogram’ approach was presented and demonstrated to have good empirical performance. This by no means represents a complete solution of the difficulties inherent in modelling porous media. This approach now allows porous media researchers to attempt more difficult reconstruction problems, on large images with multiple natural length scales.

Work is in progress on more formal connection to both Renormalisation Group and Multi-grid approaches. In practice, even more complicated energy functions are likely to be needed for this reconstruction problem, and highly accelerated cooling. Both of these facts make analysis difficult. It is hoped that at least with the right sort of simplifying assumptions, some conclusions may be drawn.

CHAPTER 6

Conclusions

The novel contributions of this thesis are the following. In the area of IFS, or 'fractal' imaging, we are initiating an attempt to go 'beyond compression', and look at what other imaging problems might benefit from these techniques. Although there are a few examples of such work in the literature (fractal zoom, denoising), there has not been a general approach to this problem to date. Motivated by this observation, a study of the statistics of map parameters and collage error has been undertaken. Directly from these results, a novel noise variance estimation technique was discussed. Following these results, and further motivated by observing the 'clustering' of edge-containing domains, a novel multiparent IFS approach has been introduced. This approach is generalisable between IFSM and IFSW type operators. The performance of the multiparent IFS on denoising applications was demonstrated. A preliminary investigation on the identification or classification of edges in images based on these techniques was sketched.

In the area of porous media reconstruction, a novel multiscale, hierarchical annealing method (HSA) was introduced. Empirical results describing the impressive computational gains realised by this method were demonstrated. Further discussion on the difficult problem of modelling in this context was offered, along with novel modelling approaches based on the HSA which improve results.

Acronyms

IFS Iterated Function Systems.

IFSM Iterated Function Systems on Grey Level Maps.

IFSW Iterated Function Systems on Wavelet trees.

pdf probability density function.

pmf probability mass function.

cdf cumulative probability density function.

MRA Multiresolution Analysis.

DWT Discrete Wavelet Transform.

CWT Continuous Wavelet Transform.

CMP Contraction Mapping Principle.

PSNR Peak Signal to Noise Ratio.

MSE Mean Squared Error.

RMSE Root Mean Squared Error.

HSA Hierarchical Simulated Annealing.

SA Simulated Annealing.

GMRF Gaussian Markov Random Field.

MRF Markov Random Field.

MCMC Markov Chain Monte Carlo.

PSNR Peak Signal to Noise Ratio.

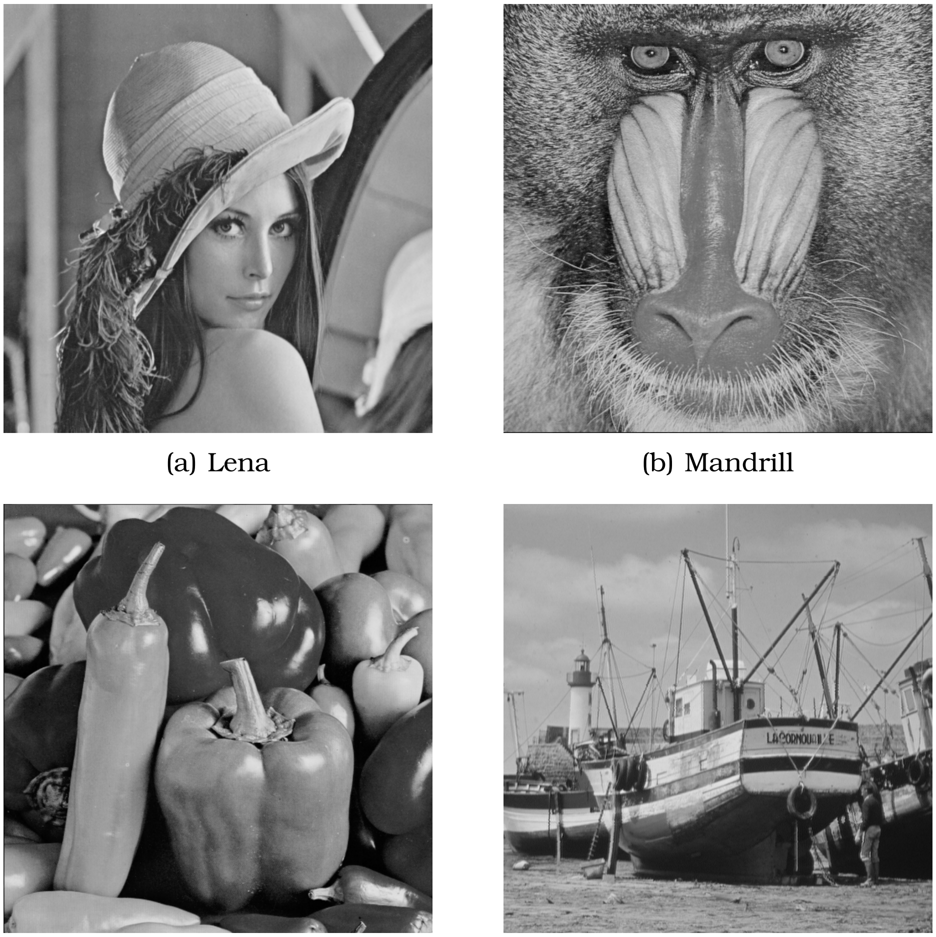
pixel picture element.

