This document is formatted for double-sided printing on A4 paper. Page numbering may differ slightly from that of the official version in the university library. Compilation date of this version: 21 May 2003

Fractal Image Compression and the Self-Affinity Assumption: A Stochastic Signal Modelling Perspective

Brendt Wohlberg
Department of Electrical Engineering
University of Cape Town
brendt@dip.ee.uct.ac.za

Submitted to the University of Cape Town in fulfilment of the requirements for the degree of Doctor of Philosophy.

Cape Town, August 1996.

Abstract

Fractal image compression is a comparatively new technique which has gained considerable attention in the popular technical press, and more recently in the research literature. The most significant advantages claimed are high reconstruction quality at low coding rates, rapid decoding, and "resolution independence" in the sense that an encoded image may be decoded at a higher resolution than the original. While many of the claims published in the popular technical press are clearly extravagant, it appears from the rapidly growing body of published research that fractal image compression is capable of performance comparable with that of other techniques enjoying the benefit of a considerably more robust theoretical foundation.

So called because of the similarities between the form of image representation and a mechanism widely used in generating deterministic fractal images, fractal compression represents an image by the parameters of a set of affine transforms on image blocks under which the image is approximately invariant. Although the conditions imposed on these transforms may be shown to be sufficient to guarantee that an approximation of the original image can be reconstructed, there is no obvious theoretical reason to expect this to represent an efficient representation for image coding purposes. The usual analogy with vector quantisation, in which each image is considered to be represented in terms of code vectors extracted from the image itself is instructive, but transforms the fundamental problem into one of understanding why this construction results in an efficient codebook.

The signal property required for such a codebook to be effective, termed "self-affinity", is poorly understood. A stochastic signal model based examination of this property is the primary contribution of this dissertation. The most significant findings (subject to some important restrictions) are that "self-affinity" is not a natural consequence of common statistical assumptions but requires particular conditions which are inadequately characterised by second order statistics, and that "natural" images are only marginally "self-affine", to the extent that fractal image compression is effective, but not more so than comparable standard vector quantisation techniques.

Acknowledgements

I wish to thank the following people for their contributions to this thesis:

- Professor Gerhard de Jager for being my supervisor.
- Alison Sawyer for editorial assistance and encouragement.
- Fred Nicolls for editorial assistance, and Greg Cox for valuable discussion.
- The members of the Digital Image Processing group for creating a pleasant working environment.
- My parents for their support and encouragement.

Nomenclature

Abbreviations

Abbreviation	Definition
AC	Alternating Current
AR	Autoregressive
ASCII	American Standard Code for Information Interchange
cdf	Cumulative density function
DC	Direct Current
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DMC	Discrete Memoryless Channel
DMS	Discrete Memoryless Source
DPCM	Differential Pulse Code Modulation
EZW	Embedded Zerotree Wavelet
fBm	Fractional Brownian motion
GLA	Generalised Lloyd Algorithm
HV	Horizontal-Vertical
HVS	Human Visual System
iff	If and only if
IFS	Iterated Function System
JPEG	Joint Photographic Experts Group
KLT	Karhunen-Loève Transform
LVQ	Lattice Vector Quantisation
MRA	Multiresolution Analysis
MSE	Mean Square Error
PCM	Pulse Code Modulation
pdf	Probability density function
PIFS	Partitioned Iterated Function System
pmf	Probability mass function
psd	Power spectral density
PSNR	Peak Signal to Noise Ratio
RMSE	Root Mean Square Error
SGVQ	Shape Gain Vector Quantisation
SNR	Signal to Noise Ratio
TSVQ	Tree Structured Vector Quantisation
VQ	Vector Quantisation

Symbols

Symbol	Definition
A	Linear part of affine transform
A^{-1}	Inverse of matrix A
A^T	Transpose of matrix A
A_{ij}	Submatrix of matrix A
A^*	Set of finite length strings over alphabet A
b	Range block width
$\mathbf{b}$	Offset component of affine transform
$c_{i,j}$	Multiresolution approximation coefficient
$\mathbb{C}$	Complex numbers
$C_{\mathbf{X}}$	Autocovariance matrix of random variable $\mathbf{X}$
$C_X(k,l)$	Covariance of random variables X_k and X_l
d	Domain block position
$\mathbf{d}$	Domain vector
D	Distortion
$d_{i,j}$	Multiresolution detail coefficient
$D(R)$	Distortion rate function
$d(\cdot,\cdot)$	Distance or distortion measure
$\mathbf{e}_C$	Collage error vector
$E[\cdot]$	Expectation operator
$f_X(x)$	Probability density function of random variable X
$F_X(x)$	Cumulative distribution function of random variable X
$h(S)$	Differential entropy of source S
$H(S)$	Entropy of source S
i_d	Domain subtree root resolution
i_r	Range subtree root resolution
I_b	Identity matrix ($b \times b$)
$I(A;B)$	Mutual information of random variables A and B
$L^2(\mathbb{R})$	Space of square-integrable functions
n	Signal length or vector dimensionality
$\mathbb{N}$	Natural numbers
N_d	Number of domains
N_r	Number of ranges
o	Affine offset coefficient
o^*	Optimal affine offset coefficient
o_i	Offset coefficient for range i
p_i	Domain position for range i
$p_X(x)$	Probability mass function of random variable X
$p_{X\|Y}(x\|y)$	Conditional pmf of random variable X given Y
P_i	Projection operator into V_i
$\mathbb{Q}$	Rational numbers
Q_i	Projection operator into W_i
$Q(\cdot)$	Quantisation function
$\mathbf{r}$	Range vector

Symbol	Definition
R	Bit rate
$\mathbb{R}$	Real numbers
$R_{\mathbf{X}}$	Autocorrelation matrix of random variable $\mathbf{X}$
$R_X(k)$	Correlation of random variables X_j and X_{j+k}
$r(A)$	Spectral radius of matrix A
$R(D)$	Rate distortion function
s	Affine scaling coefficient
s^*	Optimal affine scaling coefficient
s_i	Scaling coefficient for range i
$s_{\max}$	Upper bound on scaling coefficient magnitude
$\operatorname{sign} x$	Sign of real value x
$\operatorname{span} A$	Linear span of set A
$\mathcal{S}$	Shape codebook performance measure
T	Transform on block or signal
V_i	Multiresolution approximation space
W_i	Multiresolution detail space
$\mathbf{x}_T$	Fixed point of transform T
$\mathbb{Z}$	Integers
$\alpha(\mathbf{c})$	Centroid efficiency function for a Voronoi region
$\delta_{i,j}$	Kronecker delta
Δ_d	Domain increment
Λ	Null string
μ_X	Mean of random variable X
ρ	Correlation
ρ_{XY}	Correlation coefficient of random variables X and Y
$\rho_X(k)$	Variance normalised correlation of random variables X_j and X_{j+k}
ϱ	Deterministic correlation
$\varrho(\mathbf{d}, \mathbf{r})$	Deterministic correlation between vectors $\mathbf{d}$ and $\mathbf{r}$
σ_X	Standard deviation of random variable X
σ_{XY}	Covariance of random variables X and Y
$\varnothing$	Empty set
$\mathbf{0}$	Zero vector
$\mathbf{0}_{bb}$	Matrix $(b \times b)$ of zero elements
$\mathbf{1}$	Vector of unit elements
$\mathbf{1}_{bb}$	Matrix $(b \times b)$ of unit elements
$\lvert x \rvert$	Absolute value of real value x
$\lceil x \rceil$	Ceiling of real value x
$\overline{X}$	Closure of set X
$\lvert A \rvert$	Determinant of matrix A
$n!$	n factorial (for integer n)
$\lfloor x \rfloor$	Floor of real value x
$\nabla_{\mathbf{c}}$	Gradient operator (with respect to vector $\mathbf{c}$)
$\langle \cdot, \cdot \rangle$	Inner product
$A \otimes B$	Kronecker product of matrices A and B
$m \bmod n$	m modulus n (for integer m and n)
$\lVert \cdot \rVert$	Norm

Contents

Chapter 1

Introduction

The recent growth in the demand for digital transmission of speech, still images, and video has created potential applications for which viable hardware solutions are not available. Since a considerable degree of redundancy is present in these signals, and in many cases a signal perceptually equivalent but not identical to the original signal is acceptable, a decrease in the size of the digital representation by more than an order of magnitude is possible by employing suitable data compression techniques. While there is a considerable common theoretical basis for the compression of any of these signals, the compression of still greyscale images is the subject of this dissertation.

Digital still images are used in a wide variety of applications, including entertainment, advertising, journalism, security and law enforcement, medical imaging, and satellite imaging. Although many of these images are in colour, greyscale images predominate in medical imaging, and colour image compression is usually achieved by a relatively simple extension of the principles appropriate for greyscale images [115, pp. 664-665] [158, app. A]. The compression of colour or multi-band images is not considered here.

Despite the wide variety and considerable complexity of many data compression techniques, they are all based on the same simple principle, requiring a statistical characterisation of the data to be compressed, with which a measure of probability or frequency of occurrence is associated. In the case of reversible compression (in which decoding results in an exact reproduction of the original data) an average reduction in the length of the data representation as a string of symbols is achieved by using a shorter representation for more likely data, and a longer representation for less likely data. Irreversible compression (in which the decoded signal is merely similar to the original) achieves additional compression by discarding the least important components of the data, reducing the information to be represented. The majority of image compression techniques are based on a statistical image model which describes the statistical properties exploited by the compression scheme. Although these image models are usually only an approximation to the true image statistics, they do provide a theoretical justification for the resulting compression algorithm, as well as a reasonable characterisation

of the type of signal for which the algorithm is expected to be effective.

One of the most recent image compression techniques is so-called fractal compression, which is patented and marketed by its originator, who has made public only sketchy details of the design of the commercial software [16]. Fractal compression has been viewed with some skepticism within the engineering community, largely as a result of extravagant claims in the popular technical press of 1000:1 compression ratios [211] and "resolution independence" [6]. This has nevertheless become an active area of research, experiencing an exponential growth in publications over the last four years. The majority of research has been directed towards solving technical difficulties, such as the computational complexity of the encoding stage. In contrast, very little progress has been made towards understanding why fractal compression is as effective as it is, or characterising the type of signal for which it is effective. The research described in this dissertation is motivated by these questions.

1.1 Synopsis

A brief synopsis of the remainder of this dissertation is presented here as an aid to the reader. Chapter 2 consists of a broad overview of data and image compression. The basic elements of information theory are introduced, with a description of entropy and the rate distortion function. Examples of the most common reversible and irreversible coding methods are presented after a brief description of the properties of images relevant to their compression. The chapter closes with an introduction to fractal compression.

A survey of the current fractal coding literature, excluding colour image and video coding research, is presented in Chapter 3. The options available at each stage of algorithm design are summarised, with comments on their relative merits where possible, and the major theoretical aspects of the construction of a fractal representation are briefly described. The recent wavelet transform domain interpretation, having led to a new generation of improved fractal coding schemes, receives particular attention. Performance comparisons reveal that the best fractal compression algorithms are remarkably effective, motivating the necessity for an improved understanding of the image properties associated with this success.

Chapter 4 describes an evaluation of the effectiveness of fractal coding for a simple class of statistical signal models. This research provides valuable insight into the dependence of fractal coding on the statistical structure of the signals to which it is applied. The assumptions of fractal coding are found to be poorly adhered to for this class of signal models, resulting in significantly suboptimal performance. These results suggest that the conditions necessary for effective fractal coding are not a natural consequence of weak or common statistical assumptions.

The opposite approach is followed in Chapter 5 by defining a stochastic signal model with precisely the properties required for effective fractal compression. Examination of this

signal model reveals that the resulting second order statistics are approximately compatible with those observed for "natural" images, but that the model is poorly characterised by its second order statistics as a result of significant additional dependence. A wavelet transform domain version of this model is also constructed, and proves convenient for comparisons with properties of test images.

Chapter 6 builds on the wavelet domain form of fractal coding described in Chapter 3. The wavelet domain statistics of a set of test images are compared with those of the model defined in Chapter 5, and are observed to be approximately compatible. Direct measurement of the suitability of the test images for fractal compression suggest that they are only marginally so. The observed success of fractal coding for "natural" images is linked with the interpretation as a form of "self-quantisation" in the wavelet transform domain.

The conclusions are presented in Chapter 7, drawing together the results of the three previous research chapters, and presenting additional comments.

Elements of probability theory and statistics required for exposition of the results presented here are summarised in Appendix A. Appendix B contains a summary of results in functional analysis required for the following two appendices. A short introduction to wavelet theory is given in Appendix C, followed by the mathematical foundations of fractal coding in Appendix D. Appendix E contains reproductions of the set of test images used in the research.

Chapter 2

Data Compression and Signal Coding

Data compression algorithms may be conveniently classified into two distinct types, described as reversible or lossless compression if the original signal is exactly reconstructed on decoding, and irreversible or lossy compression if it is not. The type of algorithm utilised is dependent on the data to which it is applied. If the data consists of text, database files, program source or executable code, reversible coding is clearly necessary. When the data is a digitisation of inherently analog data such as speech or video, on the other hand, exact reconstruction may not be necessary, and considerable compression gains are possible by taking advantage of this additional freedom.

2.1 Reversible Coding

Digital data consists of a stream of discrete symbols over a finite alphabet[1] (e.g. $\{0, 1\}$ for binary data), as opposed to analog data which is, in principle at least, continuous in both time and amplitude. Each distinct string over this alphabet represents a separate "message", for example a piece of text, a segment of speech, or an image. An encoding system is a function mapping source strings to strings over a code alphabet. The goal of digital data compression is the construction of a code that minimises the average length of the encoded strings. Information theory allows bounds on the compression achievable for a particular source to be calculated from a statistical model of that source.

2.1.1 Information theory

Information theory is based on the work of Shannon, who formalised the notion of an information source as a stochastic process [179] (original emphasis):

[1] The actual symbols are clearly irrelevant; only the cardinality of the alphabet is significant.

> The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have *meaning*; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one *selected from a set* of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design.

The two main branches of information theory are concerned with *source* and *channel* coding. Source coding deals with encoding data for efficient transmission (or storage) with the shortest possible digital representation per message or data stream, whereas channel coding is concerned with reducing transmission (or storage) errors by encoding for maximum reliability. The joint source channel coding theorem [44, pp. 215-218] guarantees that, under very broad conditions, optimal transmission over a channel may be achieved by separate optimisation of the source and channel codes. Since channel coding is of secondary interest in this dissertation, the term "coding" shall, when unqualified, denote *source* coding.

The simplest model of an information source is a Discrete Memoryless Source (DMS), for which each symbol s_i of the source alphabet $S = \{s_1, s_2, \ldots, s_n\}$ has an associated probability[2] $p(s_i)$, which is independent of the previously generated symbols [158, pg. 16]. In defining the information content of a symbol generated by such a source, it is reasonable to expect an unlikely symbol to convey more information than a likely one. If the measure of information is also desired to be additive[3] for independent symbols, the definition of the information associated with symbol s_i is restricted (up to a multiplicative constant) [78, ch. 6] [179] to

$$I_b(s_i) = \log_b \frac{1}{p(s_i)},$$

where the base of the logarithm is usually 2, in which case the information is measured in *bits* (binary digits). The *entropy*[4] of a DMS, which may be considered to be the average information per source symbol, is[5]

$$H_b(S) = E[I_b(S)] = -\sum_{i=1}^{n} p(s_i) \log_b p(s_i)$$

where the subscript b is omitted for the base 2 entropy, which is denoted $H(S)$ here. If S is

[2] Although this should be denoted as $p_S(s_i)$ in the notation established in Appendix A, a slightly different notation contributes to clarity in this chapter.

[3] In the sense that the information conveyed by a number of symbols is the sum of the information conveyed by each of them.

[4] The entropy as defined here is closely related to the entropy of thermodynamics, where the term originated.

[5] Adopting the convention that $0 \log 0 = 0$, a symbol of zero probability is excluded from the sum.

a *continuous* valued memoryless source with pdf $f(s)$, the *differential* entropy [99, pg. 624]

$$h(S) = -\int_{\infty}^{\infty} f(s) \log_2 f(s)\, ds$$

may be defined, but does not have a simple interpretation corresponding to that for the absolute entropy for a discrete source [44, ch. 9].

A DMS is a rather simplistic model for most digital data, since there is usually significant dependence between neighbouring symbols (e.g. alphabetic characters in English text or pixel values in a raster scanned image). A kth order Markov process (a 0th order Markov process is simply a DMS) generates strings over a source alphabet S, where the probability of occurrence of each symbol is contingent on the previous k symbols [78, pg. 80], and is specified by the set of conditional probabilities $p(s_i|s_{j_1}, s_{j_2}, \ldots, s_{j_k})$. This Markov model may be viewed as a non-deterministic state machine of n^k states, where each state is specified by the immediate k symbol history $(s_{j_1}, s_{j_2}, \ldots, s_{j_k})$ at each stage (see Figure 2.1).

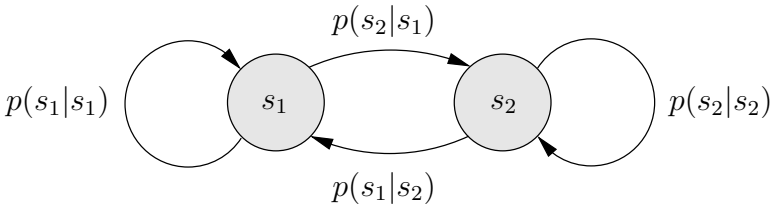


Figure 2.1: An example of a first order Markov model (note that $p(s_1|s_1) + p(s_2|s_1) = 1$ and $p(s_2|s_2) + p(s_1|s_2) = 1$).

The mth *extension* of an order k Markov model is obtained by considering m symbol blocks of the original model as a single symbol of the extension (see Figure 2.2) [1, pp. 29-30]. Any Markov model of order greater than one may be reduced to a first order model by using an appropriate extension of the higher order model.

$$s_2 s_3 s_2 s_1 s_4 s_1 s_1 s_3$$

$$\tilde{s}_{23} \tilde{s}_{21} \tilde{s}_{41} \tilde{s}_{13}$$

Figure 2.2: A source sequence and its second extension. The original sequence is over the alphabet $S = \{s_1, s_2, s_3, s_4\}$, while the second extension is a sequence over the alphabet $\tilde{S} = \{\tilde{s}_{11}, \tilde{s}_{12}, \tilde{s}_{13}, \tilde{s}_{14}, \tilde{s}_{21}, \ldots, \tilde{s}_{44}\}$.

Each state of an order k Markov model has associated entropy

$$H(S|s_{j_1}, s_{j_2}, \ldots, s_{j_k}) = -\sum_{i=1}^{n} p(s_i|s_{j_1}, s_{j_2}, \ldots, s_{j_k}) \log p(s_i|s_{j_1}, s_{j_2}, \ldots, s_{j_k}).$$

The entropy of the source is the average of the entropies for each state, weighted by the probability $p(s_{j_1}, s_{j_2}, \ldots, s_{j_k})$ of that state

$$H(S) = \sum_{S^k} p(s_{j_1}, s_{j_2}, \ldots, s_{j_k}) H(S|s_{j_1}, s_{j_2}, \ldots, s_{j_k}).$$

If the Markov process is *ergodic* [78, pp. 80-85], the state probabilities are expected to reach an equilibrium distribution independent of the initial state, in which case the entropy may be calculated. Ergodicity is necessary property for many types of statistical analysis, implying that averages obtained from an ensemble of signals may be estimated from a single signal of sufficient length [99, pg. 31] [151, pp. 245-251]. The importance of the source entropy as a lower bound on the achievable compression is discussed in the following section.

2.1.2 Source coding

A code represents a rule for translating source strings into code strings. If the source alphabet is $S = \{s_1, s_2, \ldots, s_n\}$, and the code alphabet is $U = \{u_1, u_2, \ldots, u_m\}$, a code is a function $C : S^* \rightarrow U^*$, where A^* is the set of all finite length strings over alphabet A, including the null sequence Λ.

A code translating individual source symbols into a sequence of code symbols is a *block*[6] code [1, pg. 46], while that assigning a code sequence to the entire source sequence is a *non-block* or *tree* code [158, pg. 28]. A code translating source symbols into strings of equal length over the code alphabet (e.g. ASCII) is a *fixed-length* code, whereas the code is a *variable-length* code if the code string length is variable (e.g. Morse code) [78, pg. 51]. The expected length of a code[7] is the expected encoded string length over all source symbols

$$L(C) = \sum_{i=1}^{n} p(s_i) l(C(s_i)),$$

where $p(s_i)$ is the probability of s_i, $C(s_i)$ is the code for s_i and $l(x)$ is the length of the string x. Clearly the block length is equal to $L(C)$ for a fixed-length code. While here is no advantage to be gained by employing a variable-length code if all source symbols have equal probabilities, it may be profitable to assign shorter codes to the more likely symbols if the probabilities are unequal, resulting in a smaller $L(C)$ than obtainable by a fixed-length code. The expected code length $L(C)$ is often referred to as the code *rate* for a specific code and source. In the case of a binary code, it represents the average number of bits required per source symbol.

[6] Usage differs in the case of some authors [78, pg. 51], for whom a block code corresponds to a fixed-length code here.

[7] The term "code" is used to indicate both the code as a function and the code as an encoded string; the difference should be clear from the context.

A number of constraints are required for a code to be useful. An important condition for decodability is clearly that the function C is injective[8], in which case the code is *non-singular* [44, pg. 80]. This is unfortunately not a sufficient condition to ensure unique decodability of a string unless a delimiting symbol is inserted between code symbols, which is inefficient. A code is *uniquely decodable* if all possible concatenations of source symbols result in a unique code sequence [78, pg. 53]. A condition of a slightly more practical nature is required in addition to unique decodability, since a uniquely decodable code may still require parsing of the entire code string before decoding is possible. A *prefix* or *instantaneous* code has no codeword as a prefix to any other codeword [44, pg. 81], which guarantees that a codeword in a string is recognisable as soon as that codeword has been completely scanned (see Table 2.1).

Table 2.1: An example of code taxonomy for a 3 symbol source $S = \{s_1, s_2, s_3\}$.

Non-singular	Uniquely decodable	Instantaneous	$C(s_1)$	$C(s_2)$	$C(s_3)$
			0	1	1
•			0	1	01
•	•		0	01	011
•	•	•	0	10	110

The definition of a prefix code imposes restrictions on the minimum lengths of codewords, since each code precludes the use of the same symbols as a prefix to any other code. The Kraft inequality [78, pg. 57] specifies necessary and sufficient conditions for the existence of an instantaneous code in terms of the codeword lengths; an instantaneous code C from an alphabet of n symbols into an alphabet of m symbols with codeword lengths $l_1, l_2, \ldots, l_n$ exists if and only if

$$\sum_{i=1}^{n} m^{-l_i} \leq 1.$$

The McMillan inequality [44, pg. 90], identical in form to the Kraft inequality, is valid for any uniquely decodable code. This implies that no reduction of codeword lengths is achievable by utilising a non-instantaneous code.

The Kraft inequality may be used to prove that the base m entropy of an information source is a lower bound for the average length of any m symbol instantaneous code for that source [78, pg. 112]

$$H_m(S) \leq L(C).$$

This is an extremely important result; there is no point in attempting to design an instantaneous code with average length less than the entropy of the source to be encoded. This

[8] Also termed "one-to-one", requiring in this context that no two source strings result in the same code string.

inequality suggests the definition of the *efficiency* η [1, pg. 86] of a code

$$\eta = \frac{H_m(S)}{L(C)}.$$

The *redundancy* of a code is $1 - \eta$.

It may be shown via the Kraft inequality that assigning a codeword length of $-\log p(s_i)$ to each source symbol s_i of a DMS results in a code with average length equal to the entropy bound [1, pp. 68-73] [44, pg. 85]. The actual codeword lengths must be chosen as[9] $\lceil -\log p(s_i) \rceil$ (block codes have integer codeword lengths), resulting in a coding rate which exceeds the entropy bound unless the $-\log p(s_i)$ values are integers. This integer codeword length assignment results in a coding rate bounded above by $H_m(S)+1$ [44, pp. 87-89]. Since this bound becomes $H_m(S)+\frac{1}{k}$ for the kth extension of the source [1, pg. 20, 29], the average codeword length may be made arbitrarily close to the entropy of the source by coding the source symbols in sufficiently large blocks [1, pp. 72-73].

The performance improvement obtained by grouping source symbols, despite the absence of any dependence between them, is due to the decreasing significance of non-integer symbol probabilities with increasing block size. The same choice of codeword lengths for a Markov source also results in an average codeword length approaching the source entropy with increased block length [1, pg. 75]. In this case the grouping together of source symbols in blocks allows the inter-symbol dependence to be utilised within each of the extension symbols.

2.1.3 Channel coding

While channel coding is not of direct interest in the study of data compression, a number of concepts and definitions which are most easily introduced within this context are required for rate distortion theory, which is introduced in the following section. Consider an information channel with source symbols in $A = \{a_1, \ldots, a_n\}$ (for simplicity, treatment is restricted to the case of a DMS) which are transmitted across a channel, and received as output symbols in $B = \{b_1, \ldots, b_m\}$. A Discrete Memoryless Channel (DMC) [26, pg. 18] is completely specified by the *transition probabilities* $p(b_j|a_i)$ of symbol b_j being received given that a_i was transmitted (see Figure 2.3).

The source (or *a priori*) entropy $H(A)$ of A has already been introduced as

$$H(A) = -\sum_{i=1}^{n} p(a_i) \log p(a_i).$$

Similarly, the entropy of the output B is

$$H(B) = -\sum_{j=1}^{m} p(b_j) \log p(b_j),$$

[9] The ceiling $\lceil x \rceil$ of $x \in \mathbb{R}$ is the smallest integer greater than or equal to x.

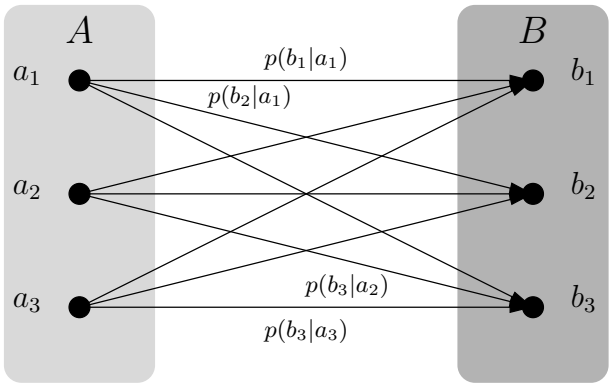


Figure 2.3: Transition probabilities for a channel between A and B.

where the probability of the output symbols is dependent on the source statistics as well as the transition probabilities

$$p(b_j) = \sum_{i=1}^{n} p(a_i)p(b_j|a_i)$$

and hence so is $H(B)$. The joint distribution [26, pg. 15]

$$p(a_i, b_j) = p(a_i)p(b_j|a_i)$$

is the probability of joint occurrence a_i and b_j. The conditional (or *a posteriori*) entropy [1, pg. 101] of A, given that symbol b_j is received at the output is

$$H(A|b_j) = -\sum_{i=1}^{n} p(a_i|b_j) \log p(a_i|b_j),$$

which is the average number of bits required to represent a source symbol given that b_j is received. Averaging over all output symbols gives the *equivocation* of A with respect to B [1, pg. 105]

$$H(A|B) = -\sum_{j=1}^{m}\sum_{i=1}^{n} p(a_i, b_j) \log p(a_i|b_j)$$

which is the average number of bits required to represent the input, given the output. On average, then, an output symbol conveys

$$I(A;B) = H(A) - H(A|B)$$

bits of information, the *mutual information*[10] of A and B.

[10]Note that $I(A;B) = I(B;A)$.

The *channel capacity* [78, pg. 146] of a channel with transition probabilities $p(b_j|a_i)$ is

$$C = \max_{p(a_i)} I(A;B)$$

where the maximum is taken over all possible source probabilities, and is therefore dependent on the channel characteristics $p(b_j|a_i)$ alone. Shannon [179] showed that data may be transmitted, with arbitrarily small frequency of error, at rate R along a channel of capacity C if $R < C$. Although this result is not of direct interest for the purposes of data compression, since source coding and channel coding may be considered as separate processes [26, pp. 71-75], the susceptibility of a particular source coding method to channel errors is often of importance.

The mutual information of two *continuous* random variables is

$$I(A;B) = h(A) - h(A|B),$$

and has the same interpretation as the mutual information for two discrete random variables [44, pg. 231].

2.1.4 Entropy coding

Although the theory of the preceding section links the entropy of a source to the minimum achievable average codeword length (or maximum data compression), it is non-constructive[11]. The entropy coding techniques presented here[12] all require *a priori* knowledge (or estimates) of the source statistics.

The Huffman coding algorithm [83] is capable of producing optimal instantaneous codes for a given source. While Huffman codes may be designed for any code alphabet [44, pg. 93], the description here is restricted to binary codes for simplicity. Due to the complexity[13] of Huffman coding of Markov sources of order higher than 0, the description is also limited to coding of a DMS. The initial stage of the process entails sorting the source symbols in order of probability. A tree is then constructed by combining the two least probable symbols at each stage into a single symbol (which is assigned the sum of its parents' probabilities) at the next stage. The construction of the tree terminates when two symbols remain. The code is generated by assigning a different code symbol to each of the top two symbols, followed by descending the tree, assigning partial codewords at each level by concatenating a different symbol at each branching to the partial codeword at the previous level. The process terminates when the bottom of the tree is reached, at which stage a codeword has been assigned to each of the source symbols (see Figure 2.4). Huffman codes may be shown to be optimal if the

[11] It does not provide an algorithm for constructing such an optimum code.
[12] Shannon-Fano coding [78] has been largely superseded by Huffman coding, and is not described here.
[13] A separate codebook is required for each state or *context*.

source probabilities are negative powers of the code radix (e.g. 2 for a binary code). When this is not the case, the restriction of codeword lengths to integer values prevents the entropy bound from being achieved (see the discussion at the end of Section 2.1.1).