HVS Human Visual System.

PDE partial differential equation

iid independent, identically distributed

Test Images



(a) Lena

(b) Mandrill

(c) Peppers

(d) Boat

Figure 6.1: Example ‘standard’ test images

BIBLIOGRAPHY

[AFV03a] S.K. Alexander, P. Fieguth, and E.R. Vrscay. Hierarchical annealing for random image synthesis. In *EMMCVPR'03*, number 2683 in LNCS. Springer, 2003. 169

[AFV03b] S.K. Alexander, P. Fieguth, and E.R. Vrscay. Image sampling by hierarchical annealing. In *ICIP'03*. IEEE, 2003. 169

[AFV04a] S.K. Alexander, P. Fieguth, and E.R. Vrscay. Hierarchical annealing for scientific models. In *ICASSP'04*. IEEE, 2004. 149

[AFV04b] S.K. Alexander, P. Fieguth, and E.R. Vrscay. Parameterized hierarchical annealing for scientific models. In *International Conference on Image Analysis and Recognition (ICIAR'04)*, volume 2683 of *LNCS*, pages 236–243. Springer, 2004. 157

[AJQ90] P.M. Adler, C.G. Jacquin, and J.A. Quiblier. Flow in simulated porous media. *International Journal of Multiphase Flow*, 16(4):691–712, July–August 1990. 136

[AKR04] C.H. Arns, M.A. Knackstedt, and Mecke K. R. Characterisation of irregular spatial structures by parallel sets and integral geometric measures. *Colloids And Surfaces A-Physicochemical And Engineering Aspects*, 24:351–372, July 2004. 152

[Ban22] S. Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations integrales. *Fundamenta Mathematicae*, 3:133–181, 1922. 20

[Bar65] A.A. Barker. Monte carlo calculations of the radial distribution function for a proton-electron plasma. *Australian Journal of Physics*, 18:119–113, 1965. 121, 123

[Bar93] M.F. Barnsley. *Fractals Everywhere.* Academic Press Professional, Cambridge, MA, second edition, 1993. 15, 22, 23, 24, 25, 26, 27, 28, 30

[BD85] M.F. Barnsley and S. Demko. Iterated function systems and the global construction of fractals. *Proc. Roy. Soc. London*, A399:243–275, 1985. 33

[BDK92] T. Bedford, F.M. Dekking, and M.S. Keane. Fractal image coding techniques and contraction operators. *Nieuw Arch. Wiskunde*, 10:185–217, 1992. 38

[BEHL86] M.F. Barnsley, V. Ervin, D. Hardin, and J. Lancaster. Solution of an inverse problem for fractals and other sets. *Proc. Nat. Acad. Sci. USA*, 83:1975–1977, 1986. 31, 33

[Ber87] N.F. Berk. Scattering properties of a model bicontinuous structure with a well defined length scale. *Physical Review Letters*, 58(25):2718–2721, June 1987. 136

[Bes86] J. Besag. On the statistical analysis of dirty pictures. *J. of the Royal Statistical Soc.*, B(38):259–302, 1986. 130

[BGHM95] J. Besag, P. Green, D. Higdon, and K. Mengersen. Bayesian computation and stochastic systems (with discussion). *Statistical Science*, 10:3–66, 1995. 125

[BH93] M.F. Barnsley and L.P. Hurd. *Fractal Image Compression*. AK Peters, Wellesley, MA, USA, 1993. 69

[Bie85] I. Biederman. Human image understanding: recent research and a theory. *Computer Vision, Graphics and Image Processing*, 32:29–73, 1985. 116

[Bré98] P. Brémaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer, New York, 1998. 2, 5, 117, 120, 121, 125, 126, 130, 133, 147

[Bri95] C.M. Brislawn. Preservation of subband symmetry in multirate signal coding. *IEEE Trans. Signal Processing*, 43(12):3046–3050, December 1995. 54

[Bri96] C.M. Brislawn. Classification of nonexpansive symmetric extension transforms for multirate filter banks. *Appl. Comput. Harmonic Analysis.*, 3:337–357, 1996. 54

[BS94] C. Bouman and M. Shapiro. A multiscale random field model for bayesian image segmentation. *IP*, 3(2):162–177, March 1994. 5, 10, 115

[Cam68] F.W. Campbell. The human eye as an optical filter. In *Proceedings of the IEEE*, volume 56, pages 1009–1014, June 1968. 8

[Can86] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679 – 698, 1986. 108, 109

[Cat92] O. Catoni. Rough large deviations estimates for simulated annealing. application to exponential schedules. *Ann. Probability*, 1992. 128

[CDV93] A. Cohen, I. Daubechies, and P. Vial. Wavelet bases on the interval and fast algorithms. *J. of Appl. and Comput. Harmonic Analysis*, 1:54–81, 1993. 54, 55

[Čer85] V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 1985. 127

[CHK83] R. Chellappa, Y.-H. Hu, and S.-Y. Kung. On two-dimensional markov spectral estimation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-31(4):836–841, August 1983. 131, 132

[CJ83] G.R. Cross and A.K. Jain. Markov random field texture models. *IEEE transactions PAMI*, PAMI(5):25–39, 1983. 5, 114

[CJ91] R. Chellappa and A. Jain, editors. *Markov Random Fields: Theory and Application*. Academic Press, Boston, MA, 1991. 114, 131, 133

[CK82] R. Chellappa and R. L. Kashyap. Digital image restoration using spatial interaction models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-30(3):461–472, June 1982. 131, 132

[CV94] P. Centore and E.R. Vrscay. Continuity of attractors and invariant measures for iterated function systems. *Canadia Mathematical Bulletin*, 37(3):315–329, 1994. 19

[Dau92] I. Daubechies. *Ten Lectures on Wavelets*. SIAM Press, Philadelphia, PA, 1992. 39, 40, 41, 42, 43, 44, 45, 48, 49, 54, 99

[Dav98] G. Davos. A wavelet-based analysis fractal image compresion. *IEEE Tran. Image Proc.*, 7:141–154, 1998. 58, 69

[DF92] A. Cohen I. Daubechies and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45:485–560, 1992. 54

[DJ94] D.L. Donoho and I.M. Johnstone. Ideal spatial adaption by wavelet shrinkage. *Biometricka*, pages 425–455, 1994. 5

[DK89] H. Derin and P.A. Kelly. Discrete-index markov-type random processes. In *Proceeding of the IEEE*, volume 77, pages 1485–1510, October 1989. 114, 131, 133

[EHF$^+$05] N. Eschricht, E. Hoinkis, Mädler F., P. Schubert-Bischoff, and B. Röhl-Kuhn. Knowledge-based reconstruction of random porous media. *Journal of Colloid and Interface Science*, 291:201–213, 2005. 136, 138

[Fal90] K. Falconer. *Fractal Geometry: Mathematical Foundations and Applications*. John Wiley and Sons, West Sussex, England, 1990. 15, 25, 26

[Fal97] K. Falconer. *Techniques in Fractal Geometry*. John Wiley and Sons, West Sussex, England, 1997. 15

[Fie93] D.J. Field. Scale-invariance and self-similar 'wavelet' transforms: an analysis of natural scenes and mammalian visual systems. In M. Farge, J. C. R. Hunt, and J. C. Vassilcos, editors, *Wavelets, Fractals, and Fourier Transforms*. Oxford University Press, 1993. 10

[Fis95] Y. Fischer, editor. *Fractal Image Compression: Theory and Application*. Springer-Verlag, New York, 1995. 69