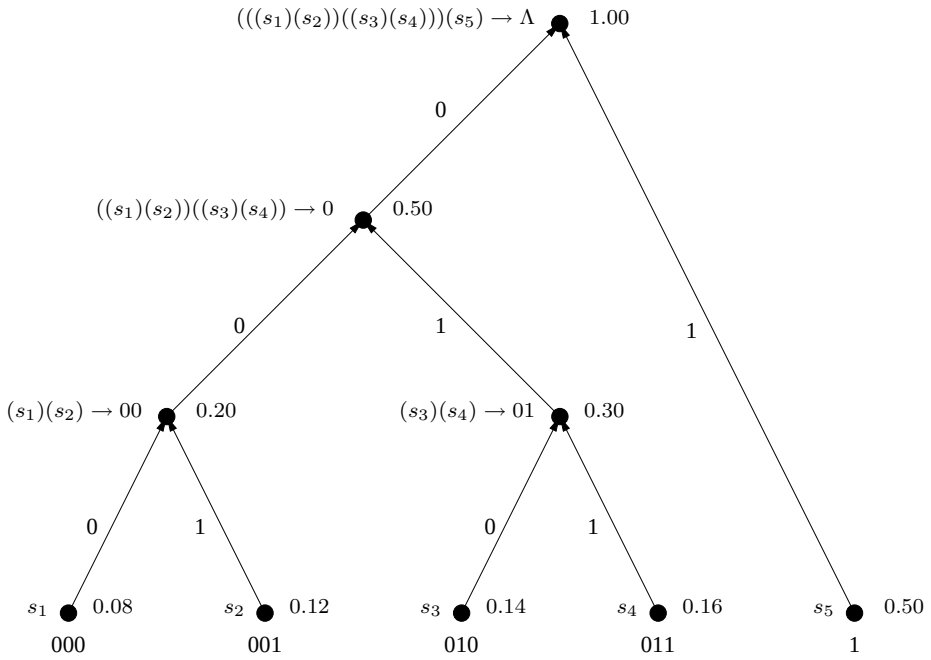


Figure 2.4: An example of Huffman code construction. Source symbols are to the left, probabilities to the right, and codeword assignments below the terminal nodes of the tree. Partial codewords and combined probabilities are to the left and right respectively of non-terminal nodes. Symbol assignments for each branch are to the left of that branch.

While Huffman coding is a block code, limiting its ability to match the source entropy for all sources, arithmetic coding [25] is a non-block code, thereby avoiding this limitation by assigning a code sequence to the source sequence as a whole. The precursor of arithmetic coding is Shannon-Fano-Elias coding [1, pg. 61] [44, pg. 101]. Arithmetic coding preserves the basic idea, but solves a number of problems of implementation. Assuming a DMS model of the source, with probabilities $p(s_i)$, clearly $\sum_{i=1}^{n} p(s_i) = 1$. The unit interval $[0, 1)$ is partitioned into subintervals of width corresponding to the probabilities of each source symbol, and the initial subinterval is selected according to the initial source symbol. This subinterval is then subdivided as before, and selection is performed on the next source symbol (see Figure 2.5). The construction ensures that each source sequence is represented by a subinterval of $[0, 1)$, and that the probability of a source sequence is equal to the width of the interval representing it. The lower bound of the interval may be represented by $\lceil -\log_2 p(x) \rceil$ bits, where $x \in S^*$ and the interval width is equal to $p(x)$ [158, pg. 29], resulting in a code with average length close to the entropy bound.

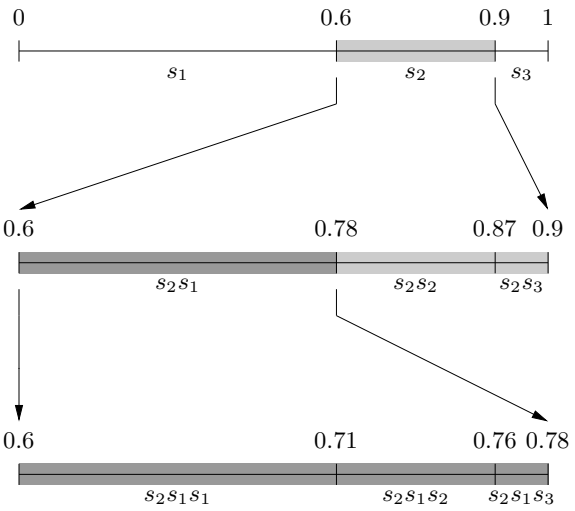


Figure 2.5: An example of arithmetic code construction. Interval limits are demarcated above the horizontal lines, and corresponding source strings are shown below.

2.1.5 Universal coding

The coding methods of the previous section require a statistical model of the information source. In many circumstances, however, such a model is difficult or impractical to construct, in which case universal coding provides an alternative. Universal coding methods include adaptive entropy coding schemes (such as adaptive arithmetic coding [164]) which build up a model of the source as a source sequence is scanned, and inherently universal schemes such as Lempel-Ziv coding [25], which is based on a complexity measure for individual strings, and does not directly involve estimation of source statistics.

The Kolmogorov complexity of a sequence is defined as the length of the shortest computer program generating that sequence [44]. This complexity measure is of theoretical interest, being in agreement with the entropy measure where comparison is possible. Aside from being impractical however, it may be shown to be non-computable in general. A practical measure proposed by Lempel and Ziv [111] depends on the rate of occurrence of novel subsequences in a sequence. This measure forms the basis of the most common universal coding algorithm. A sequence is parsed into previously un-encountered subsequences (see Figure 2.6), each of which is represented by a pointer to a previous subsequence, and an additional symbol (e.g. in Figure 2.6, sequence 011 would be represented by a pointer to the previous subsequence 01 and the additional symbol 1).

2.2 Irreversible Coding

In contrast with the symbol sources considered in the preceding sections, a *waveform* or *signal* is considered to be inherently time- and amplitude-continuous [99, pp. 2-3], and generally

Step	Dictionary Contents	Unparsed String
0	Λ	0011011001110101001
1	0	011011001110101001
2	0, 01	1011001110101001
3	0, 01, 1	011001110101001
4	0, 01, 1, 011	001110101001
5	0, 01, 1, 011, 00	1110101001
6	0, 01, 1, 011, 00, 11	10101001
7	0, 01, 1, 011, 00, 11, 10	101001
8	0, 01, 1, 011, 00, 11, 10, 101	001
9	0, 01, 1, 011, 00, 11, 10, 101, 001	Λ

Figure 2.6: An example of Lempel-Ziv sequence parsing.

represents some physically measurable quantity, such as sound amplitude, where the signal is a function of time, or light intensity across an image, in which case the signal is a function of spatial parameters. Discretisation in both time and amplitude is required in the construction of a digital representation of such a signal, referred to as sampling and quantisation[14] respectively. The meaning of "signal coding" is restricted here to the coding or re-quantisation of an existing digital representation. A consequence of the original sampling and quantisation process is that any digital reconstruction of an analog signal is necessarily lossy with respect to that signal. In many cases the fidelity of the digitised signal is sufficient to allow subsequent lossy compression without introducing unacceptable distortion.

2.2.1 Rate distortion theory

The preceding sections have dealt with *reversible* coding, where the decoder is able to reconstruct the original message exactly. In many circumstances, and in particular in the coding of quantised analog signals, a certain degree of information loss is acceptable (or even inevitable), as measured by some *fidelity criterion*. Rate distortion theory is the branch of information theory dealing with lossy encoding, subject to restrictions on the distortion between coded and reconstructed messages. The rate distortion function of a source is defined as the minimum rate required to achieve a specified distortion for that source [180]. Since the reconstruction alphabet of lossless coding is identical to the source, encoding and decoding may be represented by functions

$$C : S^* \to U^* \text{ and } C^{-1} : U^* \to S^* \tag{2.1}$$

[14] Also referred to as analog-to-digital conversion.

respectively, where S is the source alphabet and U is the code alphabet. The more general case of lossy encoding and decoding may be represented by

$$C : S^* \rightarrow U^* \text{ and } \hat{C} : U^* \rightarrow \hat{S}^* \tag{2.2}$$

where C and $\hat{C}$ are the encoding and decoding functions respectively, the reconstruction alphabet $\hat{S}$ is not necessarily equal to the source alphabet S, and in particular $\hat{C}$ is not necessarily the inverse of C. Development here is restricted to the case of *context free* or *single letter* [26, pg. 20] [70, pg. 443] distortion measures, involving only source and decoded symbols in corresponding positions in their respective strings. A more general distortion measure is more realistic for many applications, but is analytically less tractable. The distortion measure is a function $d : S \times \hat{S} \rightarrow \mathbb{R}$, where $d(s, \hat{s})$ is a measure of the cost incurred, or distortion produced, in representing $s \in S$ by $\hat{s} \in \hat{S}$. Common examples [44, pg. 339] are the Hamming distance

$$d(s, \hat{s}) = \begin{cases} 0 & \text{if} \quad s = \hat{s} \\ 1 & \text{if} \quad s \neq \hat{s} \end{cases}$$

and the squared error[15]

$$d(s, \hat{s}) = (s - \hat{s})^2.$$

The definition of the rate distortion function presented here is further restricted to the simple case of a DMS; sources with memory require a slightly more complicated definition, involving the limit of mutual information for large block size.

The fundamental idea of rate distortion theory is to model the code-decode function by a set of transition probabilities $p(\hat{s}_j|s_i)$ on a *test channel* providing the connection from S^* to $\hat{S}^*$ in Equation (2.2). In most practical situations this mapping is deterministic, corresponding to transition probabilities of 0 or 1 only, but the additional flexibility provided by a stochastic mapping is important in optimising over *all possible* mappings. Since the rate distortion function is a property of the *source*, the optimisation required in its definition is over all possible transition probabilities (representing all possible code-decode schemes), in contrast to the optimisation in the calculation of channel capacity, which is over all possible sources, the *channel* characteristics being fixed.

The average distortion [70, pg. 445] introduced by the process associated with Equation (2.2) is

$$\begin{aligned} E[d(S, \hat{S})] &= \sum_i \sum_j p(s_i, \hat{s}_j) d(s_i, \hat{s}_j) \\ &= \sum_i \sum_j p(s_i) p(\hat{s}_j|s_i) d(s_i, \hat{s}_j), \end{aligned}$$

[15] Clearly the subtraction operation must be defined between elements of S and $\hat{S}$.

where $p(s_i, \hat{s}_j) = p(s_i)p(\hat{s}_j|s_i)$ is the joint probability of s_i. The rate distortion function of the source S relative to distortion d is [26, pg. 23]

$$R(D) = \min_{\{p(\hat{s}_j|s_i) \mid E[d(S,\hat{S})] \leq D\}} I(S;\hat{S})$$

where the minimisation is over all transition probabilities $p(\hat{s}_j|s_i)$ resulting in an average distortion less than D. The source coding theorem states that $R(D)$ of a source is the minimum coding rate required to code the source with a distortion of less than or equal to D, and its converse that D is the minimum distortion achievable for a coding rate of R [44, ch. 13]. The rate distortion function $R(D)$ may be proved to be a decreasing and convex function of D (see Figure 2.7). If the distortion measure assigns a zero distortion to perfect reconstruction, then $R(0) = H(S)$, the source entropy [26, pg. 8]. As in the case of lossless coding, it may be shown that the theoretical rate distortion limit may be approached in practice by jointly encoding sufficiently large blocks of symbols [26, pp. 66-71].

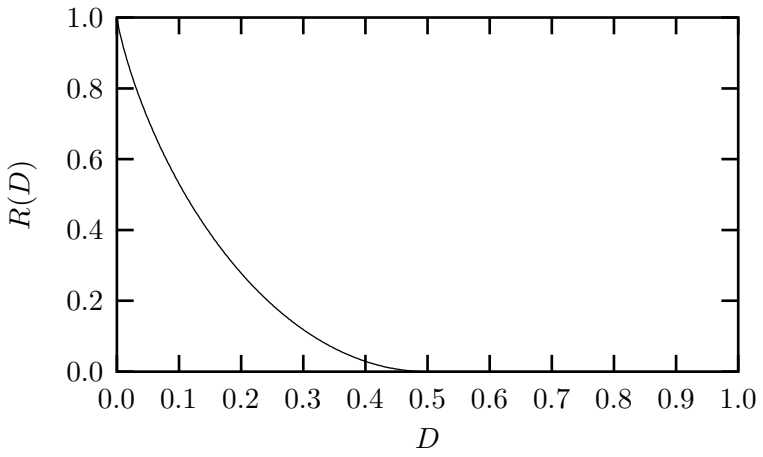


Figure 2.7: Rate distortion function (with Hamming distortion measure) for a source of two equiprobable symbols [44, pp. 342-344].

An alternative, and often more useful approach is to consider the *distortion rate* function $D(R)$ [99, app. D]

$$D(R) = \min_{\{p(\hat{s}_j|s_i) \mid I(S;\hat{S}) \leq R\}} E[d(S,\hat{S})],$$

which represents the minimum distortion D possible by representing source S at rate R. The distortion rate function, with MSE distortion measure, for a memoryless Gaussian source X with $\mu_X = 0$ is (see Figure 2.8)[16] [99, app. D]

$$D(R) = 2^{-2R}\sigma_X^2.$$

[16]Log scaled plots are convenient in comparing the rate distortion functions of other sources with those of Gaussian sources, for which $\log D$ is linear in R.

This function constitutes an upper bound for any other memoryless source. The Shannon lower bound is [99, app. D]

$$D(R) = \frac{2^{-2(R-h(X))}}{2\pi e}.$$

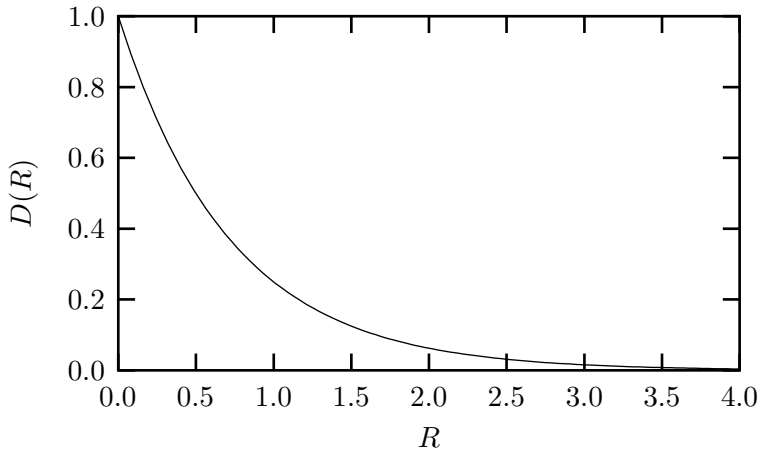


Figure 2.8: Distortion rate function (with MSE distortion measure) for a memoryless Gaussian source with $\mu = 0$ and $\sigma^2 = 1$.

Calculation of $R(D)$ or $D(R)$ is analytically intractable for all but a few simple cases, but the results available, as well as numerical algorithms for more complex cases, are useful for providing estimates of achievable rate distortion curves for practical situations. It is important to distinguish between distortion rate curves $D(R)$, which are calculated for a particular source, and the experimental distortion versus rate curves $D'(R)$ often calculated for coding of an image or set of images by a specific algorithm [99, app. D].

2.2.2 Scalar quantisation

Lossless compression of a single discrete random variable is achieved on average by variable length coding, assigning the shortest codes to the most likely values of the random variable. An obvious requirement is that the original and reproduction alphabets are the same, as in Equation (2.1). In lossy coding of a discrete or continuous random variable, on the other hand, insufficient bits are available to represent the full range of the source alphabet in the reconstruction alphabet, as in Equation (2.2). The reconstruction alphabet is usually referred to as the set of reconstruction levels or codebook; the encoder represents each source sample by the index of the closest reconstruction level, while the decoder reconstructs the sample as the value of the specified reconstruction level. The decision boundaries determined by this procedure are known as *Voronoi cells*. If there are L reconstruction levels in the codebook, the bit rate of the code is

$$R = \log_2 L.$$

The full encode-decode operation may be represented as a single function $y = Q(x)$ where x is the source value and y is the quantiser output value.

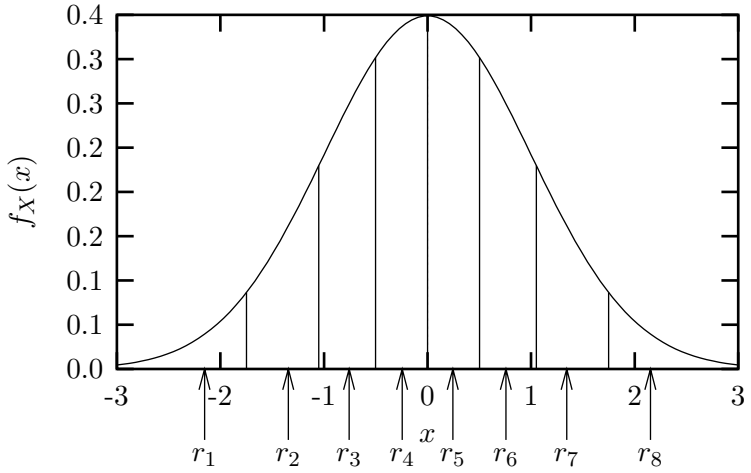


Figure 2.9: Optimal (with mean-square distortion) scalar quantisation reconstruction levels r_i and decision boundaries for scalar quantisation of a Gaussian pdf ($\mu = 0,\ \sigma = 1$) at a rate of 3 bits per sample.

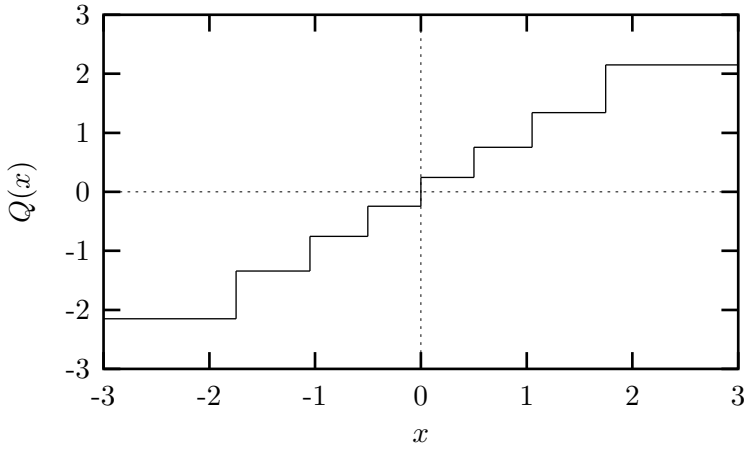


Figure 2.10: Scalar quantisation function $Q(x)$ corresponding to the reconstruction levels and decision boundaries in Figure 2.9.

Uniform quantisation is the simplest solution, assigning equally spaced reconstruction levels on a fixed interval [99, ch. 4]. An optimum quantiser [131] is achieved by ensuring that the reconstruction levels are denser in regions of high probability (see Figures 2.9 and 2.10). The *centroid condition*, which states that the optimum reconstruction level for a Voronoi cell is at the centroid (with respect to the source probability distribution) of the cell, forms the basis of the Lloyd [120] algorithm for quantiser optimisation [73, ch. 6]. Conditions for optimal decision boundaries b_i ($b_1 = -\infty,\ b_{L+1} = \infty$) and reconstruction levels r_i are [99,

pp. 131-132]

$$b_i = \frac{r_{i-1}+r_i}{2} \qquad i \in \{2, 3, \ldots, L\}$$
$$r_i = \frac{\int_{b_i}^{b_{i+1}} x f_X(x)\,dx}{\int_{b_i}^{b_{i+1}} f_X(x)\,dx} \qquad i \in \{1, 2, \ldots, L\}.$$

Given an initial estimate of the r_i, improved estimates of b_i and r_i may be calculated by iteratively applying the above relations; this procedure converges to a global optimum for the Uniform, Gaussian and Laplacian pdfs [99, pp. 131-132].

2.2.3 Vector quantisation

Vector Quantisation (VQ) is a generalisation of scalar quantisation to the joint quantisation of a vector of scalar values. Each reconstruction level or codebook entry is a vector, and the bit rate for an L-entry codebook of n-vectors is

$$R = \frac{\log_2 L}{n}.$$

The advantages of VQ over scalar quantisation are obvious where dependence exists between the scalar elements of a vector. The Voronoi regions for the distribution depicted in Figure 2.11 are displayed in Figure 2.12, illustrating how the codebook vector distribution takes advantage of the non-uniform distribution of the points in Figure 2.11.

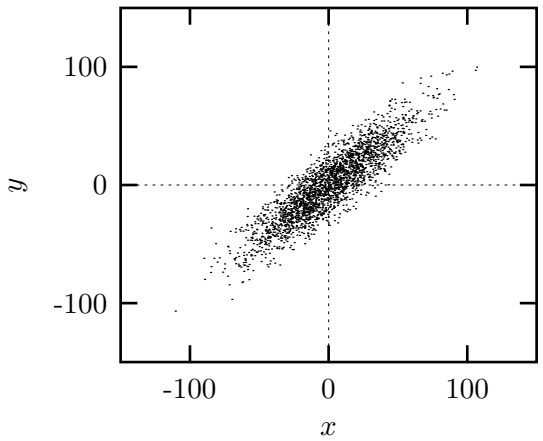


Figure 2.11: Scatter plot of (x, y) points distributed according to a correlated multivariate Gaussian distribution.

The distribution depicted in Figure 2.11 may be decorrelated by an appropriate linear transform (see Section 2.5.2), which in this case results in coefficients that are not only uncorrelated but also independent, suggesting that scalar quantisation of the transform coefficients is as efficient as VQ of the original samples. VQ is, however, more efficient than scalar quantisation even for independent scalars. This is illustrated in Figure 2.13; the hexagonal arrangement of Voronoi cells covers the plane more efficiently, resulting in a small decrease in

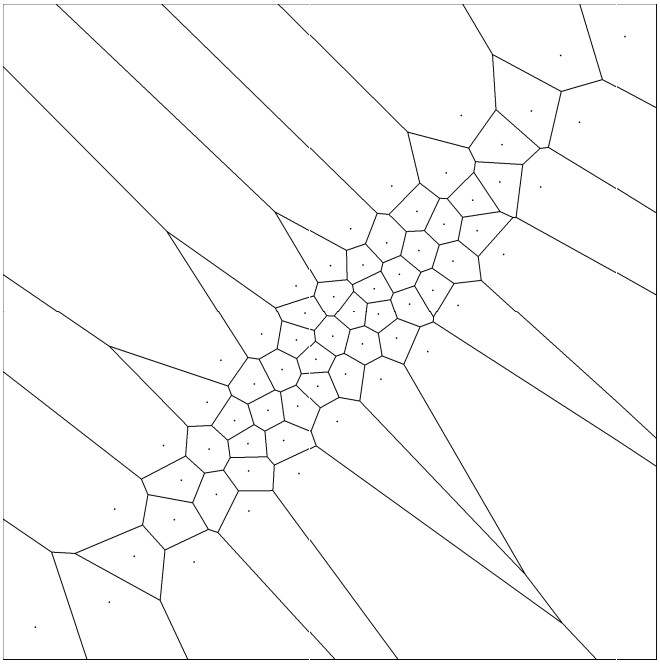


Figure 2.12: Voronoi regions and corresponding reconstruction levels for correlated multivariate Gaussian distribution as in Figure 2.11.

average distortion. The efficiency improvement of VQ over scalar quantisation increases with increasing vector dimensionality.

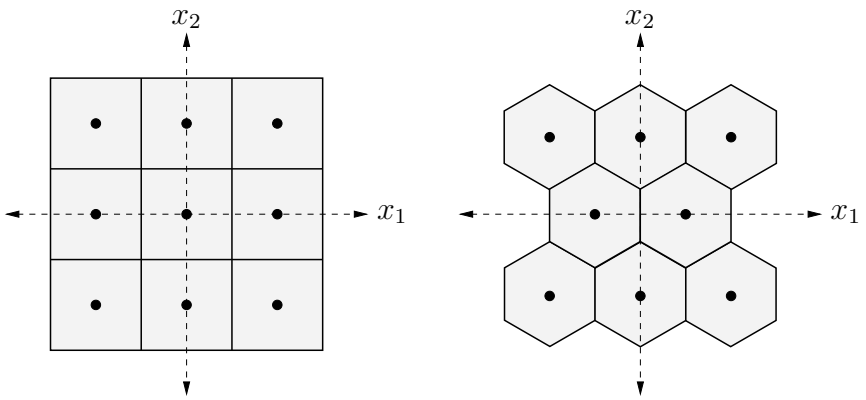


Figure 2.13: Voronoi regions for scalar and vector quantisation of uncorrelated scalars.

An optimum codebook for a given rate R is obtained by optimising the locations of the L reconstruction levels by minimising the expected distortion with respect to a known or estimated probability distribution. The codebook is usually constructed by the Generalised Lloyd Algorithm[17] (GLA) [73, ch. 11] described in Table 2.2. An initial codebook is generated by one of a variety of procedures [73, ch. 11], which include a randomly generated codebook, one generated by "splitting" a smaller codebook (starting from a single vector), or based on

[17] Also known as the Linde-Buzo-Gray (LBG) or k-means algorithm [115, pg. 607].

a merging procedure in a cluster of training vectors [59]. The GLA, which iteratively adjusts this codebook to minimise the average distortion in representing a set of *training vectors*, converges to a local optimum dependent on the initial codebook.

Table 2.2: The Generalised Lloyd Algorithm

Start with: Vector space X
Training set $T = \{\mathbf{t}_1, \ldots, \mathbf{t}_m\} \subset X$
Initial codebook $C = \{\mathbf{c}_1, \ldots, \mathbf{c}_n\} \subset X$
Distortion measure $d(\mathbf{u}, \mathbf{v})$ $\mathbf{u}, \mathbf{v} \in X$

1. Determine the Voronoi regions by assigning each training vector $\mathbf{t}_i$ to the nearest codebook vector $\mathbf{c}_j$.
2. Construct a new codebook consisting of the centroid vectors of the training vectors in each Voronoi region (the centroid for the MSE distortion measure is the arithmetic mean vector of the training vectors in a Voronoi region).
3. Calculate the average distortion

$$D = \frac{1}{m} \sum_{i=1}^{m} \min_j d(\mathbf{t}_i, \mathbf{c}_j).$$

Halt if the fractional change in the distortion is below a predetermined threshold, otherwise repeat the process.

The simplest application of VQ to signal coding entails forming separate vectors from consecutive non-overlapping blocks of fixed size within the signal. While VQ is theoretically optimal according to rate distortion theory, the complexity necessary to approach the theoretical limit, which requires large vector sizes, is often prohibitive in practice, since codebook storage requirements, and the computation required in codebook construction and searching, grow exponentially with the number of codebook entries [115, pp. 608-609]. A wide variety of VQ techniques [73] other than the simple form described here are utilised for this reason.

2.3 Image Compression

Data compression theory has thus far been introduced in a very general context, without concentrating on specific applications. There are, however, requirements peculiar to image compression which must be taken into account when designing image compression systems. The characteristics of images, as opposed to other commonly compressed signals, such as recorded speech, are also important in designing efficient coding algorithms.

Images are subjected to both lossless and lossy compression, depending on the application. Lossless compression is used where any degradation is unacceptable, as in the transmission

of reference images. This form of compression is often applied to medical images, since the effect of lossy coding artifacts on diagnostic accuracy is an important consideration. Since images are reproduced exactly, the primary goal of a lossless coding system is to maximise the achievable compression, specified either in terms of the *compression ratio*, which is the ratio of the original representation size to the encoded size, or the rate (bits per pixel) for a particular image (or an average over an ensemble of images). Lossy coding, which is the type of primary interest in this dissertation, offers significantly greater gains in compression ratio.

The three main criteria in the design of a lossy image compression algorithm are desired bit rate (or compression ratio), acceptable distortion, and restrictions on code and decode time. While different algorithms produce different types of distortion, the acceptability of which is often application dependent, there is clearly an increase in distortion with decreasing bit rate. Certain applications may require restricted code and decode times, which place an additional burden on the designer of a coding algorithm. An algorithm with similar code and decode times is called symmetric; in many cases the decode time is far more critical than the encode time and a symmetric algorithm is not required. Additional requirements are often imposed, such as the ability to perform progressive transmission, in which progressively more accurate reconstructions of the original image may be produced as the encoded representation is achieved.

2.3.1 Image acquisition

Images are most commonly formed by the measurement of electro-magnetic radiation[18] focussed onto a detector unit, resulting in a discrete function on a two dimensional support, either by mechanical translation of the detector or focusing arrangement, or by an array of detectors within the composite unit. The sample points may be scalars, as in the case of monochrome (greyscale) images, or vectors as in the case of colour images with separate values for red, green and blue at each sample point.

Discretisation of the image support is determined by the composition of the sensor array (or by the mechanical means of simulating such an array), and necessarily involves integration of the measured radiation over the finite sensor element area. The individual discrete elements of the image support are referred to as picture elements or *pixels*[19], and are usually arranged in a regular rectangular array. Digitisation of individual pixel values may include a logarithmic scaling to improve dynamic range.

[18] Exceptions include Scanning Tunnelling Electron Microscopy and Magnetic Resonance Imaging.
[19] An alternative term is *pels*.

2.3.2 Image data representation

The storage of an image within computer memory is dependent on the type of image and the required image size (the width and height are often powers of 2 for practical reasons[20]) and resolution. Although pixels often take on floating point values during processing, unsigned integer valued pixels are standard for storage and display; an image with pixel values between 0 and $2^k - 1$ requires k bits per pixel, and the image is said to be of depth k.

Documents are usually digitised as binary images (1 bit per pixel) for facsimile transmission or storage since the only distinction necessary is between black and white pixels. Monochrome images are commonly represented at 8 bits/pixel, although 12 bits/pixel are often utilised for X-ray images, which have a high dynamic range. Colour images may be represented by three image planes (one each for red, green and blue) of 8 bits/pixel each, resulting in a total of 24 bits/pixel, or by 8 bits/pixel for an index into a colour lookup table. It is important to emphasise that all of these quantities refer to memory allocation for images in uncompressed or canonical form, rather than the rate for coded images.

2.3.3 Statistical properties of images

A measure of the redundancy present in a particular signal source is useful in signal coding, providing an estimate of the potential coding gains from lossless coding, or an appropriate lower bound for these gains in lossy coding. Kersten [102] performed experiments on images (comparable to those performed by Shannon to estimate the entropy of the English language [44, pp. 138-140]), in which a certain fraction of pixels in a set of images (128×128 pixel, depth of 4 bits/pixel) were deleted, and an observer was required to guess the original value. Although valid for a reasonably small set of images, the resultant redundancy estimate of 46% to 74% provides a rough estimate of the compression ratios to be expected for lossless coding.

The complete characterisation of image statistics by a discrete Markov model would impose intractable computational and storage requirements for the transition probabilities for most image sets. Nevertheless, results are available for low order Markov models and images of limited depth [116] [177]; the measured decrease in entropy reduction with increasing order of Markov model suggesting that there are rapidly diminishing returns involved in utilising models of order higher than 3 or 4.

In the face of these practical difficulties the usual approach in signal coding is to partially characterise the signal by statistics such as the mean, variance and autocorrelation. Although this is vastly more practical than the previous approach, simplifying assumptions are usually required. In particular, since it is difficult in many cases to assemble a sufficiently large ensemble of images from which ensemble statistics may be calculated, it is common to assume at least wide-sense stationarity and ergodicity so that the required statistics may be estimated

[20] Transforms such as the Fast Fourier Transform (FFT) are simpler to implement for dimensions equal to a power of 2.

by averaging within individual images in the ensemble.