[Fis98] Y. Fischer, editor. *Fractal Image Encoding and Analysis*, NATO ASI Series F 159, New York, 1998. Springer-Verlag. 196

[FK98] O. Faugeras and R. Keriven. Variational principles, surface evolution, pde's, level set methods and the stereo problem. *IEEE Trans. Image Processing*, 7(3):336–344, 1998. 2

[Fro99] J. Froment. A compact and multiscale image model based on level sets. In *Scale-Space Theories in Computer Vision*, pages 152–163, 1999. 10

[FV98a] B. Forte and E. R. Vrscay. Inverse problem methods for generalized fractal transforms. In Fischer [Fis98]. 33, 58

[FV98b] B. Forte and E. R. Vrscay. Theory of generalized fractal transforms. In Fischer [Fis98]. 33, 55, 69

[Gem88] D. Geman. *Lecture Notes in Mathematics*, chapter Random Fields and Inverse Problems in Imaging, pages 117–186. Springer-Verlag, 1988. 133

[GFV03] M. Ghazel, G. Freeman, and E.R. Vrscay. Fractal image denoising. *IEEE Trans. Image Proc.*, 12:1560–1578, 2003. 69, 96

[GG84] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984. 5, 114, 121, 123, 125, 126, 129, 130, 133, 147

[Gid89] B. Gidas. A renormalization group approach to image processing problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):164–180, 1989. 137

[Gra87] C. Graffigne. *Experiments in texture analysis and segmentation*. PhD thesis, Brown University, 1987. 114

[GW02] R. C. Gonzales and R. E. Woods. *Digital Image Processing*. Addison Wesley, Boston, MA, 2 edition, 2002. 79

[Haa10] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Math. Annal.*, 69:331–371, 1910. 40

[Haj88] B. Hajek. Cooling schedules for optimal annealing. *Math. Oper. Res.*, 13(2):311–329, 1988. 129, 130

[Har98] H. Hartenstiein. *Topics in Fractal Image Compression and Near-Lossless Image Coding*. PhD thesis, Albert-Ludwigs-Universität Freiburg im Breisgau, 1998. 68, 69

[Has70] W.K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–108, 1970. 121, 123, 125, 126

[HC71] J. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. Unpublished manuscript, 1971. 120

[HPB98] T. Hofmann, J. Puzicha, and J.M. Buhmann. Unsupervised texture segmentation in a deterministic annealing framework. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(8):803–818, Aug 1998. 114

[HRSV86] M. Huang, F. Romeo, and A. Sangiovani-Vincentelli. An efficient general cooling schedule for simulated annealing. *Proc. Int. Conf. Computer-Aided Design*, pages 381–384, 1986. 130

[Jac92] A. Jacquin. Image coding based on a fractal theory of iterated contractive image transformations. *EEE Trans. Image Proc*, 1:18–30, 1992. 69

[JM98] M.A. Viergever J.B.A. Maintz, E.H.W. Meijering. General multimodal elastic registration based on mutual information. In K. M. Hanson, editor, *Proceedings of SPIE Medical Imaging 1998: Image Processing*, volume 338, pages 144–154, Bellingham, WA, 1998. The International Society for Optical Engineering. 2

[KBZ93] Z. Kato, M. Berthod, and J. Zerubia. Multiscale markov random field models for parallel image classification. In *Proceedings of the Fourth International Conference on Computer Vision*, Berlin, Germany, 1993. 114

[KGV83] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983. 127

[KJ97] E. Kira and C. Ji. Rates of convergence for the Gibbs sampler. *Markov Processs and Related Fields*, 3:89–102, 1997. 125

[Kos94] S. M. Kosslyn. *Image and Brain*. MIT Press, Cambridge, MA, 1994. 5

[KS80] R. Kindermann and J.L. Snell. *Markov random fields and their applications*, volume 1 of *Contemporary Mathematics*. American Mathematical Society, Providence, R.I., 1980. 121, 123

[KZB99] Z. Kato, J. Zerubia, and M. Berthod. Unsupervised parallel image classification using markovian models. *Pattern Recognition*, 32(4):591–604, 1999. 114

[Lev98] P. Levitz. Off-lattice reconstruction of porous media: critical evaluation, geometrical confinement and molecular transport. *Advances in Colloid and Interface Science*, 77:71–106, July 1998. 136, 137

[LIC00a] Z. Liang, M.A. Ioannidis, and I. Chatzis. Permeability and electrical conductivity of porous media from 3d stochastic replicas of the microstructure. *Chemical Engineering Science*, 55(22):5247–5262, November 2000. 164

[LIC00b] Z. Liang, M.A. Ioannidis, and I Chatzis. Reconstruction of 3d porous media using simulated annealing. In Bentley et al., editor, *Computational Methods in Water Resources XIII*, Balkema, Rotterdam, 2000. 140, 148

[Mal89a] S. Mallat. Multiresolution approximation and wavelet orthonormal bases of $\mathcal{L}^2$. *Transactions of the American Mathematics Society*, 315:69–87, 1989. 43

[Mal89b] S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 11(7):674, 1989. 43

[Mal98] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, 1998. 39, 40, 41, 45, 48, 49, 54, 99

[Man77] B. Mandelbrot. *The Fractal Geometry Of Nature*. Freeman, New York, 1977. 31

[Mey01] Y. Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, volume 22 of *University Lecture Series*. American Mathematical Society, Boston, 2001. 5

[MRR$^+$53] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953. 121, 123

[MS89] D.B. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications in Pure and Applied Mathematics*, 42:577–684, 1989. 5, 69

[MS95] J.-M. Morel and S. Solimini. *Variational Methods in Image Segmentation*, volume 15 of *Progress in Nonlinear Differential Equations and their Applications*. Birkhäuser, Boston, 1995. 2

[Mum95] D. Mumford. The statistical description of visual signals. In Klasu Kirchgässner, Oscar Mahrenholtz, and Reinhard Mennicken, editors, *Proceedings of the Third International Congress on Industrial and Applied Mathematics*, pages 234–255. Academie Verlag, July 1995. 133

[MV97] F. Mendivil and E.R. Vrscay. Correspondence between fractal-wavelet transforms and iterated function systems with grey-level maps. In E. Lutton J. Levey-Vehel and C. Tricot, editors, *Fractals in Engineering: From Theory to Industrial Applications*, pages 54–64, London, 1997. Springer-Verlag. 55, 58, 59

[Nil85] N. Nill. A visual model weighted cosine transform for image compression and quality assessment. *IEEE Trans, Comm.*, pages 551–557, 1985. 8

[OI04] S. Olayinka and M.A. Ioannidis. Time-dependent diffusion and surface-enhanced relaxation in stochastic replicas of porous rock. *Transport in Porous Media*, 54(3):273–295, March 2004. 164

[OS88] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988. 2

[PB99] J. Puzicha and J.M. Buhmann. Multiscale annealing for grouping and unsupervised texture segmentation. *Computer Vision and Image Understanding: CVIU*, 76(3):213–230, 1999. 115

[PD97] E. Polidori and J.-L. Dugelay. Zooming using iterated function systems. *Fractals*, 5 (Supplementary Issue):111–123, 1997. 69

[PL98] R. Paget and I.D. Longstaff. Texture synthesis via a non-causal nonparametric multiscale markov random field. *IEEE Transactions on Image Processing*, 7(6):925–931, 1998. 115

[Pra78] W.K. Pratt. *Digital Image Processing*. Wiley-Interscience, New York, 1978. 8

[RCB99] J.K. Romberg, H. Choi, and R.G. Baraniuk. Shift-invariant denoising using wavelet-domain hidden markov trees. In *33rd Asilomar Conference*, October 1999. 99, 102

[RIH+04] A.P. Radlinski, M.A. Ioannidis, A.L. Hinde, M. Hainbuchner, H. Rauch M. Baron, and S.R. Kline. Angstrom-to-millimeter characterization of sedimentary rock microstructure. *Journal of Colloid and Interface Science*, 274(2):607–612, June 2004. 164

[RK96] A.P. Roberts and M.A. Knackstedt. Structure-property correlations in model composite materials. *Physical Review E*, 54(3):2318–2328, September 1996. 136

[RO94] L. Rudin. and S. Osher. Total variation based image restoration with free local constraints. In *Proceedings of the 1st IEEE ICIP*, volume 1, pages 31–35, 1994. 5