Although Higher Order Statistics (HOS) [132] may be considered, image models are usually specified in terms of the correlation structure. A one-dimensional simplification is possible by considering each scan-line of an image individually, in which case a first order Markov model with $\rho(k) = \rho^{|k|}$ is often a reasonable approximation to measured image statistics for $\rho \approx 0.95$ [37, ch. 2]. The simplest extension to two dimensions is the separable model $\rho(k,l) = \rho_H(k)\rho_V(l)$ in terms of separate horizontal and vertical correlations, where the one-step correlation is again typically in the region of 0.95 [96]. A more complicated non-separable "isotropic" model [37, ch. 2] [99, ch. 2] [129] provides a considerably better fit to measured correlations.

2.4 Lossless Image Compression

Lossless image compression is required in a variety of situations, including the storage of standard test images, where exact duplication is required, and medical images, where the effect of degradation on diagnostic utility is difficult to quantify[21], and legal issues may be of concern.

2.4.1 Run length coding

One of the most common applications of image compression is in facsimile transmission, where a standard compression algorithm has been adopted [97, pp. 540-551] [99, ch. 10]. Since binary images of documents consist of alternating *runs* of zeroes and ones, it is profitable to represent each scan line by the lengths of these runs (see Figure 2.14). The resulting *run length code* is often compressed using Huffman coding, with separate code tables for runs of zeroes and runs of ones. Extension of run length coding to two dimensions is also possible [75, pp. 355-356].

2.4.2 Predictive coding

The first widely used signal coding technique was Pulse Code Modulation (PCM) [99, ch. 5], consisting of the independent digitisation and coding of signal samples. Limited compression is achievable, however, since inter-sample dependence is completely ignored.

The dependence between pixels may be taken into account by coding the *prediction error* for each pixel, using a prediction based on the values of previously encountered pixels in each scan line[22]. The optimum prediction of sample X_N given previous samples $X_0, X_1, \ldots, X_{N-1}$ requires the conditional probabilities $p(X_N \mid X_0, X_1, \ldots, X_{N-1})$, the computation of which is usually infeasible [99, pg. 254] [158, ch. 7]. Linear prediction provides a practical alternative,

[21] These effects have been investigated [42] due to the considerable compression gains possible by utilising lossy compression.

[22] Two-dimensional prediction is also possible by consideration of previously encountered scan lines [99, ch. 6].

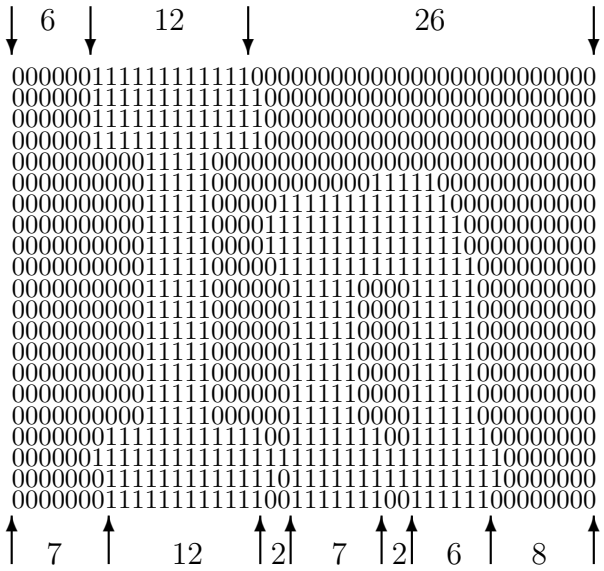


Figure 2.14: Run length representation of first and last scan lines in a subregion of a binary image of a facsimile document.

only requiring knowledge of the autocorrelation function $R_X(k)$ of the scan lines [73, ch. 4] [99, pp. 267-270] [197, pp. 454-456]. The coding of the resulting linear prediction errors is known as Differential PCM (DPCM).

Since DPCM prediction errors are often accurately modelled by the Laplacian distribution (see Appendix A) [158, pg. 62], it is possible to construct an adaptive Huffman code requiring only the measured variance, rather than the entire codebook, as side information [158, pg. 62]. The Rice Code [196] is an adaptive algorithm based on a set of predefined Huffman codes designed for this distribution.

2.5 Lossy Image Compression

Lossy coding is necessary when the desired bit rate is insufficient to support lossless coding. The degradation due to lossy coding is often considered insignificant if it is either visually imperceptible (in which case the coding is termed *perceptually lossless*), or if the lost information corresponds to the image noise (there is no point in attempting to preserve the bits representing noise, which by their random nature, are immune to compression).

2.5.1 Fidelity measures

The appropriate measure of image fidelity is application dependent, based either on a subjective measure such as perceived distortion [99, app. F], or on some more objective but complex measure such as diagnostic accuracy in the case of medical images. These measures are impractical in many circumstances, where an objectively determined and computable measure

is required [99, app. E].

The most widely utilised distortion measures are the *mean-squared error*

$$\mathrm{MSE}(\mathbf{u}, \mathbf{u}') = \frac{1}{n} \sum_{j=0}^{n-1} \left(u_j - u'_j\right)^2$$

and derived measures, where $\mathbf{u} = (u_0, u_1, \ldots, u_{n-1})^T$ and $\mathbf{u}' = (u'_0, u'_1, \ldots, u'_{n-1})^T$ are n-vector representations[23] of the original and reconstructed images respectively. Measures derived from the MSE are the *root-mean-squared error*

$$\mathrm{RMSE}(\mathbf{u}, \mathbf{u}') = \sqrt{\mathrm{MSE}(\mathbf{u}, \mathbf{u}')},$$

the *signal-to-noise ratio* [97, pg. 59]

$$\mathrm{SNR}(\mathbf{u}, \mathbf{u}') = 10 \log_{10} \frac{\sigma_{\mathbf{u}}^2}{\mathrm{MSE}(\mathbf{u}, \mathbf{u}')},$$

where $\sigma_{\mathbf{u}}^2$ is the deterministic variance[24] of the original image $\mathbf{u}$, and the *peak signal-to-noise ratio* [158, pg. 77]

$$\mathrm{PSNR}(\mathbf{u}, \mathbf{u}') = 10 \log_{10} \frac{\Delta_{\mathbf{u}}^2}{\mathrm{MSE}(\mathbf{u}, \mathbf{u}')},$$

where $\Delta_{\mathbf{u}}$ is the difference between maximum and minimum pixel values[25] of image $\mathbf{u}$. The SNR and PSNR are both measured in *decibels* (dB).

The MSE and its derived measures do not accurately reflect subjective or perceived distortion, but are frequently utilised since they are easily computed, and provide an approximate[26] basis for comparison between images. Improved measures, based on the properties of the Human Visual System (HVS) [142, ch. 4] [178, ch. 3] have been proposed [117], but as yet no numerically computable distortion measures which accurately model subjective evaluations are known. The HVS is known to have a non-uniform spatial-frequency[27] response, with a peak between 4 and 8 cycles per degree of viewing angle [37, ch. 6], and distortion measures utilising this response in the frequency domain [127] have achieved improved performance over the MSE error.

[23] In many contexts an object with two indices, such as an image, may be represented as a vector by collapsing it to an object with one index [156].

[24] That is, the variance over the ensemble consisting of the values of all the pixels in the image.

[25] A k bit image with pixel values taking on the entire range of possible values has $\Delta_{\mathbf{u}} = 2^k - 1$.

[26] Although two different reconstructions of the same original image and with similar MSE values may differ considerably in perceived distortion, a large difference in MSE is likely to correspond to a significant perceptual difference.

[27] That is, the frequency of a spatially periodic pattern.

2.5.2 Transform coding

The aim of transform coding is to reduce, or ideally remove, inter-pixel correlations in an image by the application of a linear transform to the image, or individually to subblocks of the image. Consider the representation of blocks of two[28] adjacent pixels in Figure 2.15, where the values of the first and second pixel in each block are represented by x_1 and x_2 respectively. High inter-pixel correlation implies that most of the points in this space lie near the line $x_2 = x_1$, with coordinates x_1 and x_2 having similar variances. Transforming to a new basis, however, results in coordinate x'_1 having considerably greater variance than coordinate x'_2. This is referred to as *energy packing* [37, pg. 72], allowing the data to be compactly represented with minimal error by discarding, or representing at reduced accuracy, the least important coefficients.

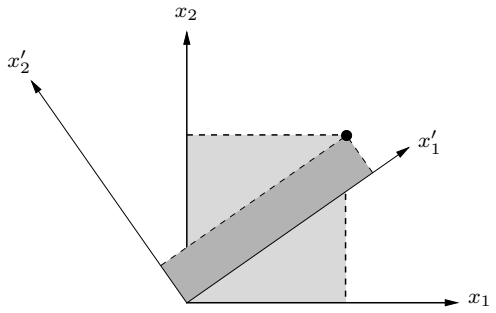


Figure 2.15: Change of basis by rotation of axes.

The coordinates $\mathbf{u}'$ of a vector $\mathbf{u}$ represented in some basis are obtained by the linear transform $\mathbf{u}' = A\mathbf{u}$, where the rows of matrix A are the basis vectors. If $A^{-1} = A^T$, the transform is *orthogonal*, and the original vector may be reconstructed from the transform coefficients as $\mathbf{u} = A^T\mathbf{u}'$. The signal "energy" represented by the Euclidean norm $\|\mathbf{u}\|$ is preserved by an orthogonal transform, since $\|\mathbf{u}\| = \|A\mathbf{u}\|$ for any norm if $A^{-1} = A^T$.

The optimum transform in the energy packing sense is the Karhunen-Loève Transform (KLT) [37, pp. 91-97] [97, pp. 163-168], which diagonalises the autocovariance matrix $C_{\mathbf{X}}$ of source $\mathbf{X}$. The KLT is, however, unsuitable for image coding purposes, since there are, in general, no fast algorithms (such as the FFT) for its computation, and there are practical difficulties in computing the basis vectors for vectors of high dimensionality.

Many fixed basis orthogonal transforms, such as the Hadamard, Haar, Discrete Fourier, Sine and Discrete Cosine transforms [75, ch. 3] exhibit the energy packing property. The Discrete Cosine Transform (DCT) is the most effective [97, pp. 150-154] for normal image characteristics, having the theoretical advantage that the KLT of a first order stationary Markov model with correlation coefficient close to unity is very similar to the DCT [2]. The

[28] While considerably larger blocks, of size 8×8 or larger are used in practice, such a visual representation becomes impossible for these sizes.

basis vectors $\{\mathbf{e}_0, \mathbf{e}_1, \ldots, \mathbf{e}_{n-1}\}$ of the DCT for n-vectors are

$$\mathbf{e}_k = \{e_{k,0}, e_{k,1}, \ldots, e_{k,n-1}\}$$

where[29] [73, pg. 245] [99, pg. 558]

$$e_{k,l} = \sqrt{\frac{2 - \delta_{0,k}}{n}} \cos\left[\frac{k\pi}{n}\left(l + \frac{1}{2}\right)\right].$$

The Fourier coefficient (see Appendix B) of basis vector $\mathbf{e}_0$ is commonly referred to as the "DC" coefficient[30], since the corresponding basis vector has constant entries, while the other coefficients are the "AC" coefficients.

A *separable* linear transform of an image is achieved by separately transforming the rows and then the columns of the image. The DCT of an image represented by the $n \times n$ matrix U is computed as $U' = AUA^T$, where A is the DCT transform for n-vectors defined above.

Compression is not actually achieved by an orthogonal transform itself, but by the subsequent quantisation of the transform coefficients. Since the reconstruction error is equal to the average quantisation error of the transform coefficients [99, pp. 524-535], the optimum bit allocation may be determined by minimising the average quantisation error over all of the n coefficients, which results in

$$R_k = R + \log_2 \sigma_k - \frac{1}{n} \log_2 \left(\prod_{i=0}^{n-1} \sigma_i\right), \tag{2.3}$$

where σ_k^2 and R_k are the variance and bit allocation for coefficient k respectively, and R is the total available bit allocation [37, ch. 4]. Some modification is necessary for practical application since this procedure may allocate a negative or non-integer number of bits; these adaptations range from simple heuristic procedures to efficient optimisations [183] [206].

An alternative bit allocation strategy is *threshold sampling* where any coefficient with a value below a predetermined threshold is discarded [99, pp. 565-566]. Bit allocation may be uniform, or dependent on the coefficient position [158, pg. 113], and run length coding is often utilised to represent the values of the retained coefficients separated by the zero-valued (discarded) coefficients.

Since the HVS has been shown to have a non-uniform spatial-frequency response, it is reasonable, at least in the case of the Discrete Fourier Transform (DFT), to improve perceived image fidelity by weighting bit allocation for a transform coefficient according to its perceptual importance. This *perceptual* or *psycho-visual* coding has been tested in conjunction with a

[29] There are in fact four different DCT transforms [189, pp. 276-281], the most effective for image coding being defined here.

[30] Since all of the AC basis vectors are zero-mean, the DC coefficient determines, using analog terminology, the "DC offset" of a signal.

number of orthogonal transforms, with varying results [37, ch. 6]. The applicability of frequency domain HVS response to DCT coefficients has been shown analytically [143], and may be utilised to improve perceived reconstruction quality [77].

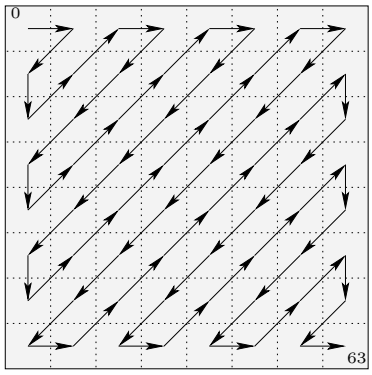


Figure 2.16: Zig-zag ordering of DCT coefficients for an 8×8 pixel image block, as applied in the JPEG system.

The JPEG[31] [153] lossy compression standard is based on DCT coding of 8×8 image blocks in a tiling of the full image. To simplify somewhat, a form of threshold sampling is applied by dividing each DCT coefficient by the corresponding entry in a *quantisation table* (based on measured HVS response) before rounding to the nearest integer. The normalised values are then scanned in *zig-zag* order (see Figure 2.16) before Huffman or arithmetic coding [158, pp. 113-128].

2.5.3 Subband and wavelet coding

A subband decomposition and reconstruction of a signal is generated by passing it through a multichannel filter bank (a two-channel filter bank is illustrated in Figure 2.17; the extension to a higher number of channels is obvious) in which the initial *analysis* filters are followed by a downsampling operation, and an upsampling operation precedes the final *synthesis* filters [4] [189] [197]. The analysis filters are selected so that each channel represents a separate frequency subband, while the synthesis filters are selected such that, in the absence of any processing of the output of the analysis stage, the original signal is reconstructed exactly by the synthesis stage.

The filter with coefficients $\{h_i\}$ is implemented by the usual convolution operation $y_i = \sum_j h_j x_{i-j}$, where the the input and output signals are represented by the sequences $\{x_i\}$ and $\{y_i\}$ respectively. Downsampling of $\{x_i\}$ by a factor of 2 to give sequence $\{y_i\}$ is defined such that $y_i = x_{2i}$, while the corresponding upsampling operation is defined such that $y_{2i} = x_i$ and $y_{2i+1} = 0$.

Since a subband decomposition is computed by linear operations, it is in fact the result of

[31] Joint Photographic Experts Group

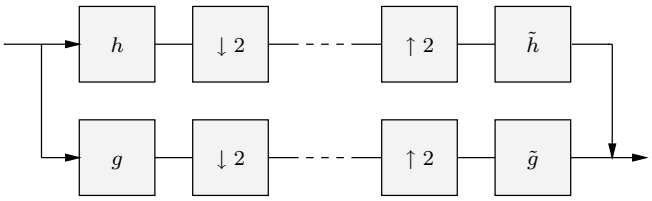


Figure 2.17: A two-channel filter bank. The analysis filter coefficients are $\{h_i\}$ and $\{g_i\}$, the synthesis filter coefficients are $\{\tilde{h}_i\}$ and $\{\tilde{g}_i\}$, and downsampling and upsampling by a factor of 2 are denoted by $\downarrow 2$ and $\uparrow 2$ respectively.

a linear transform. Given a blocked orthogonal transform, an equivalent subband decomposition may be found which differs only in the organisation of the data, with spatially related coefficients grouped together in blocks for blocked linear transforms, and coefficients related in frequency grouped together in subbands for a subband decomposition [4, pp. 4-5] [158, pg. 181] [197, pp. 401-402].

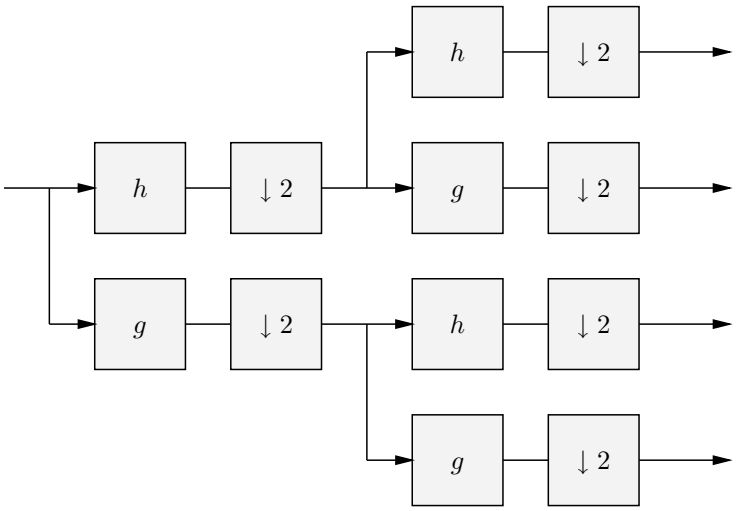


Figure 2.18: The analysis part of a uniform tree-structured filter bank.

Tree-structured filter banks [3, pp. 127-134] [197, pp. 142-156], which are constructed by cascading two-channel (or higher) filter banks, provide a convenient means of designing multichannel filter banks. A balanced tree corresponds to a uniform tree-structured filter bank (see Figure 2.18), while a maximally unbalanced tree corresponds to an octave-band tree-structured filter bank (see Figure 2.19). Multiresolution analysis (see Appendix C) presents an interpretation of a discrete wavelet decomposition as an octave-band subband decompositions [189] [197], the detail coefficients at a particular level of the tree of wavelet coefficients corresponding to the output of a single channel of a subband decomposition.

The simplest construction of two-dimensional filter banks [4, pp. 18-23] [197, pp. 176-184], for application to images, involves separate application of one-dimensional filter banks to image rows and the columns; the resulting two-dimensional filter banks are known as

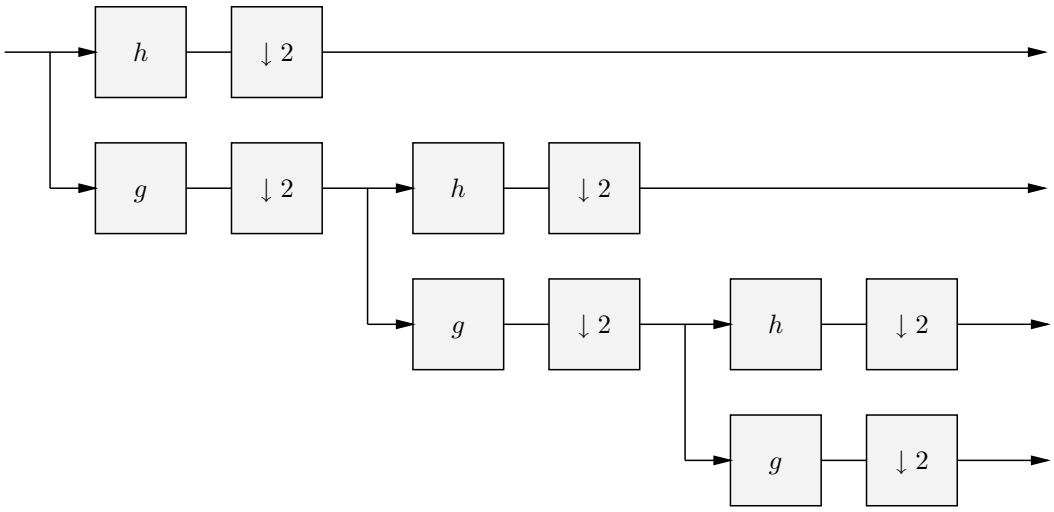


Figure 2.19: The analysis part of an octave-band tree-structured filter bank.

separable filter banks. The construction of non-separable filter banks is a considerably more complicated task.

Subband coding is, as the name suggest, the coding of a subband decomposition. The motivation for subband coding is the possibility of separately adapting the coding procedure to the statistics of each subband; in the simplest system different scalar quantisation is applied to each subband [99, ch. 11]. More complex coding techniques include VQ [43] of vectors taken from *within* each subband [7], where a separate codebook may be designed for each subband, or vectors taken *across* each subband [200].

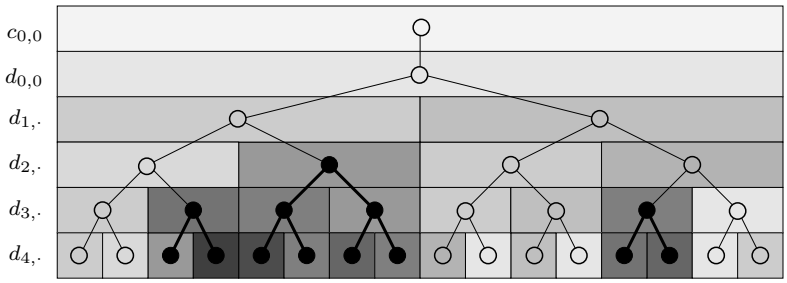


Figure 2.20: Zerotrees in a multiresolution decomposition. The intensity (increasing from dark to light) of each block is proportional to coefficient magnitude, and the filled circles indicate zerotree subtrees.

An alternative to inter- or intra-band coding is to consider the combined spatial and frequency localisation provided by the tree structure of coefficients in an octave-band decomposition such as a discrete wavelet decomposition. It is reasonable, under very general assumptions about the image model (e.g. decaying spectrum), to expect the magnitude of a wavelet coefficient to be less than that of its parent [182]. A simple coding method taking advantage of this fact recursively codes the wavelet coefficient tree of an image by represent-

ing an entire subtree with root coefficient magnitude less than some threshold as a *zerotree* (see Figure 2.20), scalar quantising the rest of the coefficients [114]. A more complex scheme, the Embedded[32] Zerotree Wavelet (EZW) algorithm [181] [182] [197, pp. 423-425] avoids the quantisation to zero of significant coefficients within a zerotree by an iterative coding process. A significance map indicates the position of detail coefficients of magnitude greater than a significance threshold, the threshold being halved at each iteration. The resulting code is conceptually similar to coding the detail coefficients in bit planes, with zerotrees coding the significance maps in each bit plane. This algorithm is highly effective[33], and its performance is often used as a benchmark in the recent literature.

2.5.4 Vector quantisation

Vector quantisation is used in image compression both as the primary coding mechanism and as secondary quantisation following some other method such as transform or subband coding. In its primary form the simplest scheme involves tiling the image by subblocks (e.g. 4×4 pixels), each of which is considered a separate vector. A codebook is constructed based on a large number of these vectors extracted from a training set of images, and the encoding consists of representing each block by the index of the closest codebook vector.

Since rate distortion theory implies that VQ becomes more efficient with increased vector size, the individual image subblocks should be as large as possible. The practical block size is severely limited however, by the rapid increase in codebooks size with increasing block size. The large codebook required for coding at a reasonably high bit rate involves considerable computation effort in the search for the nearest code vector for each image vector. Tree Structured VQ (TSVQ), which requires an enlarged codebook, achieves a significant improvement in search time (at the expense of selecting a sub-optimum codebook vector [73, pp. 410-423]) by imposing a tree structure on the codebook. Efficient nearest neighbour search techniques reduce the encoding search time while giving optimum or nearly optimum distortion [8].

Lattice VQ (LVQ) [73, ch. 10, 12] deals with the difficulties associated with large vectors by designing a codebook based on a lattice of vectors arranged according to some regular structure. The regular spacing of the vectors removes the necessity of storing a codebook, and introduces the possibility of finding the nearest codebook vector without the usual codebook search. Efficient quantisation of large vectors with a variety of distributions is possible using Lattice VQ techniques [61].

A practical solution to the unmanageable complexity of large codebooks is *product code* VQ [73, pp. 430-451]. A vector $\mathbf{x} \in X$ from a vector space X is decomposed into component

[32] An embedded code is a code for which each code string contains all lower rate representations of the same signal as a prefix.

[33] The importance of the quantisation strategy in image compression is indicated by the construction of a DCT based compression algorithm which achieves similar performance to the EZW algorithm by employing a similar quantisation strategy [207].

vectors $\mathbf{c}_i$ where $i \in \{1, 2, \ldots, p\}$ such that $\mathbf{x}$ may be recovered from $(\mathbf{c}_1, \mathbf{c}_2 \ldots, \mathbf{c}_p)$. The codebook is then composed of the Cartesian product $C_1 \times C_2 \times \cdots \times C_p$ of the individual codebooks C_i, resulting in a considerable reduction in storage requirements for the codebook C, as well as a reduction in computational complexity if each "sub-codebook" C_i is searched independently. Product codes often involve removing the mean or scaling by the standard deviation of each vector (Mean-Removed VQ and Shape-Gain VQ respectively). Murakami, Asai, and Yamazaki [139] subtracted the mean from 4×4 vectors and normalised by the standard deviation. The product code was then composed of separate codebooks for the mean, standard deviation and the normalised vectors.

There is usually considerable statistical dependence between image subblocks, resulting in blocking artifacts and a reduction in coding efficiency for simple blocked VQ treating each block independently. This deficiency may be alleviated by arranging the codebook used for each block to depend on its neighbouring blocks in a modification known as Finite-State VQ [73, ch. 14] [105].

2.5.5 Fractal compression

The fundamental principle of fractal coding is the representation of a signal by the parameters of a transform under which the signal is approximately invariant. This transform is constructed so that it is *contractive* (see Appendix D); Banach's fixed point theorem (see Appendix D) guarantees that an approximation to the original signal, called the fixed point of the transform, may be recovered by iterated application of the transform to an arbitrary initial signal. Although a more accurate description would be "fixed point coding"[34] this form of coding is termed "fractal" since the iterative decoding process creates detail at finer scales on each iteration, with the result that the fixed point signal is, in principle at least, a fractal [65, ch. 1].

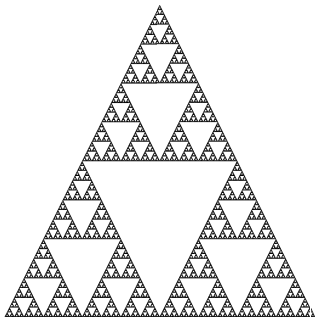


Figure 2.21: The Sierpinski Gasket.

The origins of fractal coding may be traced to Barnsley's work with Iterated Function Systems (IFS) for image modelling. An IFS is, to simplify somewhat, a collection of con-

[34] "Attractor coding" has been proposed, but has not been widely used.

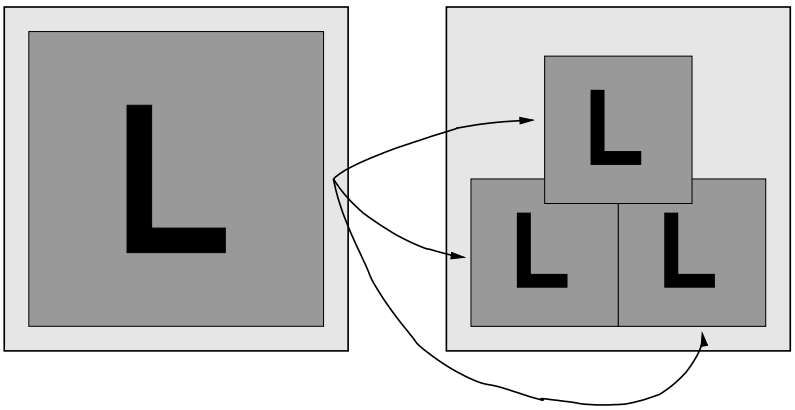


Figure 2.22: IFS for the Sierpinski Gasket [152, pg. 294].

traction mappings which are all applied to the same objects in a metric space (see Appendix B). The collection of mappings taken together constitute a "super-mapping", which being contractive, has a unique fixed point. In the simplest examples, a binary image is represented by the set of all pixel coordinates of non-zero pixels, and the individual mappings are affine[35] mappings in the Euclidean plane. The Sierpinski Gasket [152] shown in Figure 2.21 is the fixed point of an IFS consisting of the three mappings illustrated in Figure 2.22, illustrating the composition of the image as a "collage" of transformed versions of itself. The Sierpinski Gasket is in fact a fractal, since every iteration of its generating IFS adds smaller triangles as a finer scale; the image in Figure 2.21 is only a finite-resolution approximation to the real Sierpinski Gasket.

An IFS generating a desired image may be found by "covering" sections of the image by transformed version of the entire image, resulting in a set of transforms which leave the image approximately invariant. The collage theorem [15, pp. 95, 102-103] (see Appendix D) implies that the fixed point of the IFS composed of these transforms will be close to the original image. Another well known example of an IFS-generated image is Barnsley's fern, displayed in Figure 2.23, which is the fixed point of an IFS consisting of four affine mappings [152, pg. 295].

The success of IFS modelling of natural images (e.g. Barnsley's fern) in conjunction with the compactness of the resulting image representation prompted Barnsley to investigate the use of IFSs for image coding. Despite claims of 10000:1 compression ratios [17], the decoded images in question are more appropriately described as the result of image modelling than image coding. In addition, all of the images were "coded" by a human operator assisted process, with no known automatic procedure for the "inverse problem".

Most current fractal coding schemes are based on representation by a Partitioned IFS (PIFS) [65], a solution to the inverse problem of which was first published by Jacquin [93],

[35] An affine transform on a vector $\mathbf{x}$ may be expressed as $A\mathbf{x} + \mathbf{b}$, where A is a linear transform and $\mathbf{b}$ is a vector in the same space as $\mathbf{x}$.