[Rob97] A.P. Roberts. Statistical reconstruction of three-dimensional porous media from two-dimensional images. *Physical Review E*, 56(3):3203–3212, September 1997. 137

[ROF92] L. Rudin., S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D.*, 60:259–268, 1992. 5

[RU01] M.G. Rozman and Marcel Utz. Efficient reconstruction of multiphase morphologies from correlation functions. *Physical Review E*, 63(066701), 2001. 137, 138, 164

[SFAH92] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D.J. Heeger. Shiftable multi-scale transforms. *IEEE transactions on informations theory*, 38(2), 1992. 10

[SH87] H. Szu and R. Hartley. Fast simulated annealing. *Physics Letters A*, 122:157–162, 1987. 147

[SH94] D. Saupe and R. Hamzaoui. Complexity reduction methods for fractal image compression. In J. M. Blackledge, editor, *Proceedings of the IMA Conference on Image Processing: Mathematical Methods and Applications*, pages 211–229, Oxford, England, 1994. 69

[Sha48] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948. 2

[Sha93] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 41(12):3445–3462, 1993. 69

[SP96] A. Said and W.A. Pearlman. A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6:243–250, 1996. 69

[Top98] P.N. Topiwala, editor. *Wavelet Image and Video Compression*. Kluwer Academic Publishers, Boston, MA, 1998. 54

[Tor01] S. Torquato. *Random Heterogeneous Materials: Microstructure and Macroscopic Properties*. Springer-Verlag, New York, 2001. 140, 148, 149

[TTI02] M.S. Talukdar, O. Torsaeter, and M.A. Ioannidis. Stochastic reconstruction of particulate media from two-dimensional images. *Journal of Colloid and Interface Science*, 248(2):419–428, April 2002. 137, 138, 140, 148, 149, 164

[Vid99] B. Vidakovic. *Statistical Modeling by Wavelets*. Wiley Inter-Science, New York, 1999. 2

[Vio95] P. A. Viola. Alignment by maximization of mutual information. Technical Report AITR-1548, 1995. 2

[(VQ00] Video Quality Experts Group (VQEG). Final report from the video quality experts group on the validation of objective models of video quality assessment. Technical report, VQEG, 2000. 7

[Vrs96] E. R. Vrscay. Mathematical theory of generalized fractal transforms and associated inverse problems. In *Proceedings of ImageTech 96 Conference on Multimedia Imaging Technology and Applications*, Atlanta, GA, 1996. 33, 69

[Vrs98] E.R. Vrscay. A generalized class of fractal-wavelet transforms for images representation and compression. *Can. J. Elect. and Comp. Eng.*, 23(1-2):69–83, 1998. 55, 58, 59, 60, 62, 63, 69, 75

[VS99] E.R. Vrscay and D. Saupe. Can one break the 'collage barrier' in fractal image coding. In M. Dekking, J. Levy-Vehel, E. Lutton, and C. Tricot, editors, *Fractals: Theory and Applications in Engineering*, pages 307–323, London, 1999. Springer Verlag. 32, 75

[WA85] A. Watson and A.J. Ahumada. Model of human visual-motion sensing. *Journal of the Optical Society of America*, 2(2):322–341, 1985. 10

[Wan95] B. A. Wandell. *Foundations of Vision*. Sinauer Associates, Sunderland Mass, 1995. 5, 10, 116

[Win91a] C.P. Winder. *Markovian Analysis of Texture: Serial and Parallel Paradigms in Low-Level Vision*. PhD thesis, University of Oxford, 1991. 131, 133

[Win91b] G. Winkler. *Image analysis, random fields, and dynamic monte carlo methods*. Springer, Berlin, 1991. 2, 5, 120, 121, 123, 125, 130, 133

[YT98] C.L.Y. Yeong and S. Torquato. Reconstructing random media. *Physical Review E*, 57:495–506, 1998. 137, 138, 140, 149, 164

[ZFW00] J. Zhang, P. Fieguth, and D. Wang. *Handbook of Image Processing*, chapter 4.3, pages 293–304. Academic Press, 2000. 131, 132, 133

[ZKB94] J. Zerubia, Z. Kato, and M. Berthod. Multi-temperature annealing: a new approach for the energy-minimization of hierarchical markov random field models. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, volume 1, Jerusalem, Israel, 1994. 115