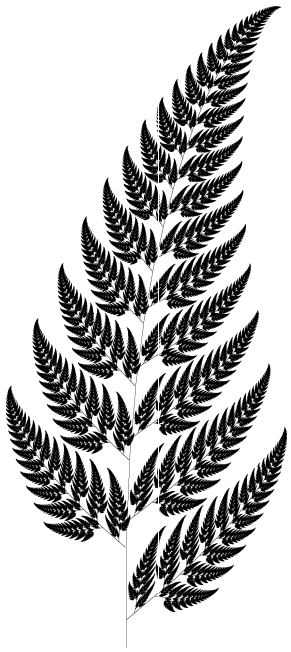


Figure 2.23: Barnsley's fern.

and subsequently patented by Barnsley. A PIFS differs from an IFS in that the individual mappings operate on a subset of the image, rather than the entire image. Instead of each iteration of the transform copying transformed version of the entire image to a new image, each transform operates only on a subregion of the image, commonly referred to as "domain blocks" due to their role in the mappings. The image subregions to which the domain blocks are mapped are called "range blocks" for similar reasons[36]. In coding of greyscale (as opposed to binary) images, the image is represented as function on the Euclidean plane, where the height of the surface at each point represents the local pixel intensity. In this representation a transform on a domain block may separately transform the block *support* and the block *intensities*[37].

The first step in a simple implementation is to tile the image (see Figure 2.24) by non-overlapping range blocks (e.g. 8×8) and larger (e.g. 16×16), possibly overlapping domain blocks. A set of admissible block transforms is defined, consisting of a contraction of the block support by a factor of two on each side by averaging neighbouring pixels (see Figure 2.26), followed by the application of one of the eight rotations and reflections (see Figure 2.27) making up the isometries of a square (see Figure 2.25), and finally an affine transform on the pixel intensities (see Figure 2.28).

The encoding phase (once again utilising the collage theorem) consist of finding for each

[36]The range/domain labels are reversed by Barnsley [16, pg. 181] for reasons which are too obscure to explain at this stage.

[37]For example, a rotation of a block is an operation on its support, whereas multiplying each of the pixels in the block by some value is an operation on its intensities.

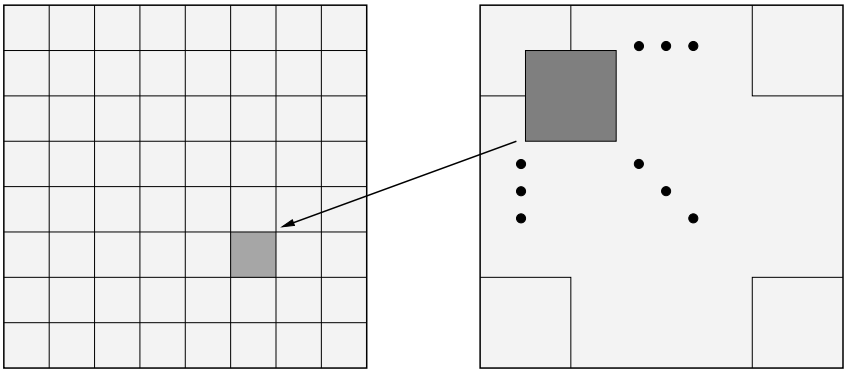


Figure 2.24: Domain and range blocks in PIFS coding.

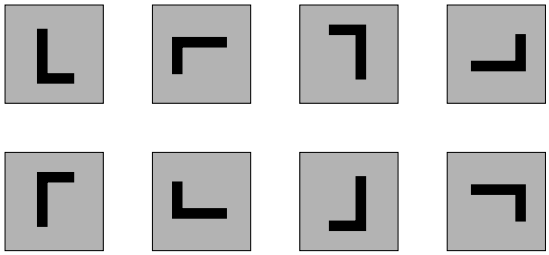


Figure 2.25: The square isometries.

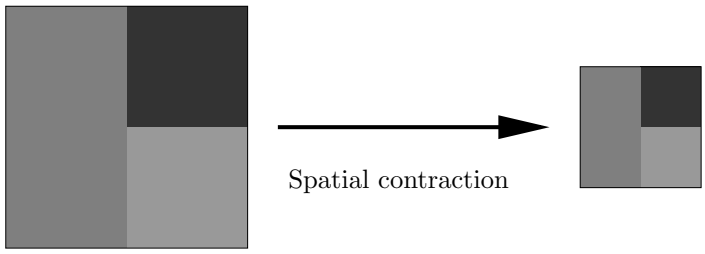


Figure 2.26: Spatial contraction of a domain block.

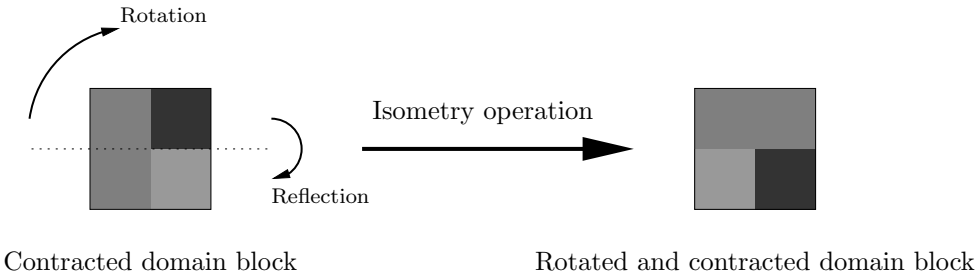


Figure 2.27: An isometry applied to a domain block.

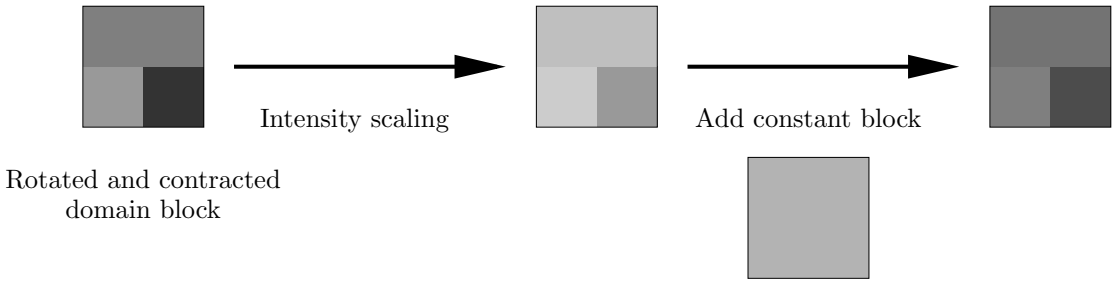


Figure 2.28: An affine transform applied to a domain block.

range block a domain block for which the pixel values can be made close to those of the range block by the application of an admissible transform. Care must be taken in selecting these transforms so that their union is a contractive transform on the image as a whole. The pool of domain blocks is often referred to as the self- or virtual codebook, since collage theorem based encoding is equivalent to Mean Removed Gain Shape VQ [139] encoding with a codebook consisting of domain blocks extracted from the image to be encoded. The distortion measured during VQ encoding, resulting from the errors in covering the image with codebook blocks, is the same as the actual distortion obtained on decoding. This is not the case for fractal coding, since any error in covering a particular range block modifies the domain blocks with which it intersects, which is not taken into account during the usual encoding process. The collage theorem nevertheless guarantees that the actual error on decoding may be made arbitrarily small, by making the collage error in covering each range by a transformed domain sufficiently small.

Once encoding is complete, the image is represented by a list containing the selected domain block and transform parameters for each range block. The image is decoded by iteratively transforming an arbitrary initial image using the transform consisting of the union of the transforms for each range block.

Fractal image compression is described in considerably greater detail in the following chapter, in which a broad overview of the fractal compression literature is presented.

Chapter 3

A Review of the Fractal Coding Literature

The fundamental principle of fractal coding, which consists of the representation of a signal by a transform of which it is the fixed point, clearly leaves considerable latitude in the design of a particular implementation. Within this broad framework, the differences between the majority of existing fractal coding schemes may be classified into the following categories:

- The partition imposed on the image by the range blocks.
- The composition of the pool of domain blocks, which is restricted to some extent by the range partition.
- The class of transforms applied to the domain blocks.
- The type of search used in locating suitable domain blocks.
- The quantisation of the transform parameters and any subsequent entropy coding.

There are unfortunately very few theoretical results on which design decisions in any of these aspects may be based, and choices are often made on a rather *ad hoc* basis. In addition, these categories are not independent, in the sense that any comparative analysis of coding performance between different options in one of these categories is usually contingent on the corresponding choices in the other categories. A comparison between the relative merits of particular choices in each category is consequently very difficult. This review is therefore intended primarily as an overview of the variety of schemes that have been investigated, although brief comparisons are made where possible. Since the research in this dissertation is restricted to greyscale images, a similar restriction is made in this chapter[1], and publications concerned with colour image or video coding are intentionally excluded.

[1] Unless otherwise stated results pertain to 8 bit/pixel greyscale images.

The survey of design choices is followed by a review of some of the more theoretical aspects of fractal compression, such as the collage theorem and convergence conditions. The chapter is concluded with a wavelet based analysis of fractal compression and a comparison of the performance of the most effective fractal coding based compression algorithms in the literature.

3.1 Partition Schemes

The first decision to be made when designing a fractal coding scheme is in the choice of the type of image partition used for the range blocks. Since domain blocks must be transformed to cover range blocks, this decision, together with the choice of block transformation described later, restricts the possible sizes and shapes of the domain blocks. A wide variety of partitions have been investigated, although the majority of coding systems are based on a square or rectangular partition.

3.1.1 Fixed square blocks

The simplest possible range partition consists of the fixed square blocks [16] [62] [64] [149] depicted in Figure 3.1. The most prominent example of a fractal compression system based on this partition was developed by Monro *et al.* [133] [134] [135].

This type of block partition is successful in transform coding of individual image blocks[2] since an adaptive quantisation mechanism is able to compensate for the varying "activity" levels of different blocks, allocating few bits to blocks with little detail and many to "active" blocks. Fractal coding based on the usual block transforms, in contrast, is not capable of such adaptation, representing a significant disadvantage of this type of block partition for fractal coding (this deficiency may, however, be addressed by introducing adaptivity to the available block transforms [18] [19] [20] as described in Section 3.2.2).

The usual solution is to introduce an adaptive partition with large blocks in low detail regions and small blocks where there is significant detail.

3.1.2 Quadtree

The quadtree [168, ch. 1] partition (see Figure 3.1) employs the well-known image processing technique based on a recursive splitting of selected image quadrants, enabling the resulting partition to be represented by a tree structure in which each non-terminal node has four descendents. The partition is constructed by selecting an initial level in the tree (corresponding to some maximum range block size) and recursively partitioning any block for which a match better than some preselected threshold is not found. Quadtree partitions are described in

[2] Such as implemented in the JPEG standard.

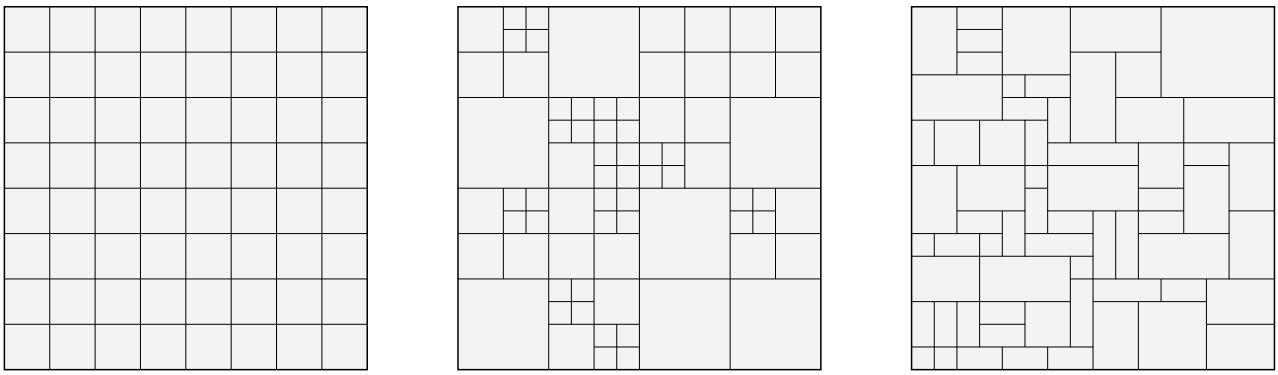


Figure 3.1: Rectangular range partition schemes.

considerable detail by Fisher [65, ch. 3] and Lu and Yew [121] [122]. Compact coding of partition details is possible by taking advantage of the tree structure of the partition.

Jacquin [92] [93] [94] used a variant of the quadtree partition in which the block splitting was restricted to two levels. Instead of automatically discarding the larger block prior to splitting it into four subblocks if an error threshold was exceeded, it was retained if additional transforms on up to two subblocks were sufficient to reduce the error below the threshold.

Reusens [160] implemented a quadtree scheme in which the range blocks were overlapped[3] in order to reduce blocking artifacts. A significant reduction in blocking artifacts was observed, but without corresponding improvement in SNR. This technique, while promising, has been overtaken to a large extent by developments in wavelet domain fractal coding, reviewed in Section 3.9.

3.1.3 Horizontal-vertical

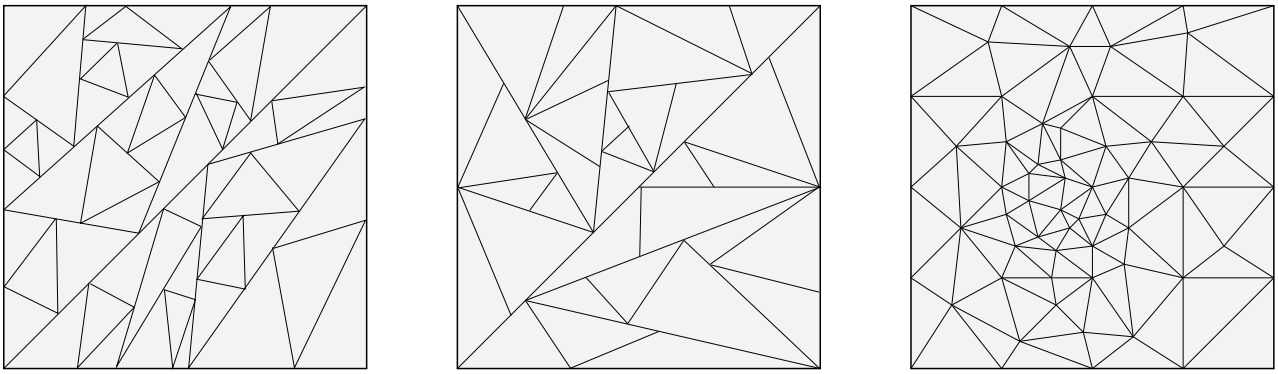
The Horizontal-Vertical (HV) partition [62] [64] [65, ch. 6] (see Figure 3.1), like the quadtree, produces a tree-structured partition of the image. Instead of recursively splitting quadrants, however, each image block is split into two by a horizontal or vertical line. A heuristic algorithm [65, pg. 120] is used to create horizontal or vertical partitions along prominent edges, but avoiding the creation of narrow rectangles. Compact coding of the partition details, similar to that utilised for the quadtree partition, is possible.

3.1.4 Triangular

There are a number of different ways of generating a triangular partition. Fisher [64] proposed a triangular partition generated by splitting the image into two triangles, followed by

[3]The areas of overlap may be considered to be generated by the weighted sum of multiple transformations, which may be made contractive.

a recursive splitting of triangles into four sub-triangles by inserting lines between split point on each side of the original triangle (see Figure 3.2). Novak [144] [145] employed a similar recursive scheme, but split each triangle into two by inserting a line from a vertex of the triangle to a point on the opposite side (see Figure 3.2).



3-side split 1-side split Delaunay triangulation

Figure 3.2: Triangular range partition schemes.

An alternative triangular partition is based on a *Delaunay triangulation* [157] of the image (see Figure 3.2), which is constructed on an initial set of "seed points", and is adapted to the image by adding extra seed points in regions of high image variance [48] [49] [50].

3.1.5 Polygonal

Reusens [161] constructed a polygonal partition (see Figure 3.3) by recursively subdividing an initial coarse[4] grid. Each polygon was allowed to be subdivided at an arbitrary position in the polygon by the insertion of a line segment at one of a restricted set of angles, thus reducing the information required in coding the partition details.

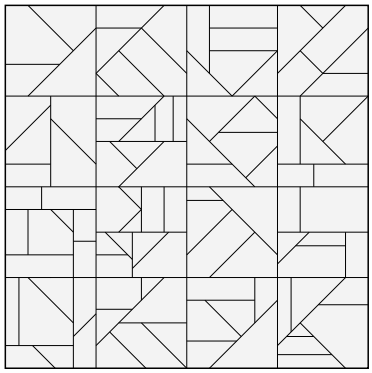


Figure 3.3: A polygonal range partition.

[4]Presumably rectangular, since details are not provided.

A generalisation of the Delaunay triangulation based partition [51] may be generated by merging suitable triangles to form quadrilaterals. An irregular partition based on a region-growing approach has also been investigated [176] [192].

3.1.6 Comparison

In designing an adaptive partition there is always a trade-off between the lower distortion expected by adapting the partition to the image content, and the additional bits required to specify the partition details.

Reusens compared rate distortion results obtained for the polygonal partition with those for the HV and quadtree partitions, concluding that the simplest partition (quadtree) provided the best results. This conclusion is not in agreement with the comparison [63] of the quadtree and HV partitions by Fisher, in which the HV partition was found to be superior. An irregular partition [192] was found to offer superior performance to a fixed square block partition, but was not compared with other adaptive partitions. Saupe and Ruhl found that a similar irregular partition outperformed a quadtree partition [176].

A significant disadvantage of the non-rectangular partitions is the additional computation involved in the block transformations (since there is often no pixel-to-pixel correspondence between domain and range blocks, interpolation is required in the block transforms). Comparisons based on published results are difficult, but the performance of non-rectangular partition based coders does not appear to justify the additional complexity. In particular, the best algorithms in the comparison of Section 3.10 are all based on rectangular partitions or their wavelet domain generalisations.

3.2 Block Transforms

The type of block transform selected is a critical element of a fractal coding scheme since not only does it determine the convergence properties on decoding, but its quantised parameters also comprise the majority of the information in the compressed representation.

A distinction is made here between transforms operating on the block support ("geometric" transforms in Jacquin's terminology[5] [94]) and those operating on the pixel values (termed "massic" transforms by Jacquin).

3.2.1 Block support

The permissible transforms on the block support are restricted by the block partition scheme, since domain block supports are required to be mapped onto range block supports.

[5]The block isometries are considered to be block support transforms here, in contrast to Jacquin's usage.

Rectangular blocks

The block support transform for rectangular blocks may be separated into an initial spatial contraction followed by one of the square isometry operations.

The spatial contraction of domains as introduced by Jacquin [93] is almost universally applied, despite being inessential for the contractivity of the image map as a whole [24] [62]. While contraction by a factor of two in width and height is standard, increasing this to a factor of three has been found to improve decoder convergence [23]. Contraction is usually achieved by the averaging of neighbouring pixels, which may be improved by the addition of an anti-aliasing filter [20]. The alternative of decimating by discarding pixels in both directions [123, pg. 141] is slightly faster, but results are inferior to those obtained by averaging [62].

The symmetry operations utilised by Jacquin are widely used as a means of enlarging the domain pool. Although Jacquin found some of the isometry operations were used more than others, other researchers found these operations were all used to a similar extent [69] [138]. These conflicting results are possibly due to the sensitivity to design choices in each of the categories introduced at the beginning of this chapter. Despite their widespread usage, there is evidence that their application is counter-productive in a rate distortion sense [138] [172] [203].

Triangular blocks

An affine mapping on the image support is sufficiently general to transform domain triangles to range triangles in a triangular partition. These affine transforms are determined by requiring that the transformed vertices of the domain blocks match those of the range blocks. The affine mappings required in this case are considerably more computationally expensive than those applied to square blocks, since interpolation of the image support is required.

Polygonal blocks

Depending on their structure, polygonal blocks may require transforms more general than affine in transforming domain to range blocks [51].

3.2.2 Block intensity

The simplest[6] intensity transform in common use is that introduced by Jacquin

$$T\mathbf{u} = s\mathbf{u} + o\mathbf{1}, \tag{3.1}$$

where s and o are variable scaling and offset coefficients and $\mathbf{1}$ is a vector of unit components.

[6] An even simpler transform, with a *fixed* scaling coefficient s is utilised by Barnsley in a tutorial example [16, ch. 6].

Modified affine transform

Barthel and Voyé [20] achieved improved decoder convergence using the transform[7]

$$T\mathbf{u} = s\mathbf{u} + (\alpha - s)\frac{\langle \mathbf{u}, \mathbf{1} \rangle}{\|\mathbf{1}\|^2}\mathbf{1} + o\mathbf{1} \quad \alpha \in [0, 1]$$

which avoids full scaling of the range block mean. Minimum transform coefficient variance was obtained for $\alpha = 1$, whereas optimum decoder convergence was achieved for $\alpha = 0$, and a compromise of $\alpha = 0.5$ provided the best results.

Øien *et al.* [147] [148] proposed removing the DC component of the domain block prior to scaling

$$T\mathbf{u} = s\left(\mathbf{u} - \frac{\langle \mathbf{u}, \mathbf{1} \rangle}{\|\mathbf{1}\|^2}\mathbf{1}\right) + o\mathbf{1},$$

corresponding to the transform above with $\alpha = 0$. The removal of the DC part of the vectors creates transformed domains which are orthogonal to the fixed block $\mathbf{1}$, with the desirable effect of decorrelating the s and o coefficients. In addition, given a few other restrictions on the coder parameters, convergence at the decoder is guaranteed within a fixed number of iterations.

Frequency domain transforms

Barthel *et al.* [18] [19] [20] experimented with a block transform allowing selective manipulation of the block spectral contents[8]

$$T\mathbf{u} = A^{-1}\left[\begin{pmatrix} s_0 & 0 & 0 & \dots \\ 0 & s_1 & 0 & \dots \\ 0 & 0 & s_2 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} A\mathbf{u} + \begin{pmatrix} o_0 \\ o_1 \\ o_2 \\ \vdots \end{pmatrix}\right]$$

where A is the DCT matrix. Adaptivity to block detail levels was achieved by varying the number of s_i and o_i that were individually specified. This hybrid scheme constitutes a transition between conventional fractal coding and transform coding, with the exact nature depending on whether the majority of information is represented by the s_i or o_i respectively.

Although discussion of the recent wavelet domain fractal coders is also appropriate here,

[7] A better choice would probably have been

$$T\mathbf{u} = s\left(\mathbf{u} - \alpha\frac{\langle \mathbf{u}, \mathbf{1} \rangle}{\|\mathbf{1}\|^2}\mathbf{1}\right) + o\mathbf{1},$$

which may be continuously varied between the original transform of Jacquin and the orthogonalised version of Øien *et al.* by varying α between 0 and 1.

[8] The s_i and o_i here should not be confused with the usual usage in which the subscript identifies the range to which the transform is assigned.

it is deferred until Section 3.9 for more detailed analysis.

Multiple fixed blocks

A more general transform is possible by utilising multiple fixed blocks $\mathbf{e}_i$

$$T\mathbf{u} = s\mathbf{u} + \sum_i o_i \mathbf{e}_i,$$

as opposed to the usual single fixed block $\mathbf{1}$ of unit entries. Orthogonalisation of the domain block term with respect to the fixed block term is possible by projecting the domain block perpendicular to the subspace spanned by the fixed domain blocks [148].

Monro and Dudbridge [134] [135] used three fixed blocks, the usual constant block and two with constant gradient in the x and y directions [199, pg. 202], referring to the transform as an "order 1 polynomial transform." In later work [133] [137] [138] this transform was extended to "order 2" by including constant blocks with quadratic form in x and y directions, and to "order 3" with the addition of cubic form blocks. In experiments with limited domain searching the "order 2" transform was found to be best in a rate distortion sense [204].

If all of the s_i are equal in the frequency domain transform of Barthel *et al.*, it becomes equivalent to the transform with multiple fixed blocks, with the DCT basis vectors as the fixed blocks. Although no explicit comparison has been made between the use of polynomial or DCT basis fixed blocks, in the absence of experimental evidence the DCT basis blocks are likely to be superior, since they are known to form an efficient basis for image blocks and, unlike the polynomial bases, are mutually orthogonal.

Multiple domains

Vines [199] utilised a transform consisting of a scaling of *multiple* domain blocks. Computational tractability was achieved by creating an orthogonal basis of the domain block set, representing each range by a scaling of as few basis vectors as possible. A variety of mappings using multiple fixed blocks as well as multiple domain blocks, including domain blocks with no spatial contractivity, was investigated by Gharavi-Alkhansari and Huang [74].

3.3 Domain Pool Selection

The domain pool used in fractal compression is often referred to as a virtual codebook, in comparison with the codebook of VQ [95]. It should be clear from this comparison that a suitable domain pool is crucial to efficient representation since, although increased fidelity may be obtained by allowing searching over a larger set of domains, there is a corresponding increase in the number of bits required to specify the selected domain.

A bound $|s| < s_{\max}$ is usually placed on the block intensity transform scaling coefficients in order to guarantee contractivity. The selection of the optimum domain for a particular range is influenced by this bound, since any domain for which the optimum scaling coefficient exceeds this bound is required to be rejected as a possible match.

Repeated references are made in this section to the *spatial* distances between domain and range blocks; it should be emphasised that this is the distance (measured in pixels) in the image support between the range and domain block centres, and *not* the distortion resulting from representing the range block by that particular domain block (the collage error for that range block).

3.3.1 Global codebook

The simplest domain pool design provides a fixed domain pool for all range blocks in the image, or for a particular class of range blocks distributed throughout the image (e.g. range blocks of one size in a quadtree partition). This design choice is motivated by experiments indicating that the best domain for a particular range is not expected to be spatially close to that range to any significant degree [62, pp. 56-57] [65, pp. 69-71] [69].

In the fixed square block or quadtree partitions domain blocks may be placed at intervals as small as one pixel. Since this results in an enormous domain pool which is slow to search[9], larger domain increments are usually selected, typically equal to the domain block width [62] [65, ch. 3] [149] or half the domain block width [31] [65, ch.3] [92] [146]. Improved convergence is also obtained with either of these increments [148]. The larger of the two corresponding domain pools was found to be superior in fidelity and compression ratio [62].

In adaptive partitions such as HV or triangular, the domain pool usually consists of the larger blocks in the range pool [62] or larger blocks created by the same partitioning mechanism [51].

3.3.2 Local codebook

A number of researchers have noticed a tendency for a range block to be spatially close to the matching domain block [20] [23], based on the observed tendency for distributions of spatial distances between range and matching domain blocks to be highly peaked at zero [91] [95] [204].

Motivated by this observation, the domain pool for each range block may be restricted to a region about the range block [92], or a spiral search path may be followed outwards from the range block position [20] [23]. More complicated alternatives include using a domain position mask centred at each range block, with positions in the mask dense near to the range and

[9] In addition, since domains selected at small increments are likely to be similar, it is difficult to justify the expenditure of extra bits to code their positions in a larger domain pool.

progressively less dense further away [91], and using widely spaced domain blocks together with a fine lattice in the vicinity of the best match in the coarse lattice [91].

The domain search may also be dispensed with entirely [134] [135], or restricted to a very small set about the range block [133]. Experimental evidence for the case of the "quadratic" transform suggests that any domain searching is counter-productive in a rate distortion sense [204].

3.3.3 Synthetic codebook

A significant variation on the usual domain pool of fractal coding is achieved by the creation of an initial low resolution image approximation from which the domain pool is extracted [103] [104]. Although these domain blocks are not likely to be as effective in matching range blocks as those derived in the usual fashion, decoding does not require iteration, and the coding error may be determined immediately at the encoder.

3.3.4 Comparison

The question of domain locality (the tendency for a range and matching domain to be spatially close) plays an important role in the design of an efficient domain pool. While the degree to which this effect is present my be dependent on the particular fractal coding scheme for which it is evaluated, this does not adequately explain the extent of the disagreement in the literature. This issue is explored in greater detail in subsequent chapters.

3.4 Search Strategies

The significant computational requirements of the domain search resulted in lengthy coding times for early fractal compression algorithms. The design of efficient domain search techniques has consequently been one the most active areas of research in fractal coding, resulting in a wide variety of solutions. The survey presented here is rather brief, since a detailed review of these techniques is available [174].

3.4.1 Invariant representation

The search for the best domain block for a particular range block is complicated by the requirement that the range matches a *transformed* version of a domain block; the problem is in fact to find for each range block, the domain block that can be *made* the closest by an admissible transform. Given a range block $\mathbf{r}$, a set of domain blocks $\mathbf{d}_i$ and the admissible transforms $T_{\mathbf{n}}$ parameterised by $\mathbf{n}$, the optimum domain block results in a collage error of

$$\min_{\mathbf{n},i} d(\mathbf{r}, T_{\mathbf{n}}\mathbf{d}_i),$$

where $d(\mathbf{u}, \mathbf{v})$ is an appropriate distance measure.

The problem is simplified by constructing, for each image block $\mathbf{u}$, a *relative distance preserving invariant representation* $\xi(\mathbf{u})$ with the properties:

Invariance: $\xi(T_{\mathbf{n}}(\mathbf{u})) = \xi(T_{\mathbf{m}}(\mathbf{u})) \ \ \forall \mathbf{n}, \mathbf{m}$
Order: $d(\xi(\mathbf{u}), \xi(\mathbf{v})) \leq d(\xi(\mathbf{u}), \xi(\mathbf{w})) \Rightarrow \min_{\mathbf{n}} d(\mathbf{u}, T_{\mathbf{n}}(\mathbf{v})) \leq \min_{\mathbf{n}} d(\mathbf{u}, T_{\mathbf{n}}(\mathbf{w}))$.

Transforming range and contracted domain blocks to this representation allows direct distance comparisons between them to determine the best possible match.

The "standard" invariant representation for the block intensity transform[10] projects each vector perpendicular to the space spanned by the fixed block terms, followed by normalisation; the invariant representation of $\mathbf{u}$ for transforms as in Equation (3.1) is

$$\xi(\mathbf{u}) = \frac{\|\mathbf{1}\|^2\mathbf{u} - \langle \mathbf{u}, \mathbf{1} \rangle \mathbf{1}}{\|\mathbf{1}\|^2\|\mathbf{u}\|^2 - \langle \mathbf{u}, \mathbf{1} \rangle^2}.$$

An alternative representation for the single constant block case utilises the DCT (or other orthogonal transform) of the vector, followed by zeroing of the DC term and normalisation. An advantage of the orthogonal transform representation is the possibility of utilising an HVS adapted distance measure [19] [23] [201].

Many of the features used in the classification techniques described below are also invariant under the transforms applied. The distinction between an invariant *representation* and an invariant *feature* is that there exists a transform making two blocks with the same invariant representation exactly equal, whereas such a transform is not guaranteed to exist for two blocks with the same invariant feature vector; they are merely likely to be similar.

3.4.2 Domain pool reduction

One of the simplest ways of decreasing coding time is to decrease the size of the domain pool in order to decrease the number of domains to be searched, which is often achieved by a spatial constraint on the domain pool for each range, as described in Section 3.3.2. Noting that a contractive transform requires a domain with a higher magnitude invariant representation than the range to which it is mapped, domains with low magnitude invariant representation may be excluded from the domain pool [173]. Alternatively, the domain pool may be pruned in order to exclude domains which have similar invariant representations [184] to other domains in the pool. The classification and clustering schemes discussed below also fit, to a certain extent, within the framework of domain pool reduction.

[10] A relative distance preserving invariant representation with respect to the block isometries is not possible.

3.4.3 Classification

Classification based search techniques often do not explicitly utilise an invariant representation as formalised above, but rely instead on features which are at least approximately invariant to the transforms applied. These techniques may be separated into classification based on non-metric and metric features[11].

Non-metric features

The algorithm introduced by Jacquin [92] [93] employed a block classification scheme designed for VQ [159] in reducing the domain search. Domain and range blocks were separately classified as "Shade", "Midrange" or "Edge" blocks, and a matching domain for each range was only sought within the same class. Fisher designed a classification into 72 classes [65, ch. 3] based on relative averages[12] of the quadrants of each block.

Metric features

Frigaard *et al.* [69] computed features in a space with a metric, searching blocks with features within some distance threshold of the range block features. The features utilised were the block standard deviation[13] and the number of "dominant grey levels" which is the number of distinct pixel values for which the number of pixels with that value exceed some threshold. Novak [144] classified blocks according to a set of invariant features based on the moments of the block pixel values in a triangular partition.

An alternative set of features may be defined by calculating inner products with a fixed set of vectors [24]. These inner products provide lower bounds on distances between domain and range blocks, allowing many of the domains to be excluded from the actual distance calculation.

Götting and Ibenthal [76] transformed the standard invariant representation into a set of features which were also invariant to the block isometries, arranging them in a tree structure to speed up the search. A tree search has also been applied to a pyramid of progressively coarser resolution domains, with the search progressing at each level in the region of the best match in the previous level [101] [119].

3.4.4 Clustering

Lepsøy [112] [113] reduced the search effort by identifying clusters of domain blocks in the domain pool, the cluster centres being located by applying the GLA to the domain blocks using a distortion measure based on an invariant representation equivalent to the cosines of

[11] Often respectively called discrete and continuous features, but the important distinction is really whether a useful metric exists in the feature space.

[12] It is interesting to note that the distribution of these feature as observed by Hürtgen and Stiller [91, pg. 405] confirms the importance of horizontal and vertical edges in images.

[13] The utility of this feature is unclear, since it is anything but invariant to the transform applied.

the angles between vectors. The optimum domain for each range was located by a comparison of the range with each cluster centre, followed by a comparison with all cluster members of the best cluster. A similar clustering approach, but using the standard invariant representation was evaluated by Hamzaoui [79]. A clustering approach based on the GLA has also been applied to domain blocks in a triangular partition [48].

Boss and Jacobs [31] avoided the computational cost of clustering during encoding by designing the clusters on an initial training set rather than determining them adaptively for each image.

3.4.5 Convolution

The collage error between a domain and range block[14] may be expressed in terms of the magnitudes of the domain and range blocks, the inner products of domain and range blocks with a constant block, and the inner product of the domain and range blocks [175]. All except the last of these values are used repeatedly in finding a best match for each range block; the computational cost of the domain search is consequently dominated by the calculation of the inner products between domain and range blocks. These inner product calculations may be efficiently performed in the frequency domain by considering the calculation of the inner products between a particular range and all domains as a convolution of the image with that range block [175]. A comparison with the computational required for the usual inner product calculations revealed a considerable advantage for the frequency domain technique when applied to range blocks larger than 4×4 in size.

3.4.6 Nearest neighbour search

The nearest neighbour search techniques described in this section are all designed for an efficient search for nearest neighbours (subject to a predetermined distance measure) in a high dimensional space. The set of vectors from which neighbours are selected is referred to as the set of records (or file), and the vector for which neighbours are sought is the query record. Application of these techniques to fractal coding is achieved by using the invariant representations of the domains as records, and the invariant representation of each range as a query record.

A simple technique for decreasing search time is the *partial distance* [73, pp. 479-480] method used in VQ, which in conjunction with a running minimum, compares the query record with each record, terminating the comparison for a particular record if the partial sum of initial record elements exceeds the running minimum. Beaumont [23] constructed an invariant representation using the Hadamard transform, applying the zig-zag scan order to create an ordered list of coefficients. The transform improved the efficiency of the partial

[14] In this context the collage error is the distance between the range block and optimally transformed domain block.

distance search since its energy packing property shifted most of the variance to the initial elements of the vector.

More complex nearest neighbour search algorithms utilise a preprocessing stage to arrange the records in a data structure, usually a tree, enabling efficient access. The records are placed in the tree so as to minimise the number of nodes to be "visited" during the search.

Saupe [169] [170] [171] used the standard invariant representation[15] in a k-d tree search [68] [168, ch. 2]. A faster *approximate* nearest neighbour search algorithm [8] [9] was also tested, resulting in a considerable decrease in search time together with a small decrease in image fidelity. The same invariant representation has been applied [106] in an R-tree [168, ch. 3] based search algorithm, which is expected to be less efficient than the k-d tree search [171, pg. 18]. Wohlberg and de Jager [201] found a number of advantages in utilising the DCT representation for the k-d tree search[16].

Bani-Eqbal [11] [12] arranged the standard invariant representations of domain blocks in a tree structure similar to the k-d tree. A fast search was effected by applying distance lower bounds at each node of the tree, allowing branches of the tree to be excluded from the search process.

3.4.7 Comparison

An objective comparison of the various rapid search techniques is difficult, as results are strongly dependent on the images used (including the sizes thereof) as well as the other parameters of the coding scheme. As a further complicating factor, classification methods may be used *in conjunction* with nearest neighbour methods to achieve further improvements in search efficiency [174, pp. 15-16]. Some tentative conclusions may however, be drawn from the survey by Saupe and Hamzaoui [174], in that the fast nearest neighbour method introduced by Saupe appears to have an edge over classification methods, especially for large images and large domain pools, while there are problems in utilising the adaptive clustering method for a quadtree codebook [174].

3.5 Quantisation

Domain positions, and any additional partition information required in an adaptive partition, are represented by discrete values and are not subjected to quantisation. There are usually compact methods of representing the range partition details in adaptive partitions such as quadtree or HV [65, ch. 3, 6]. Efficient representation of the domain positions is achieved by

[15] If low resolution versions of image blocks are used as records to conserve memory, the *representation* is really a metric *feature* vector.

[16] It was discovered in additional unpublished research that the reduction in search time due to the energy packing property of the DCT did not justify the additional time required for DCT computation when a sufficiently loose error threshold was applied for the *approximate* nearest neighbour search.

techniques such as the spiral search described in Section 3.3.2.

Although the distributions for the scaling and offset coefficients have been observed to be non-uniform, quantisation is usually uniform [65, ch. 3] [149] (with the possibility of compensation for inefficiency by subsequent entropy coding). Bit allocations for the scaling and offset coefficients have been respectively 2 and 6 [95], 5 and 8 [146], and between 2 and 4 for the scaling and between 3 and 8 for the offset [88]. Fisher compared performances for a number of bit allocations [65, pp. 61-65], observing the best performance for an allocation of 5 and 7 bits to the scaling and offset coefficients respectively.

Since both scaling and offset are usually non-uniformly distributed, with a peak or peaks around the origin, non-uniform quantisation is indicated. Øien investigated pdf optimised quantisation for the scaling coefficients [146], but no comparison with uniform quantisation was performed[17]. Signes [184] suggests that finer quantisation is required for the higher values of the scaling coefficients, which is somewhat at odds with the requirements for a pdf optimised quantiser for the distribution observed by Øien [146] and others.

It has been observed that the usual block transform (without DC removal[18]) results in correlated scaling and offset coefficients [21] [88]. Alternative responses to this observation have been VQ of combined scaling and offset coefficients [21], and linear prediction of the offset from the scaling [88]. Since there is usually also some correlation between the offset coefficients for neighbouring blocks, some form of predictive coding is indicated, but presents practical difficulties for some range partitions [146].

Quantisation optimisation for polynomial fixed block transforms is discussed by Monro and Woolley [137] [138], and VQ of the transform coefficients for the frequency domain transform is referred to by Barthel and Voyé [20].

A global bit allocation optimisation was implemented by Barthel *et al.* [19] for the frequency domain block transform. All range blocks were initially coded with the simplest block transform, requiring the least bits to specify, and a subsequent iterative procedure allocated additional bits to selected range blocks, allowing a fidelity gain by using the more adaptable block transforms. At each stage the additional bits were allocated to the range for which they resulted in the maximum fidelity gain.

Since the scaling coefficients are often rather coarsely quantised there is a significant advantage in calculating collage errors for each domain block using quantised transform coefficients [62, pg. 45] [146]. This may however, be difficult to achieve for some of the fast domain search methods [171, pg. 17].

[17] Fisher [65, pg. 63] found no improvement over uniform quantisation for logarithmic quantisation of the scaling coefficients.

[18] The same transform with DC removal does not result in a significant correlation between these coefficients [146].

3.6 Reconstruction

Reconstruction of the encoded image is achieved by computing the fixed point of the image transform T from its encoded coefficients. Since the encoded representation of a transform may be independent of the size of the encoded image, a form of interpolation is possible by reconstructing the fixed point at a higher resolution than the encoded image.

3.6.1 Decoding

Reconstruction of the fractal coded approximation of a signal is usually based on Banach's fixed point theorem (see Appendix D) which guarantees that the sequence constructed by the iterative application of a contractive transform to an arbitrary initial element of a complete metric space converges to the fixed point of that transform. Denoting the arbitrary initial element of the sequence by $\mathbf{x}_0$, the elements of the sequence are

$$\mathbf{x}_n = T^n \mathbf{x}_0$$

and the fixed point $\mathbf{x}_T$ is

$$\mathbf{x}_T = \lim_{n\to\infty} T^n \mathbf{x}_0.$$

The majority of existing fractal coding schemes restrict T to be an affine transform

$$T\mathbf{x} = A\mathbf{x} + \mathbf{b},$$

where A is a linear transform and $\mathbf{b}$ is an offset vector, in which case the elements of the decoding sequence become

$$\mathbf{x}_n = A^n \mathbf{x}_0 + \sum_{i=0}^{n-1} A^i \mathbf{b}.$$

Reconstruction of the fixed point by matrix inversion

$$\mathbf{x}_T = (I - A)^{-1}\mathbf{b},$$

is possible if $|I - A| \neq 0$; equivalent to requiring that A has no eigenvalues equal to 1. If the spectral radius[19] $r(A) < 1$, a Taylor series expansion [53, pp. 171-172] [195, pp. 82-83] yields

$$(I - A)^{-1} = I + A + A^2 + A^3 + \ldots$$

giving

$$\mathbf{x}_T = \mathbf{b} + A\mathbf{b} + A^2\mathbf{b} + \ldots \tag{3.2}$$

Imposing a few restrictions allows the fixed point to be expressed exactly by a finite number

[19] The spectral radius $r(A)$ of A is the maximum absolute value of the eigenvalues of A.

of the terms above [148].

Postprocessing in the form of smoothing along block boundaries has been found to be beneficial in reducing blocking artifacts [65, pg. 59].

3.6.2 Resolution independence

"Resolution independence" has been cited in the popular technical press as one of the main advantages of fractal compression [6] [54], but image interpolation by fractal decoding has not been subjected to a detailed comparison with classical interpolation techniques (although it has been discussed briefly [65] [76]). Subsampling an image to a reduced size, fractal encoding it, and decoding at a larger size has been found to produce results comparable to fractal coding of the original image [121], although there is no indication that replacing the fractal interpolation stage by another form of interpolation would not produce comparable results.

3.7 The Collage Theorem

Fractal coding is achieved by representing a signal $\mathbf{x}$ by a quantised representation of a contractive transform T which is chosen such that $T\mathbf{x} \approx \mathbf{x}$. Although an approximation to $\mathbf{x}$ may be recovered from T by the iterative process described previously, there is usually no simple expression for the fixed point $\mathbf{x}_T$ of T in terms of its quantised coefficients. As a result, and given the constraints on T imposed by its dependence on its constituent coefficients, it is not usually possible to optimise those coefficients to make the fixed point as close as possible to a given signal $\mathbf{x}$.

Since the the distortion $d(\mathbf{x}, \mathbf{x}_T)$ introduced by the fractal approximation can usually not be directly optimised for these reasons, the standard approach is to optimise T to minimise the collage error $d(\mathbf{x}, T\mathbf{x})$, which is usually computationally tractable. The collage theorem guarantees that $d(\mathbf{x}, \mathbf{x}_T)$ may be made small by finding T such that $d(\mathbf{x}, T\mathbf{x})$ is sufficiently small[20].

3.7.1 Contractive transforms

The simplest form of the collage theorem states that for a contractive transform T with Lipschitz factor α (i.e. $d(T\mathbf{x}, T\mathbf{y}) \leq \alpha d(\mathbf{x}, \mathbf{y})$) and with fixed point $\mathbf{x}_T$,

$$d(\mathbf{x}, \mathbf{x}_T) \leq \frac{d(\mathbf{x}, T\mathbf{x})}{1 - \alpha}.$$

[20]It is important to note that the collage error $d(\mathbf{x}, \mathbf{x}_T)$ is usually smaller than the actual distortion [47], whereas the various forms of the collage theorem provide an upper bound *in terms of* the collage error.

In image coding terms this implies that a transform T, for which the fixed point $\mathbf{x}_T$ is close to an original image $\mathbf{x}$, may be found by designing the transform T such that $T\mathbf{x}$ (the "collage") is close to $\mathbf{x}$.

3.7.2 Eventually contractive transforms

Convergence to a fixed point is also guaranteed by the less restrictive condition of *eventual contractivity*. In this case, T (with Lipschitz factor α) is not contractive, but some power k of T (with Lipschitz factor β) is. The collage theorem may then be stated as [65, ch.2]

$$d(\mathbf{x}, \mathbf{x}_T) \leq \frac{1-\alpha^k}{1-\alpha} \frac{d(\mathbf{x}, T\mathbf{x})}{1-\beta}.$$

A tighter collage bound is possible by imposing certain restrictions, consisting primarily of requiring DC removal in the block transform and setting the domain increment to be equal to the range block size [147] [148]. Despite the considerable improvement over the usual collage theorem bound, this bound is still rather pessimistic [147].

3.7.3 Optimal encoding

Although the collage theorem currently forms the basis of virtually all fractal coders, it does not result in an optimal image representation given the constraints imposed on the transform. Suboptimality is, amongst others, a result of optimisation of individual block transforms with respect to the domains in the original image, whereas only the fixed point domains are available during decoding. Collage based encoding may however be shown to be optimal under certain restrictions [112] [148].

Updating the scaling and offset coefficients after coding, by re-optimising them with respect to domains extracted from the coded image, was found to result in reduced distortion on reconstruction [20]. Hürtgen [85] found that optimising the transform subsequent to collage coding resulted in an SNR improvement of approximately 1.5dB for test images, which is in agreement with results of Domaszewicz and Vaishampayan [56].

3.8 Convergence

One of the most problematic areas of fractal coding is the convergence of the image transform. Convergence may be guaranteed by setting $s_{\max} = 1$; while this choice of the scaling coefficient bound results in more rapid convergence on decoding [65, pg. 62], it has also been shown to degrade image quality [62]. This restriction is sufficient, but not necessary for convergence, and although appropriate contractivity criteria are known, their computation during coding is not feasible, posing a significant problem for a practical encoder implementation, since sufficient convergence at the decoder should be guaranteed during encoding.

3.8.1 Orthogonalisation

The introduction of an orthogonalisation operator to each domain block, making it orthogonal to the constant blocks, results in a transform which may be shown to converge exactly within a fixed number of iterations (given additional constraints on the domain block spacing) [148].

3.8.2 Mapping cycles

The interdependence between ranges at one iteration of decoding and domains at the next may be analysed in terms of "mapping cycles", each of which consists of an independent set of domain to range mappings [86] [107]. The full image transform is convergent if each of its independent cycles is convergent.

3.8.3 Transform eigenvalues

Hürtgen [84] [86] [87] [90] showed that when the image transform is affine, a necessary and sufficient condition for convergence of the transform sequence on decoding is that the spectral radius of the linear part be less than unity (equivalent to eventual contractivity). The spectral radius was determined in terms of the transform parameters for a few simple cases, allowing analytic determination of convergence requirements on the transform coefficients for these cases. Since the computation of the spectral radius for the general case is difficult, a statistical distribution for the eigenvalues, based on a probability distribution for the transform parameters, allowed the determination of the probability of contractivity based on these distributions [89].

3.9 Wavelet Analysis

The most significant recent development in fractal coding theory is undoubtedly the independent discovery by a number of researchers of a multiresolution analysis (see Appendix C) description of certain classes of fractal coding. This discovery has not only resulted in improved fractal coders, but a better understanding of the mechanism underlying standard fractal coding.

3.9.1 Notation

The following notation is adopted to simplify the presentation of the mathematical results of this section. The range block width is denoted by b, the domain block width is assumed to be $2b$, and the number of domain and range blocks are N_d and N_r respectively. The domain increment (the number of pixels between successive domain blocks) is Δ_d, and the scaling and offset coefficients for range block i (where $0 \leq i < N_r$) are s_i and o_i respectively, while the

index of the domain block selected for range block i is denoted by p_i (where $0 \leq p_i < N_d$). Domain and range blocks are indexed from the start of the signal, with an initial index of 0.

3.9.2 Hierarchical decoding

One of the first applications of a multiresolution analysis view of fractal coding was in the construction of a hierarchical decoding method [13], intended to decrease the computational requirements of decoding the fractal representation of an image. Noting that a PIFS representation of a function is not explicitly linked to a particular vector size in a sampled representation, the notion of function f at resolution i was formalised, although no reference was made to the correspondence between this construction and the scaling function coefficients in an unnormalised Haar basis.

A PIFS may be decoded iteratively at a vector size corresponding to $b = 1$ if $\Delta_d = b$. Since the projection operator P_j (see Appendix C) in the Haar basis is equivalent to the usual averaging spatial contraction operator, functions in the approximation space V_j contain the *pre-contracted* domains of functions in V_{j+1}, allowing functions at progressively higher resolutions to be built up hierarchically from the initial vector [14].

The description presented here is restricted to one-dimensional signals for notational simplicity, but the extension to two dimensions follows easily. At resolution i the affine transform may be represented in terms of the Haar basis scaling function coefficients $c_{i,j}$ as[21]

$$c_{i,j} = s_{\lfloor j/b \rfloor} \frac{1}{2} \left(c_{i,p_{\lfloor j/b \rfloor} b + 2(j \bmod b)} + c_{i,p_{\lfloor j/b \rfloor} b + 2(j \bmod b)+1} \right) + o_{\lfloor j/b \rfloor}.$$

Since for the Haar basis

$$c_{i,j} = \frac{1}{\sqrt{2}} \left(c_{i+1,2j} + c_{i+1,2j+1} \right),$$

$c_{i,j}$ may be written as

$$c_{i,j} = s_{\lfloor j/b \rfloor} \frac{\sqrt{2}}{2} c_{i-1,p_{\lfloor j/b \rfloor} b/2 + (j \bmod b)} + o_{\lfloor j/b \rfloor}, \tag{3.3}$$

which describes the hierarchical decoding process. The entire decoding process is detailed in Table 3.1.

A similar algorithm for hierarchical decoding of images coded with multiple fixed blocks is described by Monro and Dudbridge [136].

3.9.3 Fractal coding as wavelet detail extrapolation

The possibility of simple wavelet domain interpretation of a class of fractal coding schemes has been described independently by a number of authors [47] [109] [185] [193], each describing

[21] If $m, n \in \mathbb{Z}$, then $m \bmod n$ is the remainder on dividing m by n. The floor $\lfloor x \rfloor$ of $x \in \mathbb{R}$ is the largest integer less than or equal to x.

Table 3.1: Hierarchical Fractal Decoding

Start with: Range partition of $N_r = 2^\alpha$ blocks
Domain partition with $\Delta_d = b$
Signal is to be decoded to size $n = 2^\beta$
Set of N_r transform coefficients (o_i, s_i, p_i)

1. Iteratively apply PIFS to decode signal at resolution α, resulting in $c_{\alpha,0}, c_{\alpha,1}, \ldots, c_{\alpha,2^\alpha-1}$.

2. Iterate through resolutions $\alpha < i \leq \beta$
   Iterate through positions $0 \leq j < 2^i$

$$c_{i,j} = s_{\lfloor j/b \rfloor} \frac{\sqrt{2}}{2} c_{i-1,p_{\lfloor j/b \rfloor} b/2 + (j \bmod b)} + o_{\lfloor j/b \rfloor}$$

equivalent results but utilising a variety of notations.

If $\Delta_d = 2b$ the Haar wavelet transform of a signal contains the wavelet transform coefficients of the individual domain and range blocks, and the mapping process may be described as an extrapolation of fine resolution from coarse resolution coefficients in a fashion similar that described by Equation (3.3). A subtree in the Haar basis corresponds to an image block after subtraction of the DC component; the initial iteration required in [13] may be dispensed with if DC orthogonalisation is applied, as the range block averages immediately supply the values of $c_{\alpha,0}, c_{\alpha,1}, \ldots, c_{\alpha,2^\alpha-1}$. The full decoding process is described in Table 3.2.

Table 3.2: Wavelet Domain Fractal Decoding

Start with: Range partition of $N_r = 2^\alpha$ blocks
Non-overlapping domain partition ($\Delta_d = 2b$)
Signal is to be decoded to size $n = 2^\beta$
Block affine transforms remove DC component
Set of N_r fractal coefficients (o_i, s_i, p_i)

1. Create offset vector $\mathbf{o} = (o_0, o_1, \ldots o_{N_r-1})$ and apply wavelet transform to calculate $c_{0,0}$ and detail coefficients $d_{0,0}$ to $d_{\alpha-1,2^\alpha-1}$.

2. Iterate through resolutions $\alpha \leq j < \beta$
   Iterate through positions $0 \leq k < 2^j$
   Define $m = 2^{j-\alpha}$

$$d_{j,k} = \frac{1}{\sqrt{2}} s_{\lfloor k/m \rfloor} d_{j-1,mp_{\lfloor k/m \rfloor} + k \bmod m}$$

3. Perform inverse wavelet transform to calculate final signal

Davis [47] explicitly introduced the view of the process as mappings on subtrees, depicted

in Figures 3.4 and 3.5. Encoding in the wavelet transform domain is achieved by locating the best matching domain subtree for each range subtree, in the sense that the MSE distance between the range subtree and appropriately scaled domain subtree is a minimum. An analysis of error propagation in the extrapolation of coarse to fine resolution coefficients indicates a similar propagation of quantisation errors, suggesting higher weighting for the coarse resolution coefficients when matching subtrees [47].

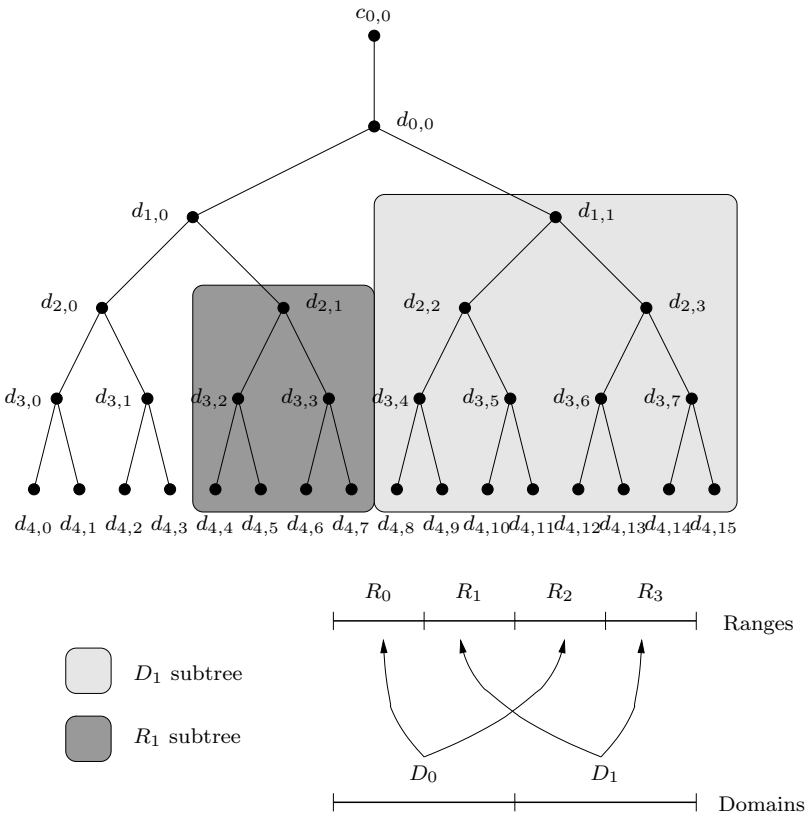


Figure 3.4: Domain and range blocks as wavelet detail coefficient subtrees.

The notion of an α-*scale-extending* operator was introduced by Davis [46] as an operator on the wavelet signal decomposition for which $d_{i,j}$ depends only on detail coefficients at the same or coarser resolution when $i \geq \alpha$. Only α iterations of an α-scale-extending operator, applied to a signal initialised with the known low resolution $i > \alpha$ coefficients, are required on decoding, and contractivity is not required for convergence (although it is required for coding utilising the collage theorem). In matrix notation, this type of linear operator is strictly lower triangular for all rows other than the initial α. Convergence problems for finer domain partitions may be seen as a result of dependency loops from fine to coarse coefficients; a consequence of representation by operators which are not α-scale-extending for the appropriate value of α.

The wavelet framework may be extended to images using the non-standard decomposition (see Appendix C). In this extension a subtree rooted at a particular position and resolution

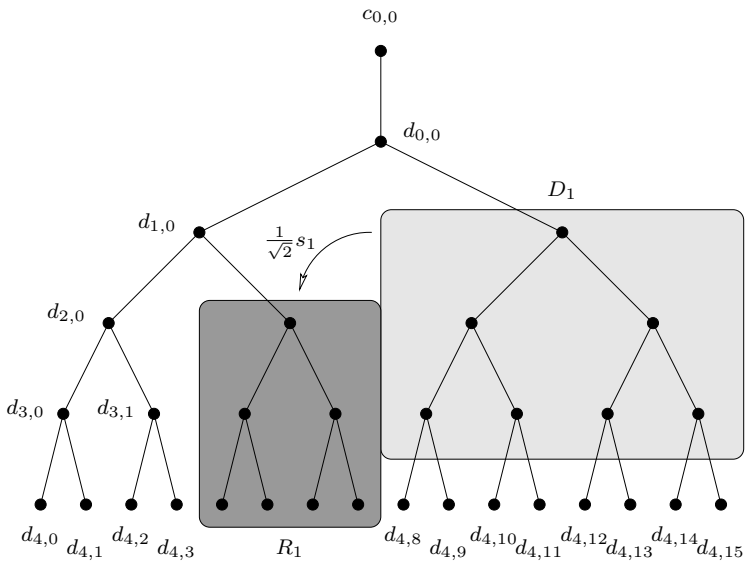


Figure 3.5: Detail coefficient extrapolation by subtree mappings.

contains the three directional subtrees[22] rooted at the same position and resolution. The block isometries may be applied in the wavelet domain and correspond to operations on the horizontal, vertical and diagonal directional subbands. A generalisation of the usual fractal coder is possible by applying the extrapolation process described above to detail coefficients of a signal with respect to a smooth wavelet basis. This significantly reduces the usual blocking artifacts and improves the reconstruction PSNR [47].

Van de Walle [194] [193] described a coder implementing a quadtree partition in the wavelet domain. Any range subtree which could not be acceptably approximated by a mapping from a domain subtree was split into separate subtrees, with scalar quantisation of the parent node. This scheme allows for a continuous transition between fractal coding and wavelet coefficient scalar quantisation.

The smooth wavelet basis coder described by Krupnik *et al.* showed significant advantages over the usual Haar basis, but the algorithm did not compare favourably with EZW [109]. Davis [46] described an advanced implementation with results very close to those of EZW, with an improvement of 1dB in PSNR over the other fractal coders with which it was compared. This coder utilised an advanced bit allocation algorithm, and explicitly included zerotrees in the coding.

3.9.4 Subband block prediction

In contrast to the generalisation of the usual *subtree* prediction described above, Rinaldo and Calvagno [162] [163] proposed a *subband* prediction scheme in the non-standard image decomposition. Each image subband was covered by range blocks which were mapped from domain blocks of the same size from the next lower resolution subband (see Figure 3.6).

[22] The subtrees in each of the directional subbands are combined to form a composite subtree.

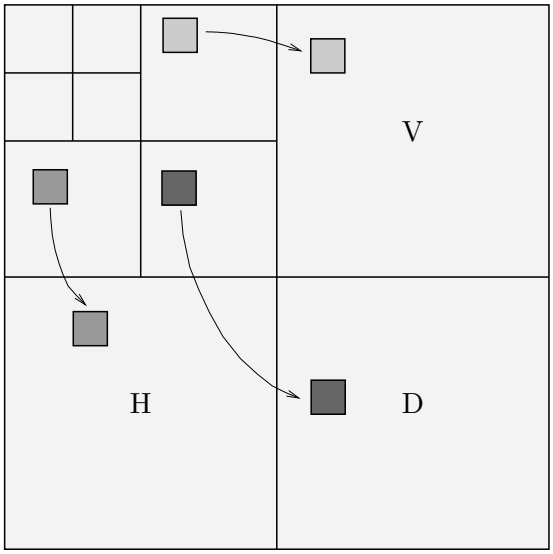


Figure 3.6: Subband prediction in the coding scheme of Rinaldo and Calvagno.

This form of prediction[23] is justified by the similarity between the psds of different signal subbands, suggesting that the spectral content of blocks used for domains should be similar to that of the range blocks. The use of domain and range blocks of the same size is motivated by the observation that the effects of an event such as an edge in a signal have similar durations in different subbands [163].

Maximum range block size was increased with increasing resolution, applying a quadtree splitting if a match threshold was not met. The block mappings consisted of a scaling factor and one of four rotations, and the domain search region was considerably reduced by restricting it to a small region within the previous subband. A considerable advantage of this method is that contractivity is not required, and the decoding error may be evaluated at coding time since each subband is predicted from the coded version of the previous subband[24].

In the initial system [163], the lowest resolution subbands, and any other subbands for which an adequate match was not found were scalar quantised using a Laplacian quantiser and the standard optimum bit allocation method. This method reduces to scalar quantisation of wavelet coefficients when block prediction is poor. The PSNR was significantly better than that for JPEG, with no visible blocking artifacts.

A subsequent system [162] employed a sophisticated Lattice Vector Quantisation technique called Pyramid Vector Quantisation (PVQ) [61] for coding of low resolution subbands (after application of a DCT) and residual errors after block prediction. This scheme reduces to PVQ when block prediction is poor. Results were improved over those in [163], and appear to compare well with those reported by Davis [46].

[23] This compression scheme differs substantially from Predictive VQ [73, ch. 13] where a vector is predicted based on a fixed predecessor.

[24] As a result, it is strictly speaking not fractal coding, despite the considerable similarities.

3.9.5 Fractal multiresolution analysis

Cheng and Zhu [35] defined a multiresolution-like analysis based on the "fractal transform", with finer resolution corresponding to a finer range block partition in a PIFS representation. The multiresolution properties achieved were

1. $V_j \subset V_{j+1}$
2. $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$
3. $\overline{\cup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$

for "approximation spaces" V_j, where item 3 is the important representation property.

Bogdan [27] [28] [29] [30] presented an idea very similar in spirit, but corresponding to a decomposition into detail signals. This decomposition of a function f is initialised by creating a low resolution representation[25] [30, pg. 74] f_N, after which a detail function f_{N-1} is created by standard fractal encoding of the original function f, with range blocks of size 2^{N-1} and the function f_N as affine part. This process is repeated [30, pg. 112], encoding f_N as

$$f_{i-1} = A_{i-1} f_{i-1} + f_i$$

with range block sides of 2^i until f_0 is reached, which provides an exact representation since range blocks correspond to one sample each. The low resolution signal f_N, together with the A_i comprise the function decomposition. The decomposition is however, not orthogonal [29] [30, pp. 79-82]. A compression system based on this decomposition produced similar performance to the JPEG standard in terms of PSNR, but superior subjective quality was claimed for low bit rates [29].

3.10 Performance comparisons

The greatest difficulty in comparing results of different lossy coding algorithms is the absence of an objective distortion measure which accurately reflects perceived distortion. A further complication in the comparison of fractal coding algorithms is the scarcity of theoretical results to support design choices. As a result, virtually every stage of coder design is based on empirical studies, and the lack of consensus on important issues is probably largely a result of the dependence between different aspects of fractal coder referred to in the introduction to this chapter. A "greedy algorithm" for coder optimisation which optimises each stage separately is therefore doomed to failure.

Since the most widely used test image is the 512×512 8 bit/pixel Lena image[26] (see Figure

[25] The index ordering is reversed from the order established here for wavelet decompositions.

[26] There are, unfortunately, slightly different versions of the same image in common use, which should be considered when making this type of comparison.

E.4), a comparison of published PSNR results is possible for a variety of coding schemes. A comparison[27] of published coding results for this image is displayed in Figure 3.7.

Performances vary considerably, but four different coders appear to offer comparable performance to the EZW algorithm. The first of these [21] is based on the original coder of Jacquin [92], but with VQ of transform coefficients and three levels of block splitting instead of two. The significantly superior performance of this system in comparison with that from which it was derived [92] is difficult to explain, but the most significant contributing factor is probably the additional attention paid to quantisation and representation of transform parameters. The second of these coders [20] is similar to the first, but is based on the adaptive frequency domain transform described in Section 3.2, with the addition of a transform optimisation procedure as described in Section 3.7. The third of these coders is a smooth wavelet transform domain fractal coder with globally optimised bit allocation [46], while the fourth is based on a form of vector prediction from one subband to the next of a smooth wavelet image decomposition [162].

Although the difference in performance between these four coders and the rest is large enough to suggest some common basis, this is difficult to isolate. Significant common features of the best four coders appear to be that they all use a quadtree[28] or multiresolution equivalent range partition, as well as including mechanisms for improved quantisation or bit allocation. The significance of the operation in the transform domain of three of these algorithms is highlighted by the absence of any transform domain coders in the less effective class of coders.

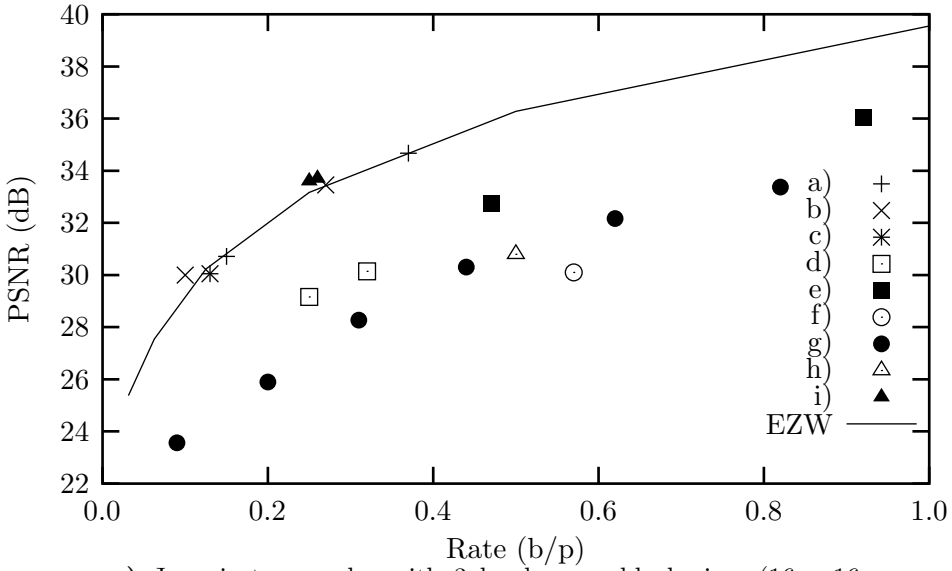
3.11 Conclusions

The absence of any solid theoretical foundation for making design decisions for a fractal coder is illustrated by the rather *ad hoc* nature of many of the wide variety of options that have been evaluated in each of the main categories of this survey. Although the recent wavelet domain analysis of fractal coding has contributed to an improved understanding of its operation, as well as resulting in a new generation of more effective fractal coding algorithms, the theoretical foundation remains tenuous. Nevertheless, the performance comparisons presented here imply that the better fractal coders offer rate distortion performance at least comparable with the current state of the art.

Despite the recent growth in research activity in the area of fractal image compression, a comprehensive explanation of the relationship between the assumptions made and the statistical properties of "natural" images remains to be presented (for example, neither of the two currently available books [16] [65] on the subject contain such an explanation). The

[27] Only those publications presenting actual PSNR and bit rate figures have been included; where these results are only displayed in graph form, they have been omitted.

[28] This is clearly not sufficient in itself for superior performance, since one of the poorest coders in this comparison is also based on a quadtree partition.



a) Jacquin-type coder with 3 level range block sizes (16×16, 8×8, 4×4), spiral domain search and VQ of transform coefficients [21].

b) Coder[a] as for a), but with DCT domain block transform [20].

c) Hybrid Self-Quantisation of Subtrees (SQS) coder with smooth wavelet basis [46].

d) Triangular partition coder [51].

e) HV partition coder [62].

f) Original Jacquin[b] 2-level coder [92].

g) Quadtree partition coder[121].

h) Fixed square block partition coder [149].

i) Subband block prediction with PVQ [162].

EZW EZW coder results [182].

[a]Superior results are achieved for a subsequent algorithm [19], but no actual PSNR values are given.

[b]In a later publication [95] Jacquin reports on a similar coder for which a PSNR of 31.4dB is achieved at a rate of 0.06 b/p; since the coder described is very similar to the earlier one [92], it is likely that a typographical error transformed 0.6 b/p into 0.06 b/p.

Figure 3.7: Comparison of the performance of fractal coding and EZW for the 512×512 8 b/p Lena image.

underlying assumption of fractal coding, which is presumably that each subblock in an image may be closely approximated by an affine transform on another subblock, has been described as affine redundancy [6], piecewise self-similarity [95], self-affinity [38] and a host of similar terms. Self-affinity is necessary, but not sufficient, for effective fractal coding since the union of the individual transforms is also required to be a contraction mapping in the space of images.

Exact self-affinity, in which each image subblock is exactly equal to an affine transform on another subblock corresponds to the type of *deterministic* fractal represented by the Sierpinski triangle (see Section 2.5.5) rather than form of *stochastic* fractal represented by fractional Brownian motion (fBm) [60]. Unfortunately, while stochastic fractals do find application in modelling properties of images, there is no evidence that images are effectively modelled as deterministic fractals [38]. Given the lack of evidence for the fundamental assumption of fractal image compression, it is certainly not clear why it is capable of effective image compression. Although the actual rate distortion performance of a compression scheme is not a perfect indicator of the soundness of the underlying signal model[29], performance observed to be comparable with other state of the art techniques suggests strongly that the fundamental assumptions are at least approximately accurate.

Three main questions are posed in evaluating the validity of the self-affinity assumption, namely

1. Is self-affinity a natural consequence of more common, or weak, statistical conditions?
2. What are the consequences of self-affinity on signals exhibiting this property?
3. To what extent do "natural" images exhibit this property?

These are the primary questions addressed in the research chapters of this dissertation.

[29] In transform coding, for example, some compensation for an inappropriate decorrelating transform may be achieved by the use of more effective quantisation and entropy coding stages.

Chapter 4

Self-Affinity of First Order Gauss-Markov Models

Since the statistical properties required for the existence of self-affinity are not well understood, it is not obvious whether self-affinity is a natural consequence of some weak statistical constraints, such as decaying autocorrelation, or whether it is dependent on the existence of a strongly restrictive set of conditions. This question was addressed by evaluating aspects of the performance of fractal compression, and hence the degree of self-affinity, for a standard class of signal models.

Despite their simplicity, first order Gauss-Markov models offer a number of advantages for application in this context:

- They often provide a reasonable approximation to measured image characteristics [37, ch. 2], and have been used with considerable success in modelling textured regions [115, pp. 656-658].
- Convenient performance benchmarks are available since the distortion rate function is known analytically [99, app. D], and transform coding is capable of close to optimum coding for this model [99, ch. 12].
- If the maximum correlation of $\rho = 1$ is selected the model generates Brownian motion signals, which are stochastic fractals [152].

A number of restrictions were, from practical necessity, imposed on the scope of evaluation for these models. First, although the main focus of this dissertation is fractal compression of *images*, all evaluations were performed on one-dimensional signals in order to reduce the computation required in Monte Carlo simulations, as well as to avoid the additional complications in two dimensions of the issue of the separability and the additional difficulty of presentation of results in graph form. While introducing additional complexity, there is no reason to expect (although this is not proved here) that qualitatively different behaviour would be observed for

an extension to two dimensions. Second, the set of fractal schemes investigated was restricted to a fixed block size partition, with domain blocks twice the width of range blocks, and block transforms as in Equation (3.1), with spatial contraction by a factor of two by averaging. Since only one-dimensional signals were considered, the square isometry operations are not applicable[1]. Finally, it should be emphasised that the meaning of self-affinity in a particular context is dependent on the structure of the fractal coding scheme in question.

4.1 The Signal Model

A Gaussian first-order Markov (also called first-order autoregressive, or AR(1)) process X_i is generated [99, ch. 2] by

$$X_i = Z_i + \rho X_{i-1},$$

where ρ is the correlation between consecutive samples and the Z_i, often referred to as the *innovations process*, are independently distributed Gaussian random variables with zero mean and variance σ_Z^2. The autocorrelation function of this process is

$$R_X(k) = \sigma_X^2 \rho^{|k|}$$

where

$$\sigma_X^2 = \frac{\sigma_Z^2}{1-\rho^2}.$$

A variety of aspects of fractal coding were investigated, in each case by an evaluation performed over an ensemble of signals generated by an AR(1) model with a particular correlation ρ. The variance σ_Z^2 was held constant at a value of 1.0 for all values of ρ so that identical distortion rate curves are associated with all correlation values (see Section 4.5.5). Evaluation was focussed primarily on the collage error and the efficiency of the domain pool as a codebook, although more practical signal coding issues such as transform parameter distributions were also investigated. The majority of the statistical evaluations described here are too complex for analytic evaluation, and were consequently performed by Monte Carlo simulation[2].

Given range and (spatially contracted) domain blocks $\mathbf{r}$ and $\mathbf{d}$ respectively, the collage[3]

[1] Block reflection, which remains an option in the one-dimensional case, was not utilised.

[2] The Gaussian innovations process values were generated by the *Normal* class, using the *ACG* random number generator class (with seed 94280 and size parameter 98), of the libg++ class library supplied with the g++ C++ compiler.

[3] The term collage error is used to describe three related but distinct quantities. The first, as used in this section, is the distance (in terms of the vector norm, not the spatial distance) between a range block and a particular transformed domain block. The second remains applicable to a *particular* range block, describing the minimum of the first type of collage error between that range and a pool of domain blocks, while the third usage corresponds to the sum of the second type over all range blocks in a signal.

error vector $\mathbf{e}_C$ for range $\mathbf{r}$ and domain $\mathbf{d}$ is

$$\mathbf{e}_C = \mathbf{r} - s\mathbf{d} - o\mathbf{1},$$

where s and o are the scaling and offset coefficients respectively. The square of the magnitude of the collage error vector is

$$\|\mathbf{e}_C\|^2 = \|\mathbf{r}\|^2 - 2s\langle\mathbf{r},\mathbf{d}\rangle - 2o\langle\mathbf{r},\mathbf{1}\rangle + 2so\langle\mathbf{d},\mathbf{1}\rangle + s^2\|\mathbf{d}\|^2 + o^2\|\mathbf{1}\|^2,$$

where $\|\mathbf{1}\|^2 = b$ is the range block width and $\frac{1}{b}\|\mathbf{e}_C\|^2$ is the collage error in terms of the MSE measure. The scaling factor and offset minimising $\|\mathbf{e}_C\|^2$ for fixed domain and range block are

$$s^* = \frac{\|\mathbf{1}\|^2\langle\mathbf{d},\mathbf{r}\rangle - \langle\mathbf{d},\mathbf{1}\rangle\langle\mathbf{r},\mathbf{1}\rangle}{\|\mathbf{1}\|^2\|\mathbf{d}\|^2 - \langle\mathbf{d},\mathbf{1}\rangle^2} \quad o^* = \frac{\|\mathbf{d}\|^2\langle\mathbf{r},\mathbf{1}\rangle - \langle\mathbf{d},\mathbf{1}\rangle\langle\mathbf{r},\mathbf{d}\rangle}{\|\mathbf{1}\|^2\|\mathbf{d}\|^2 - \langle\mathbf{d},\mathbf{1}\rangle^2},$$

where the offset may also be expressed as

$$o^* = \frac{\langle\mathbf{r},\mathbf{1}\rangle - s\langle\mathbf{d},\mathbf{1}\rangle}{\|\mathbf{1}\|^2}.$$

The collage error vector for $o = o^*$ is

$$\begin{aligned}\mathbf{e}_C &= \mathbf{r} - s\mathbf{d} - o^*\mathbf{1} \\ &= \left(\mathbf{r} - \frac{\langle\mathbf{r},\mathbf{1}\rangle}{\|\mathbf{1}\|^2}\right) - s\left(\mathbf{d} - \frac{\langle\mathbf{d},\mathbf{1}\rangle}{\|\mathbf{1}\|^2}\right)\end{aligned}$$

and the square of the magnitude in this case is

$$\|\mathbf{e}_C\|^2 = \|\mathbf{r}\|^2 - 2s\langle\mathbf{r},\mathbf{d}\rangle + s^2\|\mathbf{d}\|^2 - \frac{1}{\|\mathbf{1}\|^2}\langle\mathbf{r},\mathbf{1}\rangle^2 + \frac{2s}{\|\mathbf{1}\|^2}\langle\mathbf{r},\mathbf{1}\rangle\langle\mathbf{d},\mathbf{1}\rangle - \frac{s^2}{\|\mathbf{1}\|^2}\langle\mathbf{d},\mathbf{1}\rangle^2.$$

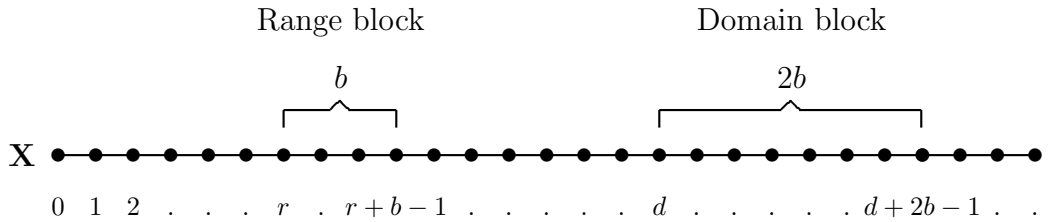


Figure 4.1: Domain and range blocks in a random signal $\mathbf{X}$.

If range and domain blocks are extracted from a random signal as in Figure 4.1, the terms

in the collage error expression become[4]

$$\langle \mathbf{r}, \mathbf{d} \rangle = \frac{1}{2} \sum_{i=0}^{b-1} X_{r+i} X_{d+2i} + \frac{1}{2} \sum_{i=0}^{b-1} X_{r+i} X_{d+2i+1}$$

$$\langle \mathbf{r}, \mathbf{1} \rangle = \sum_{i=0}^{b-1} X_{r+i} \qquad \langle \mathbf{d}, \mathbf{1} \rangle = \frac{1}{2} \sum_{i=0}^{2b-1} X_{d+i} \qquad \|\mathbf{r}\|^2 = \sum_{i=0}^{b-1} X_{r+i}^2$$

$$\|\mathbf{d}\|^2 = \frac{1}{4} \sum_{i=0}^{2b-1} X_{d+i}^2 + \frac{1}{2} \sum_{i=0}^{b-1} X_{d+2i} X_{d+2i+1}$$

where r and d are the positions of the range and domain blocks respectively, and where spatial contraction of domain blocks is achieved by averaging of adjacent pixels. Where necessary, the number of domain and range blocks are denoted by N_d and N_r respectively, and the domain increment is denoted by Δ_d.

4.2 Deterministic Correlation

If $s = s^*$ the collage error may be simplified to

$$\begin{aligned} \|\mathbf{e}_C\|^2 &= \left(\|\mathbf{r}\|^2 - \frac{1}{\|\mathbf{1}\|^2} \langle \mathbf{r}, \mathbf{1} \rangle^2 \right) - (s^*)^2 \left(\|\mathbf{d}\|^2 - \frac{1}{\|\mathbf{1}\|^2} \langle \mathbf{d}, \mathbf{1} \rangle^2 \right) \\ &= \left(\|\mathbf{r}\|^2 - \frac{1}{\|\mathbf{1}\|^2} \langle \mathbf{r}, \mathbf{1} \rangle^2 \right) (1 - \varrho^2(\mathbf{d}, \mathbf{r})), \end{aligned} \tag{4.1}$$

where

$$\varrho(\mathbf{d}, \mathbf{r}) = \frac{\langle \mathbf{d}, \mathbf{r} \rangle - \frac{\langle \mathbf{d}, \mathbf{1} \rangle \langle \mathbf{r}, \mathbf{1} \rangle}{\|\mathbf{1}\|^2}}{\sqrt{\|\mathbf{d}\|^2 - \frac{\langle \mathbf{d}, \mathbf{1} \rangle^2}{\|\mathbf{1}\|^2}} \sqrt{\|\mathbf{r}\|^2 - \frac{\langle \mathbf{r}, \mathbf{1} \rangle^2}{\|\mathbf{1}\|^2}}}$$

is the angle between vectors $\mathbf{d}$ and $\mathbf{r}$ after subtraction of their deterministic mean values, and may be interpreted as the deterministic[5] correlation coefficient by comparison with the usual definition of the statistical correlation coefficient. Equation (4.1) reveals that the collage error between a range and spatially contracted domain depends on the angle between the AC parts of the domain and range vectors, represented by $\varrho(\mathbf{d}, \mathbf{r})$, and the magnitude of the AC part of the range. Note that the optimum scaling factor may be expressed as

$$s^* = \frac{\|\mathbf{r} - \frac{\langle \mathbf{r}, \mathbf{1} \rangle}{\|\mathbf{1}\|^2}\|}{\|\mathbf{d} - \frac{\langle \mathbf{d}, \mathbf{1} \rangle}{\|\mathbf{1}\|^2}\|} \varrho(\mathbf{d}, \mathbf{r}),$$

which implies that the sign of ϱ indicates the sign of s^*.

Since the collage error $\|\mathbf{e}_C\|^2$ may be expressed as the product of a function of the range

[4]The range and domain vectors are random variables, but the bold lower case notation is retained for notational simplicity.

[5]This description is in analogy with the *deterministic autocorrelation* of a function [197, pg. 48].

vector and $(1 - \varrho^2)$, it may be minimised over a domain pool by maximising ϱ^2 (that is, given range vector $\mathbf{r}$, the best domain vector $\mathbf{d}$ in the domain pool is the one resulting in the largest value of $\varrho^2(\mathbf{d}, \mathbf{r})$). The square of the deterministic correlation represents the similarity between domain and range vectors dependent only on their "shape", and is invariant to their DC offsets and magnitudes; the larger this value the more similar the shapes. An exactly self-affine signal is one in which there is, for each range block, a corresponding domain block such that the square of their mutual deterministic correlation is unity (i.e. $\varrho^2 = 1$). Since the correlation is bounded ($0 \leq \varrho^2 \leq 1$) the maximum value of ϱ^2 over a set of domain vectors provides a convenient measure of the effectiveness of a particular domain pool in providing vectors of the appropriate shape. This measure of domain pool effectiveness clearly corresponds to a measure of its performance as a codebook.

An alternative measure of domain pool effectiveness may be based on the minimum collage error for each range, which is zero for exactly self-affine signals. If the ϱ^2 are independent[6] of the magnitude of the AC part of the range blocks this measure of self-affinity is equivalent to the ϱ^2 based measure.

4.2.1 Variation with positional offset between range and domains

The expected[7] deterministic correlation between a range block with its centre at the centre of a signal and each domain extracted[8] from the same signal is graphed in Figure 4.2 for $b = 16$, with slices at various ρ values graphed in Figure 4.3. Note that positive scaling coefficients are likely for domains near the range centre, while negative scaling coefficients become likely further away as a result of the negative expected deterministic correlation[9]. The expected deterministic correlation falls to zero within a small distance of the range block for all values of ρ, the greatest distance being required for $\rho \approx 0.95$, and there is a significant increase in the expected value of ϱ at the position of the range block as the signal correlation increases.

Figures 4.4 and 4.5 display the expected square of the deterministic correlation for $b = 8$ and $b = 16$ respectively, while slices at various ρ values are graphed in Figure 4.6. There is a small trough to either side of the peak for larger values of ρ, and the squared deterministic correlation approaches a constant within a short spatial distance of the peak. The magnitude of the central peak increases with increasing signal correlation. These observations reveal that there is a general improvement in the shape match between the range and domain blocks with increasing signal correlation, with the best expected match for the domain block with

[6] This would not be the case if, for example, a particularly poor match in terms of ϱ^2 were more probable for range blocks of particularly high magnitude.

[7] Expectations were calculated for an ensemble of 100000 signals generated by the appropriate AR(1) model. The deterministic correlation was calculated between the range block at the centre and all domains for each signal.

[8] The location of a domain block is determined by the position of its centre in this section.

[9] The quantisation of the scaling coefficient should therefore depend on the spatial distance between the range block and selected domain block.

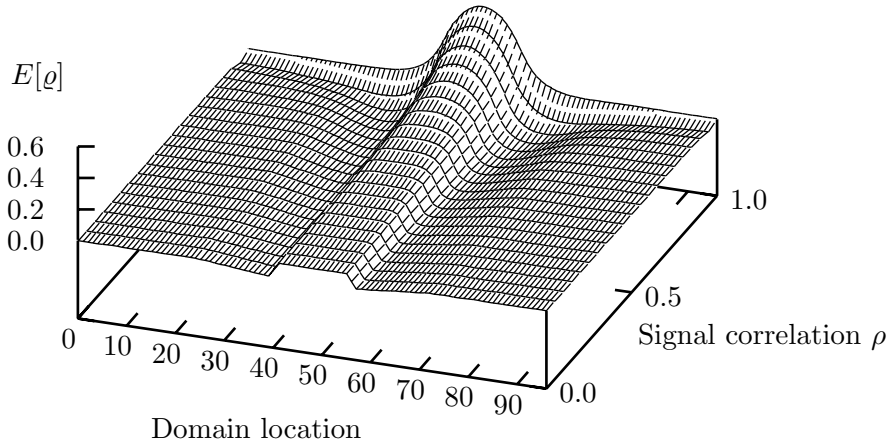


Figure 4.2: Expected deterministic correlation for $b = 16$. The domain and range block centres coincide at the line of symmetry.

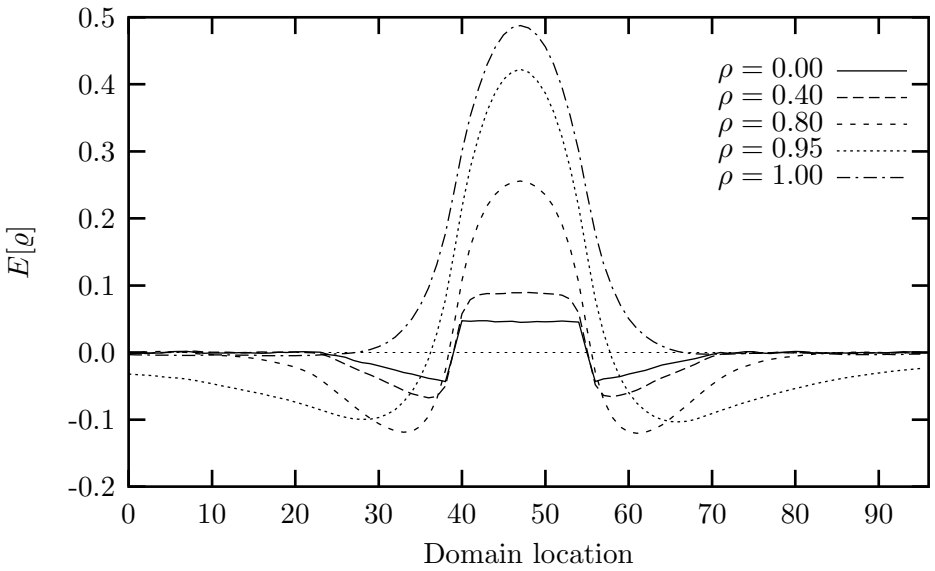


Figure 4.3: Expected deterministic correlation for $b = 16$. The domain and range block centres coincide at the line of symmetry.

centre coinciding with that of the range block. This is not surprising, since domain blocks become progressively less statistically connected to the range block as their spatial distance from it is increased.

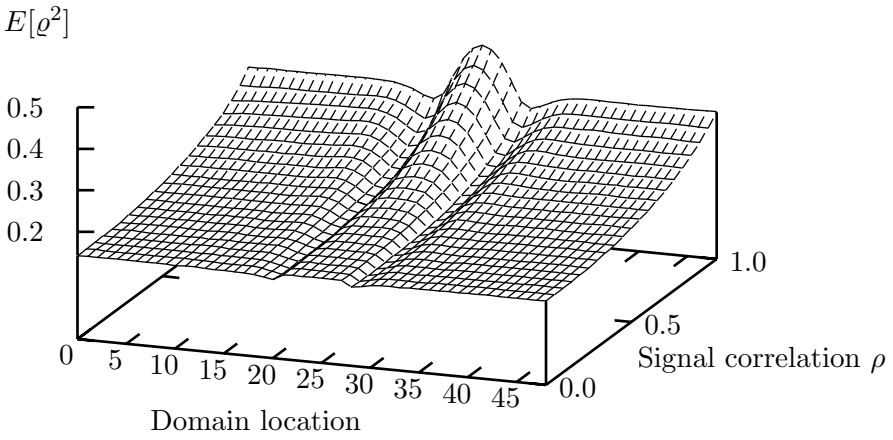


Figure 4.4: Expected squared deterministic correlation for $b = 8$. The domain and range block centres coincide at the line of symmetry.

Since the collage construction process entails locating the optimum domain for each range, the significant value in this context is the maximum deterministic correlation over a set of domain blocks, rather than for a single domain block. It is unfortunately not possible to determine the expected maximum ϱ^2 over a domain pool directly from the expectations for each domain. Since $E[\max_i\{X_i\}] \geq \max_i\{E[X_i]\}$ the peak value of the expected ϱ^2 is a lower bound for the maximum over a domain pool. While this lower bound is higher for a domain pool including the central domain than one excluding it, this does not necessarily guarantee a higher expected maximum.

4.2.2 Maximisation over a domain pool

The running maximum deterministic correlations encountered while visiting domain blocks, starting at the range position (range centres are at the start of the displayed signal for these results) and moving forwards in the signal, are displayed in Figure 4.7. Values are the lowest for $\rho = 0.0$, and increase with increasing ρ and domain pool size.

One might expect, particularly for high correlation signals, that neighbouring domain blocks would be very similar, resulting in a domain pool with a small domain increment forming an inefficient codebook. The comparison, presented in Figure 4.8, of maximum ϱ^2 values achieved over a domain pool of 64 domains, confirms this suspicion, with significantly

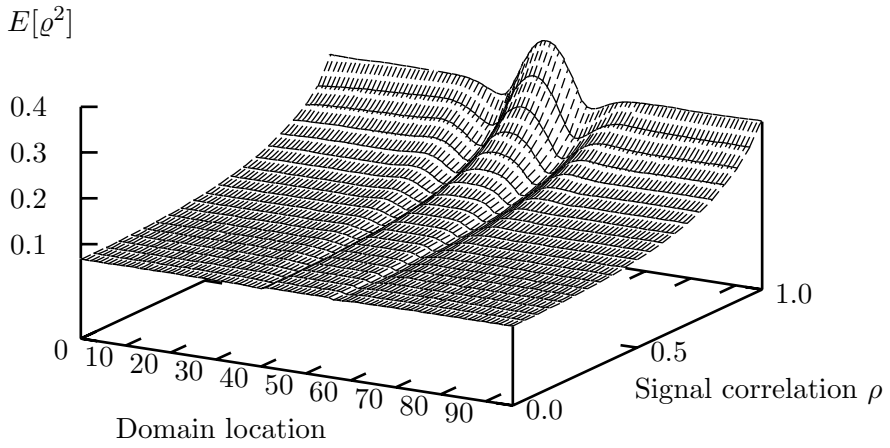


Figure 4.5: Expected squared deterministic correlation for $b = 16$. The domain and range block centres coincide at the line of symmetry.

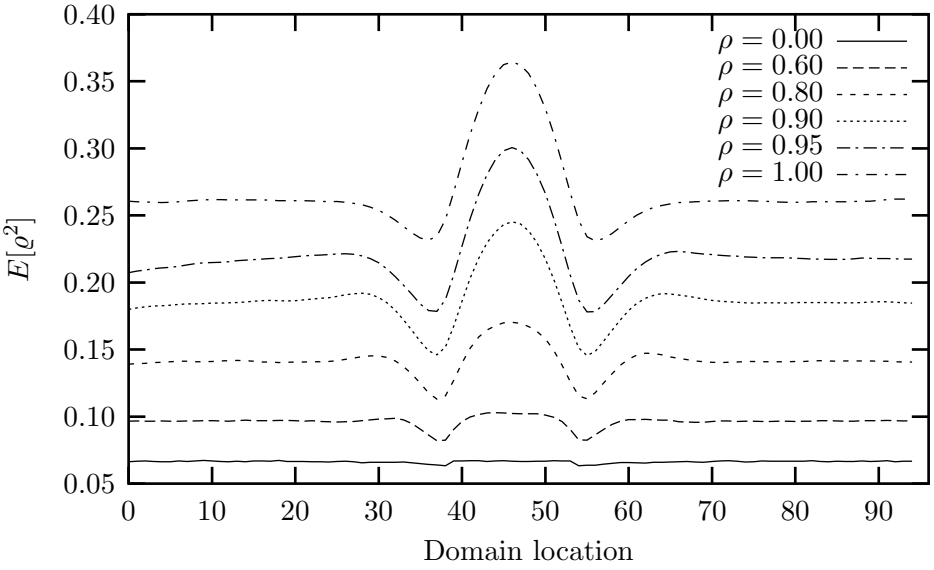


Figure 4.6: Expected squared deterministic correlation for $b = 16$. The domain and range block centres coincide at the line of symmetry.

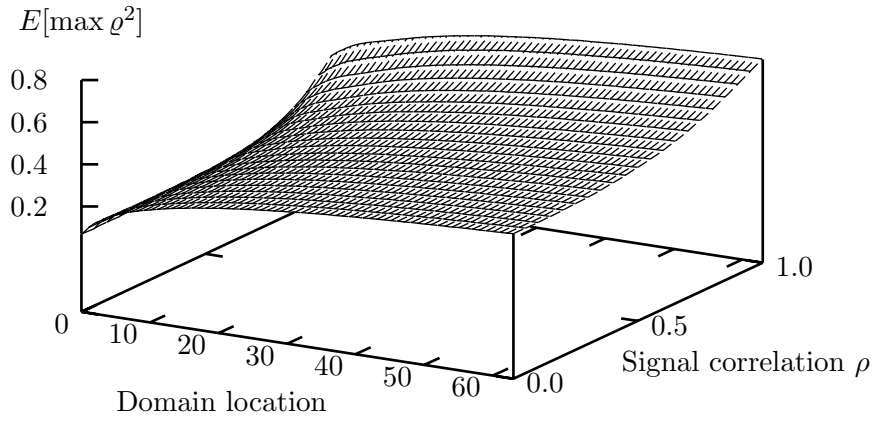


Figure 4.7: Expected running maximum squared deterministic correlation for $b = 16$.

improved performance for larger domain increments, although no significant improvement is observed for $\Delta_d > 4$ in this case.

4.2.3 Comparison with random codebooks

It is clear from the definition of self-affinity as the ability of each subblock in a signal to be approximated by an affine transform on another subblock, that the presence of this property to any significant extent requires some "special relationship" between subblocks in the same signal, leading to an unusual degree of affine similarity between them. The existence of such a relationship was determined by comparing the performance of the domain pool codebook with that of a random codebook [73, pg. 359] consisting of blocks with the same distribution as the domain blocks[10], but not extracted from the same signal as the range blocks (see Figure 4.9), and consequently independent of them. Comparisons were also made between "forward" and "reverse" running maxima, as depicted in Figure 4.9, to determine the significance of the central peak in expected ϱ^2 (described in Section 4.2.1).

Results of such a comparison for $\Delta_d = 1$ and $\rho = 1.0$ are displayed in Figure 4.10. The initial separation between "forward" and "reverse" curves, a result of the initial peak in expected ϱ^2, decreases rapidly with increasing domain pool size, reaching zero where the two domain pools coincide exactly. This suggests that although the central peak in expected ϱ^2 does increase the expected maximum over a domain pool in which it is included, it does not have a particularly significant effect in a large domain pool, and one would therefore not expect

[10]That is, the same intra-domain distribution, not the inter-domain distribution, since these blocks are independent of each other.

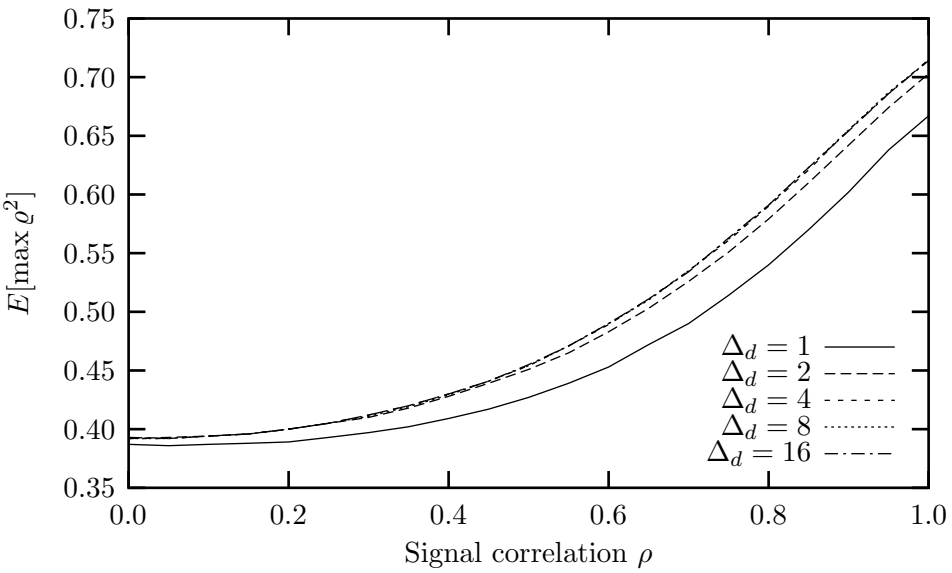


Figure 4.8: Comparison of expected maximum squared deterministic correlations for different domain increments ($b = 16$ and $N_d = 64$).

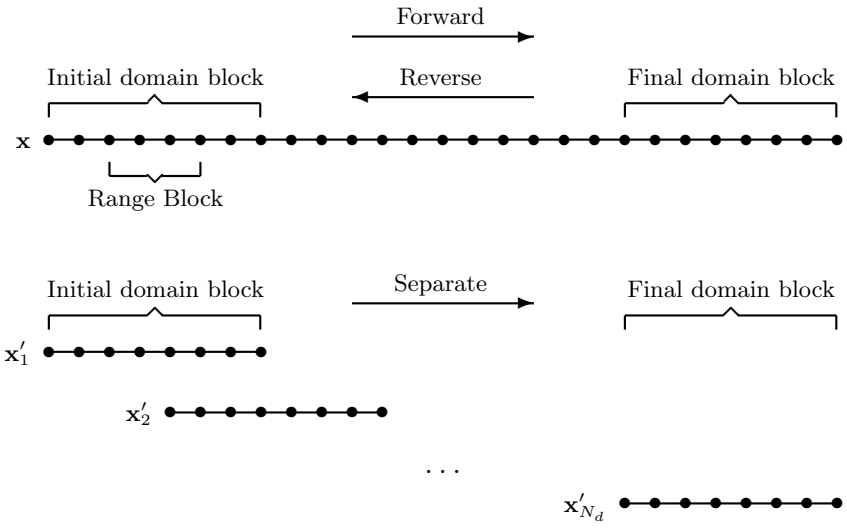


Figure 4.9: "Forward", "reverse" and "separate" domain sets. The range block is selected at the fixed position indicated in signal $\mathbf{x}$, which is a member of the ensemble of randomly generated signals. The "forward" domain set is searched from the initial domain block, while the "reverse" domain set is searched in reverse order. The domain blocks in the "separate" set are each extracted from separate signals $\mathbf{x}'_j$, generated by the same random process as $\mathbf{x}$.

significant domain locality[11] for a large domain pool. Such an effect may partially explain the disagreement in the literature over the existence of domain locality, since its significance is dependent on the size of the domain pool. The change in gradient of the "forward" curve in the region of the domain at location 10 is attributable to the corresponding ϱ^2 trough in that vicinity. The superior performance of the "separate" random codebook in this case is largely a result of the domain pool inefficiency related to the small domain increment. The "forward" and "reverse" curves are virtually indistinguishable in similar comparisons for $\rho = 0.0$, while the "separate" curve is slightly superior. The curves diverge slowly as ρ is increased, reaching the positions depicted in Figure 4.10 when $\rho = 1.0$.

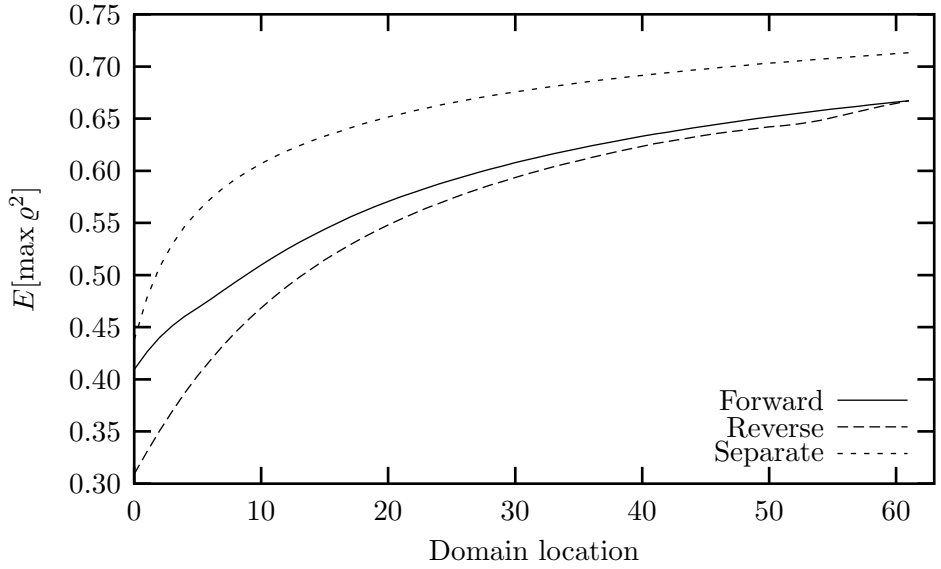


Figure 4.10: Comparison of expected running maximum squared deterministic correlations for "forward", "reverse" and "separate" codebooks ($b = 16$, $\Delta_d = 1$ and $\rho = 1.0$).

A similar comparison for $\Delta_d = 16$ resulted in very similar "forward", "reverse" and "separate" curves for all values of ρ, but with the curves diverging slightly with increasing ρ. The "forward" curve contained the highest values and the "separate" curve the lowest for ρ near unity, with the "reverse" curve sandwiched between them, equal to the "separate" curve to the left of the graph and ending with the same value as the "forward curve". The domain pool codebook is therefore slightly better than the "separate" form of random codebook for large domain increments, but with the significance of the difference decreasing with increasing domain pool size. The similarity between "forward" and "separate" curves is to be expected; since the AR(1) models are stationary and ergodic, a domain sufficiently far away from a range in the *same* signal might as well have been extracted from a *separate* signal. One may conjecture that the slight superiority of the domain pool codebook is a residual effect of the central ϱ^2 peak, which was the only form of "special relationship" or statistical dependence

[11] In the sense that the best domain for a range is likely to be found spatially close to it.

observed between domain and range blocks in a signal.

An interesting phenomenon was observed in comparing performances of the "separate" random domain codebook as above and a similar "separate" codebook constructed with random range block sized signals which were not spatially contracted. The resulting curves were indistinguishable for $\rho = 0.0$, with the random codebook without spatial contraction improving relative to the random codebook with spatial contraction until $\rho \approx 0.6$, returning to similar curves for $\rho \approx 0.85$ and the positions reversing thereafter. The relevant model statistics are therefore approximately invariant under spatial contraction, justifying to some extent the use of contracted domains as a form of random codebook for quantisation of range blocks.

4.2.4 Maximisation analysis

As noted in Section 4.2.1, the expected maximum ϱ^2 over a set of domains can not be determined from the individual expectations of ϱ^2 for each of the domains, and is dependent on the multivariate probability distribution of all the ϱ^2 involved[12]. The marginal distributions of ϱ^2 for the two domains closest to the range are compared with the marginal distribution[13] of ϱ^2 for the "separate" random domains in Figure 4.11. Although there is a significant difference between the distributions corresponding to domains 0 and 1, distributions corresponding to the subsequent domains are virtually identical to the "separate" distribution. This observation is consistent with the conjecture that there is no significant statistical relationship, resulting from their extraction from the same signal, between a range block and spatially removed domains.

The distribution of a random variable Y defined as the maximum of k independent identically distributed random variables, with the same distribution as a random variable X, is [110, pg. 245]

$$f_Y(y) = k f_X(y)[F_X(y)]^{k-1}.$$

Since the distributions of the ϱ^2 are not all identical or exactly independent, this model does not describe the observed distributions of the maxima very accurately, although there are strong qualitative similarities. The observed distributions of the maxima of ϱ^2 over a set of domain blocks are displayed in Figure 4.12.

4.2.5 Vector quantisation

Although the domain pool performance was found to be similar to that of a random codebook, fractal compression can only be truly effective if the domain pool codebook is at least of comparable efficiency to that of an equivalent fixed and optimised codebook. A comparison with

[12] That is, the ϱ^2 between the range and each of the domain positions in the domain pool.
[13] As expected the ϱ^2 for "separate" random domains are identically distributed.

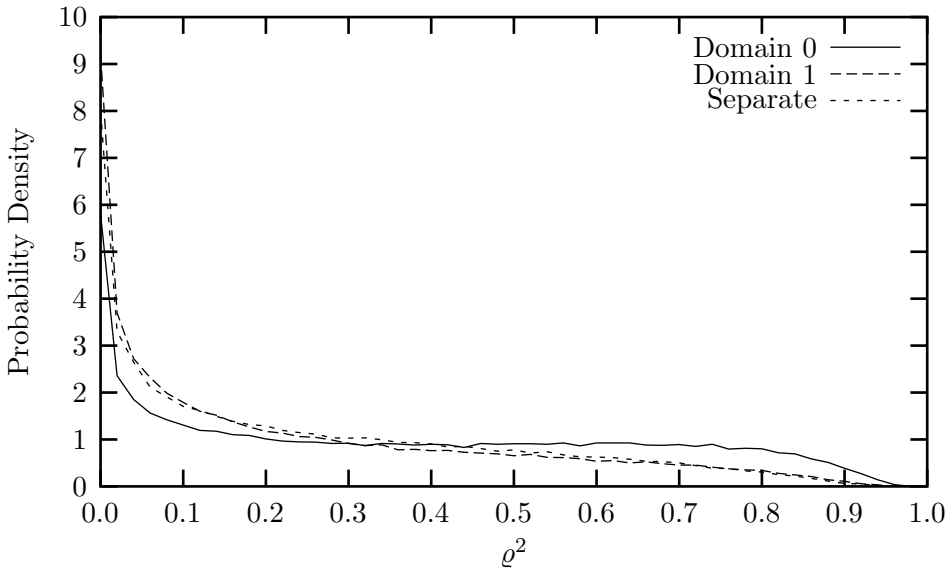


Figure 4.11: Marginal distributions of squared deterministic correlation for $b = 16$, $\Delta_d = 8$ and $\rho = 1.0$. Domains are labelled from left to right, starting with 0 for the domain with centre coinciding with that of the range block.

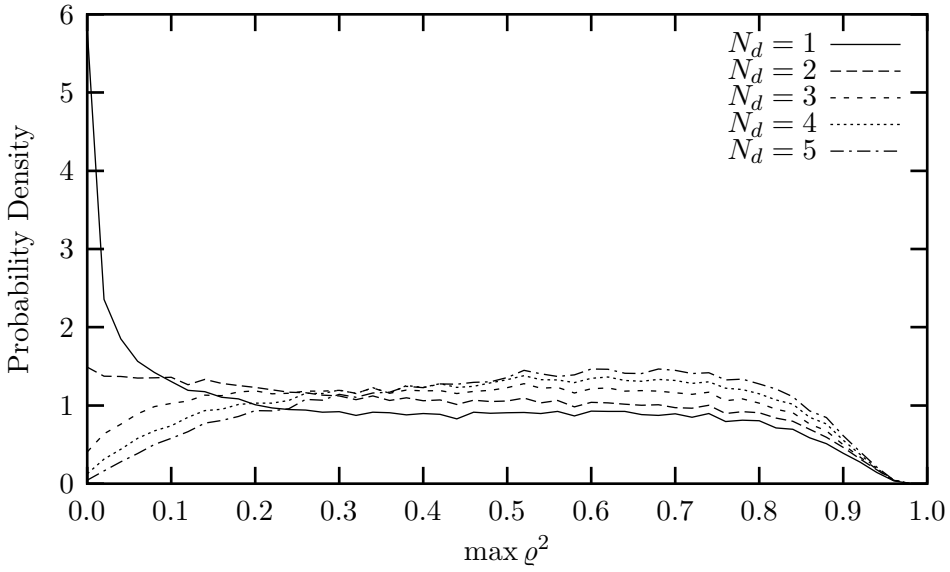


Figure 4.12: Marginal distributions of maximum squared deterministic correlation for $b = 16$, $\Delta_d = 8$ and $\rho = 1.0$. The domain pool size is increased by including additional domains to the right, starting with domain 0 of Figure 4.11.

the performance of an optimised codebook was achieved by generating initial codebook and training vectors according to the appropriate AR(1) model. The Mean Removed Shape Gain VQ [139] codebooks appropriate for comparison with the domain pool were constructed using the SGVQ GLA maximising squared deterministic correlation (described in Section 6.4.1), by applying mean removal to each of the initial vectors[14]. The comparison between the performance of a domain pool codebook, a random "separate" codebook and a fixed and optimised VQ codebook, presented in Figure 4.13, shows a significant advantage to the VQ codebook over the other two, which are virtually indistinguishable. The general improvement in performance with increasing ρ may be attributed to the concomitant narrowing of the variety of block "shapes", resulting in a more clustered distribution of the "shape" vectors. Although the domain pool is as effective as a random codebook, it is substantially outperformed by an optimised codebook, especially for the higher signal correlations.

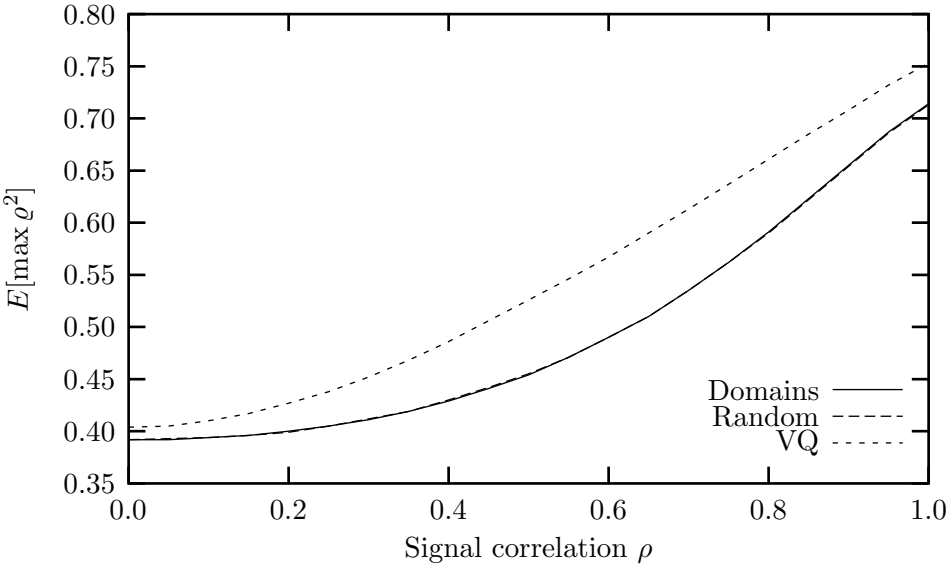


Figure 4.13: Comparison of expected maximum squared deterministic correlations for self, random and optimised codebooks ($b = 16$, $\Delta_d = 8$ and $N_d = 64$).

4.3 Collage Error

Since the factors $\left(\|\mathbf{r}\|^2 - \frac{1}{\|\mathbf{1}\|^2}\langle \mathbf{r}, \mathbf{1}\rangle^2\right)$ and $(1 - \varrho^2(\mathbf{d}, \mathbf{r}))$ in the expression for $\|\mathbf{e}_C\|^2$ are not necessarily independent, it is not certain that the rankings in terms of ϱ^2 of the previous section are to be observed for an evaluation in terms of the expected collage error. An analytic evaluation of this expectation is shown to be possible for a restriction of the usual block transforms, while Monte Carlo simulation is required for the more general case.

[14]Since the centroid of a set of zero deterministic mean vectors is itself zero mean, the GLA as described locates an optimal mean-removed codebook.

4.3.1 Affine transforms with fixed scaling

Analytic evaluation of $E[\|\mathbf{e}_C\|^2]$ is complicated by the factors of s in the expansion in terms of inner products and magnitudes of the domain, range and constant vectors. If s is fixed[15] however, $E[\|\mathbf{e}_C\|^2]$ may be expressed in terms of the autocorrelation function of $\mathbf{X}$. The expectations of the required terms for wide-sense stationary $\mathbf{X}$ are

$$\begin{aligned}
E[\|\mathbf{r}\|^2] &= bR_X(0) \\
E[\langle \mathbf{r}, \mathbf{d}\rangle] &= \frac{1}{2}R_X(d-r) + \sum_{i=1}^{b-1} R_X(d-r+i) + \frac{1}{2}R_X(b+d-r) \\
E[\|\mathbf{d}\|^2] &= \frac{b}{2}(R_X(0) + R_X(1)) \\
E[\langle \mathbf{r}, \mathbf{1}\rangle^2] &= bR_X(0) + 2\sum_{i=1}^{b-1} iR_X(b-i) \\
E[\langle \mathbf{r}, \mathbf{1}\rangle\langle \mathbf{d}, \mathbf{1}\rangle] &= \frac{1}{2}\sum_{i=1}^{b-1} iR_X(d+2b-r-i) + \frac{b}{2}\sum_{i=0}^{b} R_X(d-r+i) + \\
&\quad \frac{1}{2}\sum_{i=1}^{b-1} iR_X(d-b-r+i) \\
E[\langle \mathbf{d}, \mathbf{1}\rangle^2] &= \frac{b}{2}R_X(0) + \frac{1}{2}\sum_{i=1}^{2b-1} iR_X(2b-i),
\end{aligned}$$

resulting in an expression for the expected collage error between the range block at r and the domain block at d (as in Figure 4.1) as

$$\begin{aligned}
E[\|\mathbf{e}_C\|^2] &= bR_X(0) - \\
&\quad 2s\left(\frac{1}{2}R_X(d-r) + \sum_{i=1}^{b-1} R_X(d-r+i) + \frac{1}{2}R_X(b+d-r)\right) + \\
&\quad \frac{bs^2}{2}\left(R_X(0) + R_X(1)\right) - \\
&\quad \frac{1}{b}\left(bR_X(0) + 2\sum_{i=1}^{b-1} iR_X(b-i)\right) + \\
&\quad \frac{s}{b}\left(\sum_{i=1}^{b-1} iR_X(d+2b-r-i) + b\sum_{i=0}^{b} R_X(d-r+i) + \sum_{i=1}^{b-1} iR_X(d-r-b+i)\right) - \\
&\quad \frac{s^2}{b}\left(\frac{b}{2}R_X(0) + \frac{1}{2}\sum_{i=0}^{2b-1} iR_X(2b-i)\right).
\end{aligned}$$

[15] Barnsley [16, pp. 188-215] utilises such a scheme as an example.

Utilising the identities [186, pg. 107]

$$\sum_{i=0}^{n-1} ar^i = \frac{a(1-r^n)}{1-r} \quad r \neq 1$$

and

$$\sum_{i=0}^{n-1} (a+id)r^i = \frac{a(1-r^n)}{1-r} + \frac{rd\{1-nr^{n-1}+(n-1)r^n\}}{(1-r)^2} \quad r \neq 1,$$

and substituting in $R_X(k) = \sigma_X^2 \rho^{|k|}$, gives the expected squared distortion for an $AR(1)$ model (with the restriction $d \geq r+b+1$ required to avoid additional complication stemming from the absolute value in the expression for $R_X(k)$) as

$$\begin{aligned} E[\|\mathbf{e}_C\|^2] \quad = \quad & \frac{\sigma_Z^2}{1-\rho^2}\left(b - s\rho^{d-r} - 2s\rho^{d-r}\frac{\rho-\rho^b}{1-\rho} - s\rho^{b+d-r} + \right. \\ & \frac{bs^2}{2} + \rho\frac{bs^2}{2} - 1 - \frac{2}{b}\frac{b\rho(1-\rho)+\rho(\rho^b-1)}{(1-\rho)^2} + \\ & \frac{s}{b}\rho^{b+d-r}\frac{b\rho(1-\rho)+\rho(\rho^b-1)}{(1-\rho)^2} + s\rho^{d-r}\frac{1-\rho^{b+1}}{1-\rho} + \\ & \frac{s}{b}\rho^{d-r-b}\frac{\rho(1-b\rho^{b-1}+(b-1)\rho^b)}{(1-\rho)^2} - \frac{s^2}{2} - \\ & \left. \frac{s^2}{2b}\frac{2b\rho(1-\rho)+\rho(\rho^{2b}-1)}{(1-\rho)^2} \right). \end{aligned}$$

The complexity of this expression for such a simple case illustrates the difficulty inherent in an analysis of more general cases. In particular, the complex dependence on parameters ρ, b and s suggests that similar calculations for a more general case are unlikely to be fruitful. As one might expect, however, the above expression is linear in σ_Z^2 and the only dependence on d and r is on the term $d-r$.

The restriction $d \geq r+b+1$ unfortunately excludes the validity of the expression from the "interesting" region in the immediate vicinity of the range block. The behaviour within the region of validity is qualitatively similar to that resulting from the more general transform applied in the following section. In particular, it is interesting to note that for $b = 4$ the expected collage error decreases with increasing ρ, while the response is reversed for $b \geq 6$.

4.3.2 Affine transforms with variable scaling

Expected collage error results for optimal scaling as in Equation (4.1) are displayed in Figures 4.14 through 4.17. The general dependence on domain location is approximately what one might expect from the deterministic correlation results, with the most significant features being a minimum where the domain and range centres coincide, followed to either side of the centre by a maximum which decays to an approximately constant level as the domain

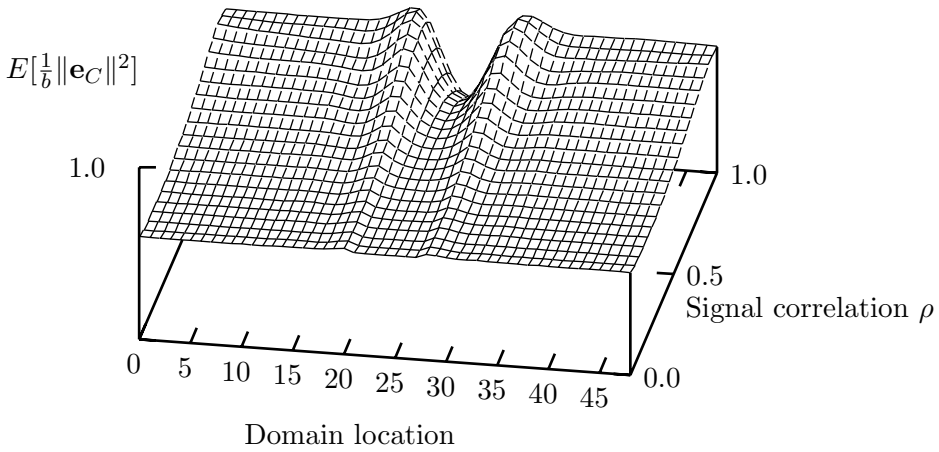


Figure 4.14: Expected mean square collage error for $b = 8$.

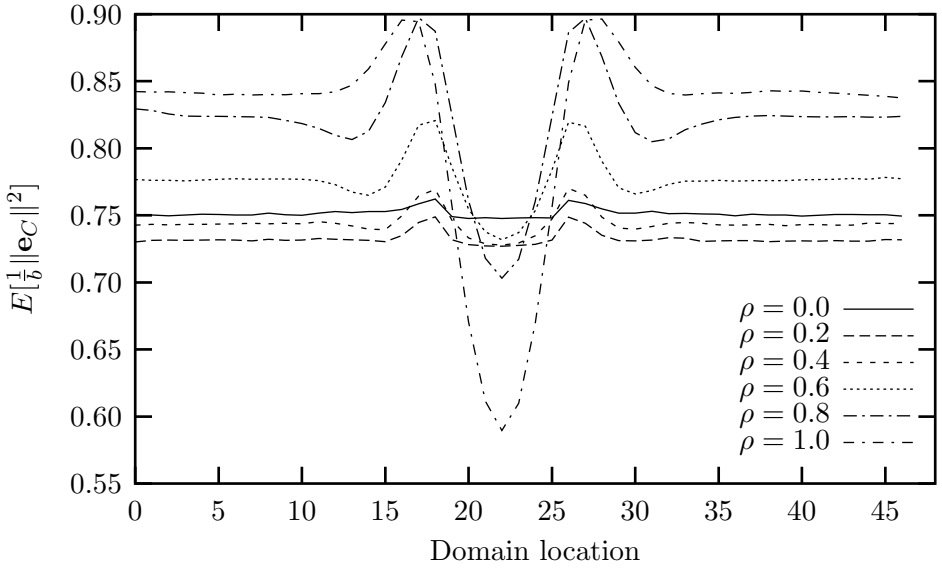


Figure 4.15: Expected mean square collage error for $b = 8$.

becomes further away from the range.

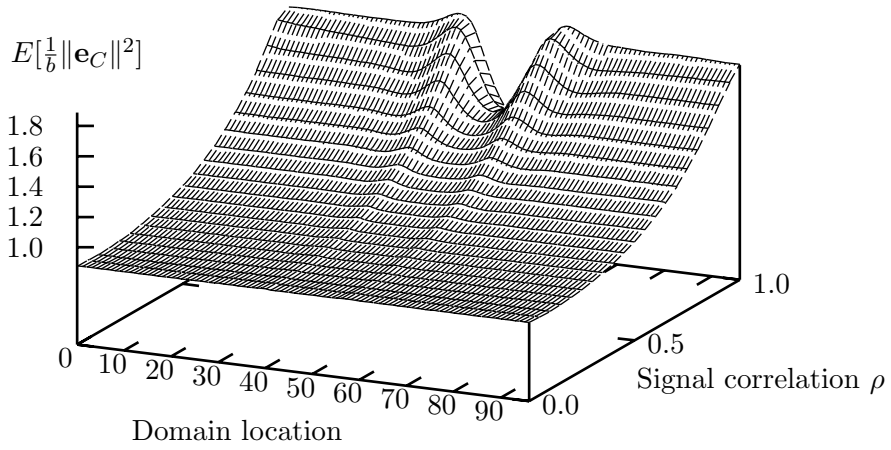


Figure 4.16: Expected mean square collage error for $b = 16$.

Comparisons of expected minimum collage error for the "forward", "reverse" and "separate" codebooks revealed the same rankings[16], for both $\Delta_d = 1$ and $\Delta_d = 16$, as those observed in terms of ϱ^2 in Section 4.2.3.

The expected maximum collage error for the domain pool, a separate random codebook and a fixed optimised codebook are compared in Figure 4.18. The performance of the optimised VQ codebook is significantly superior at high signal correlations, while the other two are very similar for all signal correlations. The complex shape of the curves for the domain pool and separate random codebooks is likely to be a consequence of the dependence of $\|\mathbf{e}_C\|^2$ on the magnitude of the AC part of the range as well as on ϱ^2, which each have different responses to increasing signal correlation[17].

There is an obvious improvement in performance of the VQ codebook with increasing signal correlation, despite the invariance of the distortion rate function to changing correlation. This observation should be considered in the light of the suboptimality of the VQ codebook, resulting both from the restricted block sizes and its product code structure.

4.4 Self-Affinity

Since most signal models are clearly not exactly self-affine, a means of quantifying the degree of approximate self-affinity is required. Although one could base such a measure on a comparison between the performance of fractal compression and some benchmark algorithm for the signal model of interest, it would be extremely complex and subject to the vagaries of representation, quantisation, and entropy coding, and consequently dependent on factors other than the

[16]The best performance corresponds to the *largest* ϱ^2 values and the *smallest* collage errors.
[17]Note that the signal variance increases with increasing ρ since σ_Z^2 was held constant.

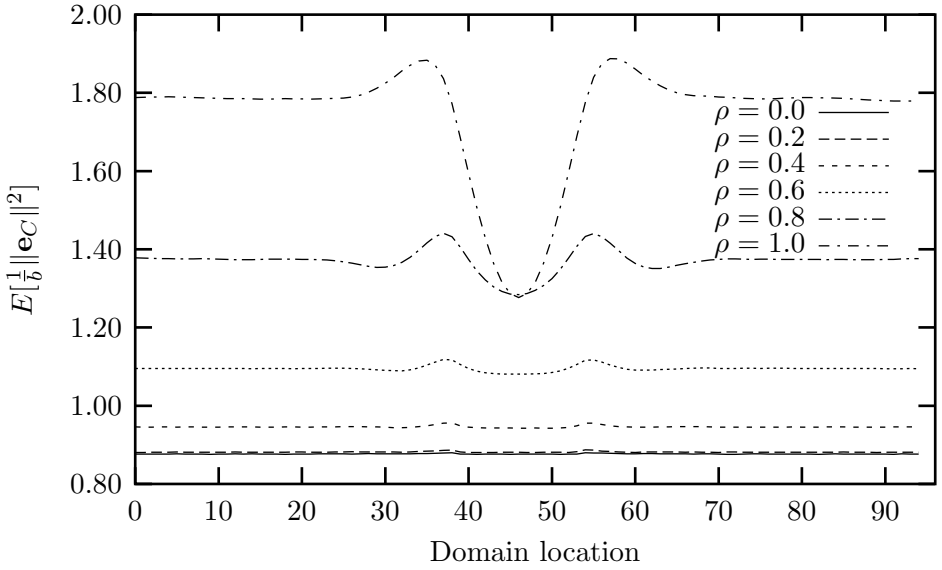


Figure 4.17: Expected mean square collage error for $b = 16$.

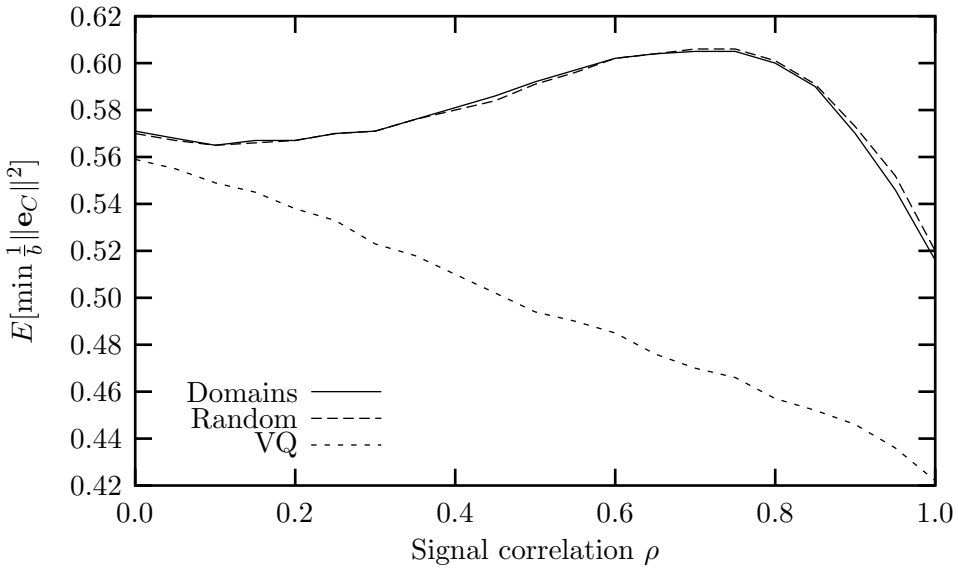


Figure 4.18: Comparison of expected minimum mean square collage error for self, random and optimised codebooks ($b = 16$, $\Delta_d = 8$ and $N_d = 64$).

accuracy of the underlying assumptions of fractal compression. The measure proposed here is based on the performance of the domain pool as a Mean Removed Shape Gain codebook [73, ch. 12] [139], which may be conveniently quantified as

$$\mathcal{S} = E[\max \varrho^2]$$

since $0 \leq \mathcal{S} \leq 1$. The effectiveness of a codebook of fixed size increases with increasing $\mathcal{S}$, and for a domain pool codebook $\mathcal{S} = 1$ corresponds to exact self-affinity[18].

Consider, however, constructing a fixed codebook, of the same size as the domain pool, by arranging the vectors to maximise the minimum ϱ^2 between any pair of code vectors. There is clearly a minimum ϱ^2 between a range vector and any of these vectors, increasing with increasing codebook size and decreasing with increasing vector dimensionality (this minimal value is a variant of the *minimal angle* for spherical codes [41, ch. 1]). A domain pool efficiency of $\mathcal{S}$ close to unity therefore does not necessarily indicate the presence of any "special relationship" between domain and range blocks, since a similar value might be obtained for a fixed codebook. The self-affinity should thus be determined by a comparison of $\mathcal{S}_{\text{self}}$ calculated for the usual domain pool with suitable reference values, such as $\mathcal{S}_{\text{random}}$ for a set of random domains with the same distribution as the signal, or $\mathcal{S}_{\text{VQ}}$ for a fixed codebook optimised for that pdf. The classification suggested here is as "strongly self-affine" if $\mathcal{S}_{\text{self}} > \mathcal{S}_{\text{VQ}}$ and "weakly self-affine" if $\mathcal{S}_{\text{self}} > \mathcal{S}_{\text{random}}$. Weak self-affinity indicates the presence of the required statistical dependence between range and domain blocks, whereas strong self-affinity indicates that this dependence is sufficiently strong for fractal coding to be able to compete with an equivalent form of VQ. Strong self-affinity, as defined in terms of the performance of the domain pool codebook, provides a necessary but not sufficient condition for the effective operation of fractal coding, since practical signal coding issues such as contractivity requirements are not considered.

A variety of alternative measures of self-affinity are possible, including one based on a comparison of the minimum expected distortion for a domain pool codebook and reference codebooks. Reference distortion values might reasonably be for Mean Removed Shape Gain codebooks, or for unrestricted codebooks of vectors of the same block size. Since the influence of the observed local dependence between domains and ranges is significant but small (especially for large domain pools), AR(1) models are only weakly self-affine to a marginal extent, and are certainly not strongly self-affine[19].

[18]This measure of self-affinity is related to the minimum attainable collage error rather than the minimum attainable distortion on decoding (see Section 3.7), and thus does not correspond *exactly* to the suitability of signals for fractal representation, particularly when collage optimisation is allowed. The correspondence is exact, however, for exact self-affinity, and may be expected to increase in accuracy with increasing self-affinity since the collage theorem provides an upper bound for the distortion in terms of the collage error.

[19]Since the performance of an appropriate random codebook asymptotically approaches that of an optimum codebook with increasing codebook size [210], the reference values converge, making the distinction between weak and strong self-affinity progressively less significant.

4.5 Signal Coding Issues

The preceding evaluation of the effectiveness of the domain pool as a codebook did not take into account some of the more practical issues involved in fractal signal coding, the most important of which are the restrictions required to guarantee contractivity of the signal representation. These issues were investigated by performing the fractal coding process (scaling coefficients were subjected to the restriction $|s| \leq s_{\max}$) for an ensemble of 1000 signals (signal length is denoted by n where required) for each value of ρ, which was varied from 0 to 1 in step sizes of 0.05. These simulations allowed an evaluation[20] of the actual performance of fractal coding of AR(1) signals and the dependence of this performance on the parameters of the coding algorithm such as $s_{\max}$. Since repeating the fractal coding process for a large ensemble of signals is, even for relatively small signals, very computationally expensive, a fast nearest neighbour search[21] [8] was used in the search for an optimum domain for each range. Unless otherwise stated, all distortions are measured in terms of the MSE.

4.5.1 Real and collage errors

The relationship between real and collage errors is discussed in Section 3.7. The differences between these errors for three different block sizes are displayed in Figure 4.19, in which the dependence of the shapes of the curves on block size may be explained in terms of the different dependence of ϱ^2 and range block magnitude on block size. In these and all other cases the real error was significantly greater than the collage error, with a decrease in the difference with increasing ρ. The collage error consequently provides a rather optimistic measure of fractal coding performance in the absence of collage optimisation (see Section 3.7.3). Since the collage error may be calculated more rapidly than the real error, it is used as a lower bound for the real error in many subsequent comparisons.

A tentative explanation for the tendency of the collage error to be a lower bound on the real error is presented here. Consider representing signal $\mathbf{x}$ by $\mathbf{x}_T$ where $\mathbf{x}_T = A\mathbf{x}_T + \mathbf{b}$ and A is contractive. The collage signal generated during encoding is $\mathbf{x}_C = A\mathbf{x} + \mathbf{b}$, and the collage and real errors are $\mathbf{e}_C = \mathbf{x}_C - \mathbf{x}$ and $\mathbf{e}_R = \mathbf{x}_T - \mathbf{x}$ respectively. It is easily shown, since $\mathbf{x}_T = (I - A)^{-1}\mathbf{b}$, that

$$\begin{aligned} \mathbf{e}_R &= (I - A)^{-1}\mathbf{e}_C \\ &= \mathbf{e}_C + A\mathbf{e}_C + A^2\mathbf{e}_C + \ldots . \end{aligned}$$

Since A is not necessarily positive definite [53] it is not obvious that A does not "reverse" the direction of $\mathbf{e}_C$ in the series expansion, enabling the magnitude of $\mathbf{e}_R$ to be smaller than that of $\mathbf{e}_C$. A tendency for $\langle A\mathbf{e}_C, \mathbf{e}_C \rangle \geq 0$, together with a similar tendency for higher powers of

[20] Part of the research described in this section has been published previously [202].

[21] Thanks to David Mount for supplying the source code written by himself and Sunil Arya.

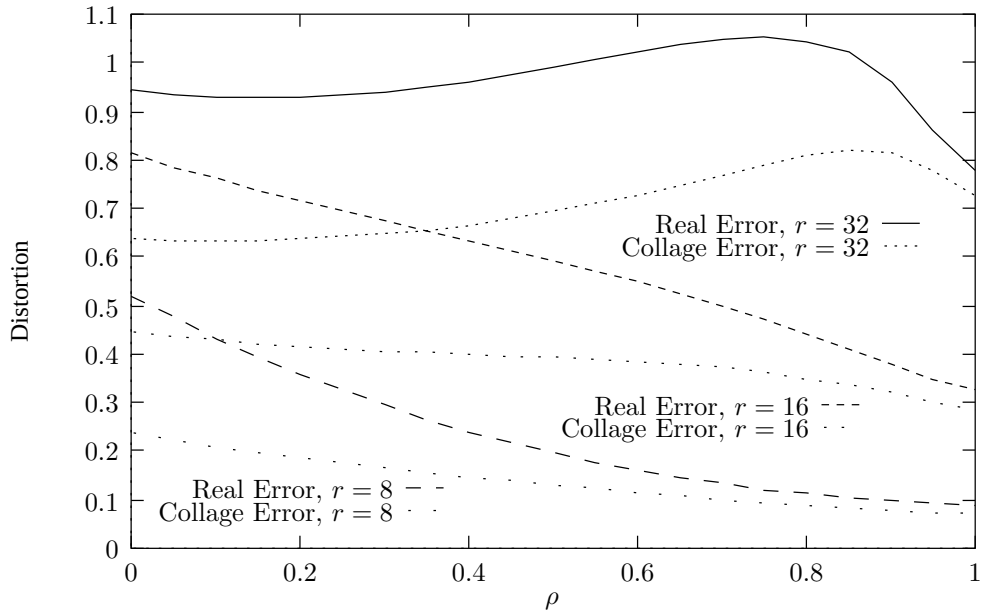


Figure 4.19: Comparison of real and collage errors for different block sizes ($n = 4096$ and $s_{\max} = 1.0$).

A, would be sufficient to expect $\|\mathbf{e}_R\| > \|\mathbf{e}_C\|$.

Consider the action of A on $\mathbf{e}_C$ for a range block at some fixed position. If A maps a spatially distant domain to this range, the inner product between this range block in $\mathbf{e}_C$ and in $A\mathbf{e}_C$ may be positive or negative, and it is reasonable to assume that the average is approximately zero. Alternatively, if the range is mapped from a domain with which it overlaps significantly, the scaling coefficient is likely to be positive (see Section 4.2.1), and therefore so is the appropriate inner product. Since there are a large number of ranges mapped from nearby domains, this effect may be expected to lead to a positive $\langle A\mathbf{e}_C, \mathbf{e}_C \rangle$ on average.

4.5.2 Contractivity

The restriction $s_{\max} < 1$ is often relaxed since it is sufficient, but not necessary, for contractivity [84]. It is clear from Figure 4.20 that the significant differences in collage error for different values of $s_{\max}$ are considerably reduced with increasing signal correlation. Convergence on decoding of signals encoded with an increased $s_{\max}$ was found not to be consistently reliable for all values of ρ, with the minimum ρ for which convergence was reliable (tabulated in Table 4.1) increasing with increasing $s_{\max}$. The real error, in the region of convergence, also decreased with increasing $s_{\max}$. Since the imposition of bounds on the scaling coefficients necessitates excluding as a match for a particular range all domains for which the optimum scaling factor exceeds these bounds, there is an associated reduction in the effectiveness of the domain pool as a codebook.

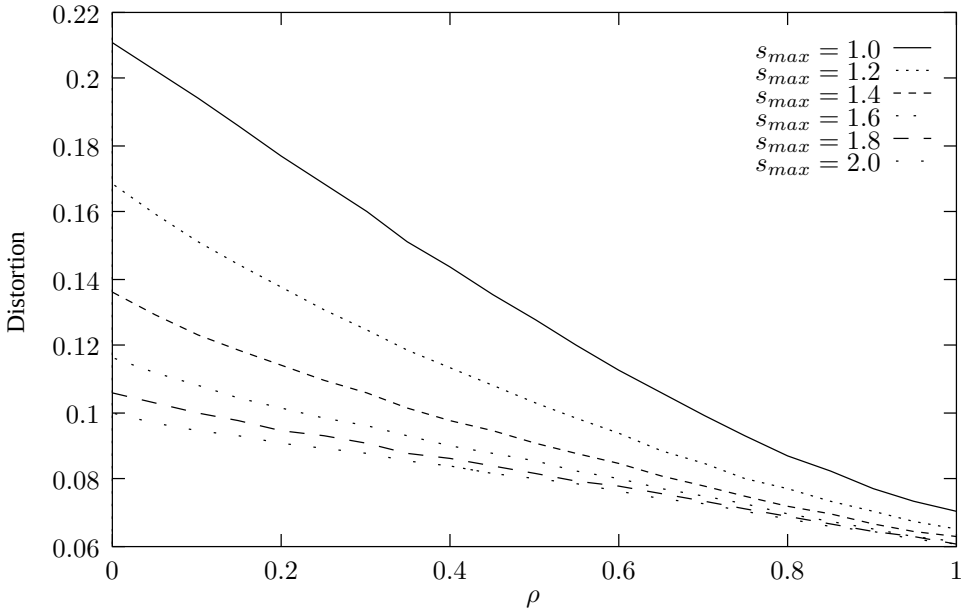


Figure 4.20: Comparison of collage errors for different scaling coefficient bounds ($n = 4096$, $b = 8$ and $\Delta_d = 2$).

Table 4.1: Minimum correlation at which reliable contractivity occurred for different values of $s_{\max}$ (for $n = 4096$ and $b = 8$). Contractivity with $s_{\max} = 2.0$ was not reliable for any signal correlation.

$s_{\max}$	1.0	1.2	1.4	1.6	1.8	2.0
$\rho_{\min}$	0.0	0.3	0.5	0.6	0.8	

4.5.3 Affine transform coefficient distributions

Figure 4.21 illustrates the distributions of scaling values for $s_{\max} = 1$ and a variety of signal correlations. The distribution is highly symmetrical about 0, and the modes decrease with increasing correlation. A comparison with Figure 4.22, which displays the same curves for $s_{\max} = 100$, illustrates the effect of the bound on s. The decreased collage error resulting from increased correlation in the case of $s_{\max} = 1$ (see Figure 4.20) may be explained by the increased proportion of scaling coefficients lying within the required range.

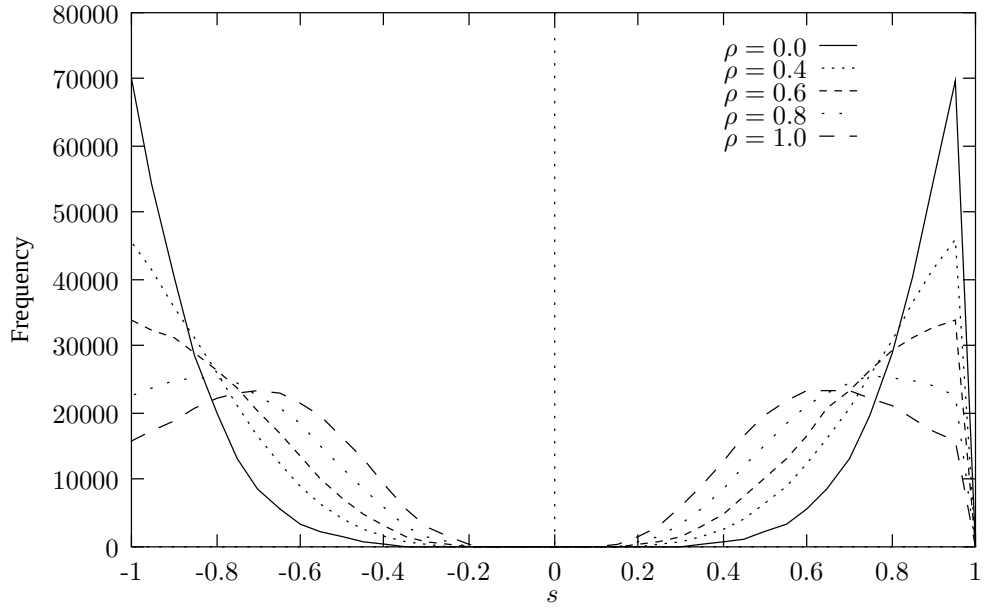


Figure 4.21: Scaling coefficient distributions for different signal correlations ($n = 4096$, $b = 8$, $s_{\max} = 1.0$ and $\Delta_d = 2$).

The offset coefficients have close to zero-mean Gaussian distributions (see Figure 4.23). The increasing variance with increasing signal correlation may be explained in terms of the increasing signal variance σ_X^2 with increasing ρ and constant σ_Z^2.

4.5.4 Spatial offsets between domains and ranges

The distribution of spatial distances between ranges and best matching domains for large domain pools is very close to that resulting from the assumption that best matching domains are uniformly distributed across the signal, with the exception of a significant excess in the immediate neighbourhood of the matching range block. This excess may be attributed to the trough in expected collage error when the domain and range centres coincide, resulting in a raised probability of a match very close to the range. It is clear from Section 4.2.3 that the significance of this effect is rapidly diminished with increasing domain pool size.

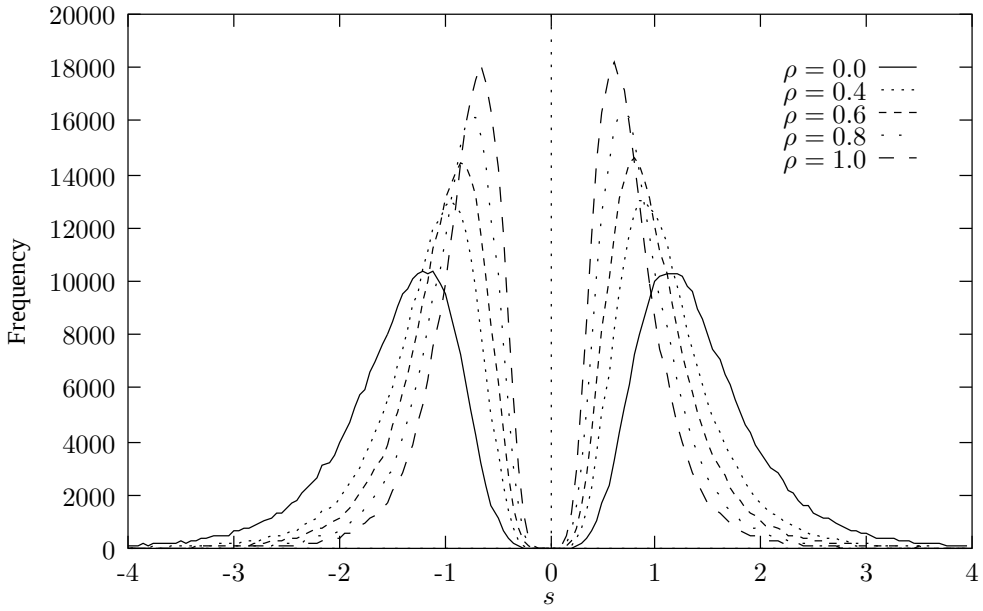


Figure 4.22: Scaling coefficient distributions for different signal correlations ($n = 4096$, $b = 8$, $s_{\max} = 100$ and $\Delta_d = 2$).

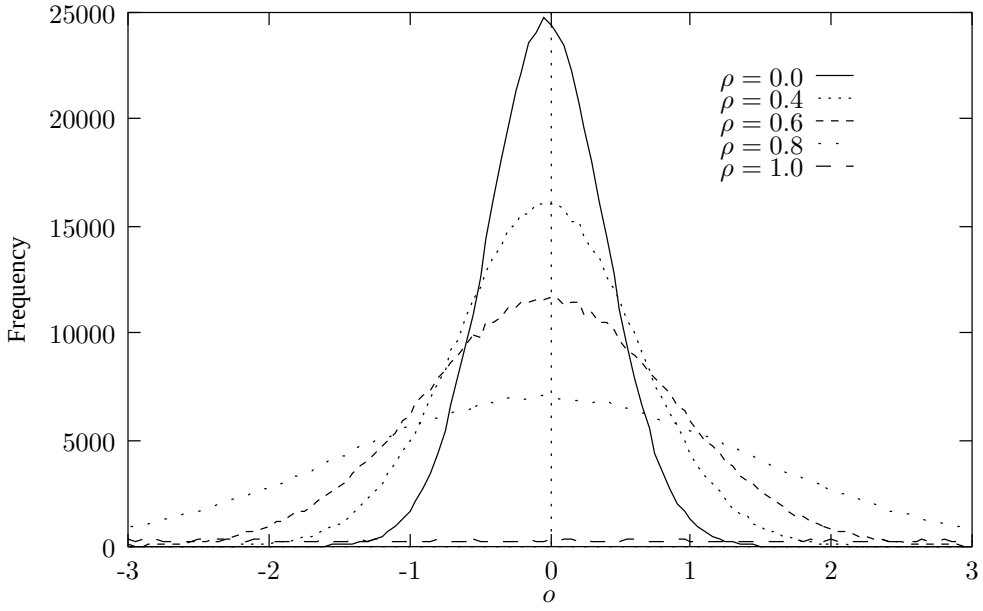


Figure 4.23: Offset coefficient distributions for different signal correlations ($n = 4096$, $b = 8$, $s_{\max} = 1.0$ and $\Delta_d = 2$).

4.5.5 Distortion rate comparisons

A comparison with the theoretically optimum coding performance is possible for Gaussian AR(1) models since the distortion rate function is known. The distortion rate function for a first order Gauss-Markov model is [99, app. D]

$$D(R) = (1-\rho^2)2^{-2R}\sigma_X^2 = \sigma_Z^2 2^{-2R} \tag{4.2}$$

in the small distortion region, where $R \geq \log_2(1+\rho)$. This small distortion requirement is met for all $0 \leq \rho \leq 1$ if $R \geq 1$. The $D(R)$ curves are identical for all values of ρ used here since the variance of the innovations process was fixed at $\sigma_Z^2 = 1$.

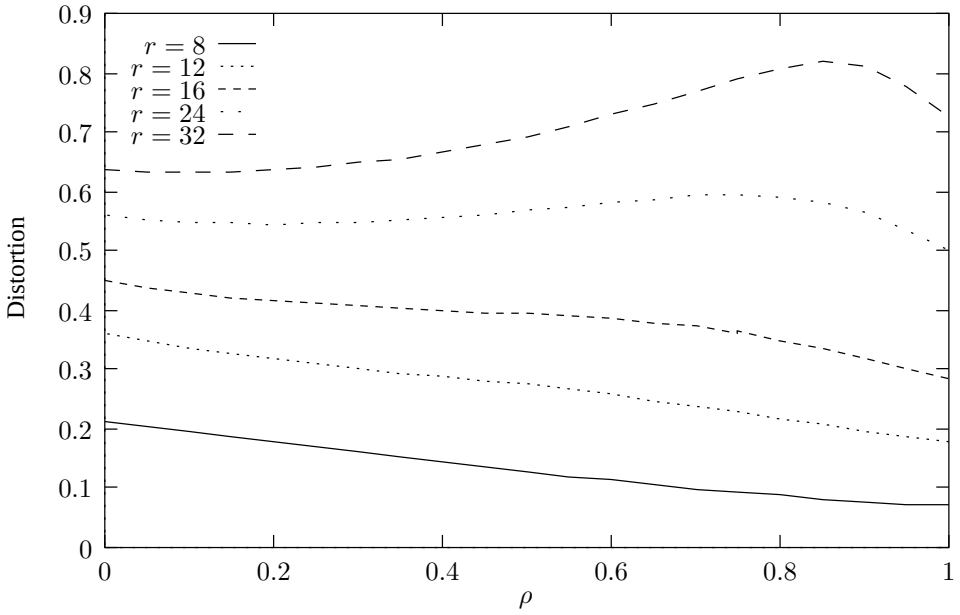


Figure 4.24: Comparison of collage errors for different block sizes ($n = 4096$, $s_{\max} = 1.0$ and $\Delta_d = 2$).

The lowest rate at which the small distortion requirement is satisfied for all ρ is $R = 1$, for which $D = 0.25$. On inspection of Figure 4.24 it is clear that for block sizes larger than 12, the *collage* error with *unquantised* coefficients is considerably greater than the distortion limit at this rate for all ρ. In addition, since domain blocks are chosen from a pool of more than 2000 blocks, insufficient bits are available to code just the domain positions (neglecting the scaling and offset values) for range block sizes less than 12. Reducing the size of the domain pool by considering only neighbouring domain blocks for each range block, or by increasing the domain increment reduces the number of bits required to specify domain position, but simultaneously increases the collage error beyond the distortion limit for unquantised scaling and offset coefficients.

Similar arguments hold for rates below 1 bit/sample. At higher rates the actual distortions

after quantisation of the coefficients were evaluated. An ensemble of signals was generated according to the model for each of several different values of ρ. Each signal was fractal coded, the scaling and offset coefficients were quantised and the signal was reconstructed in order to calculate distortions[22]. The offset was quantised by an optimum Gaussian quantiser for the offset variance, while the absolute value of the scaling was quantised by an optimum Gaussian quantiser for the variance of the scaling absolute value, with an extra bit specifying the sign. Bit allocations for each range block were calculated by multiplying the bit rate by the range block size. Subtracting the number of bits required to specify the domain block position provided the number of bits available for the offset and scaling coefficients. Given R_{so} bits for both coefficients, the scaling bits R_s and offset bits R_o were allocated according to the experimentally derived heuristic

$$R_o = \left\lfloor \frac{2}{3} R_{so} \right\rfloor \qquad R_s = R_{so} - R_o.$$

In each case all other reasonable values of R_s and R_o were compared with the allocation above; other allocations always resulted in a higher distortion, or in a few cases comparable or very slightly lower distortion. Two additional quantisation strategies were tested - to a limited extent, as they required a computationally expensive exhaustive domain search. In the first, identical quantisation to the original scheme was applied, with the exception that the scaling and offset coefficients were quantised *prior* to collage error calculations. The second additional strategy also quantised prior to collage error calculations, but uniform quantisation was applied to the scaling coefficient on the interval $[-s_{\max}, s_{\max}]$. The latter of these strategies resulted in superior performance for $\rho > 0.8$ (with an optimum at $\rho \approx 0.9$), and both avoided the enormous distortions at $\rho = 1.0$ resulting from the original strategy. Other than at $\rho = 1.0$, results were not significantly superior to the original strategy, which was used for all of the results presented here in graph form. Although performance at $\rho = 1.0$, which generates stochastic fractal signals, was massively improved by the alternative quantisation strategies, it remained significantly worse than the minimum distortion at $\rho = 0.9$.

Figure 4.25 illustrates distortions obtained for fractal coding at a rate of $R = 1$, for which the optimum distortion from Equation (4.2) is $D = 0.25$. Bit allocation is as in Table 4.2 (for $\Delta_d = 32$ allocating $R_s = 4$ and $R_o = 5$ resulted in a very small improvement for some values of ρ).

Figure 4.26 illustrates distortions obtained for fractal coding at a rate of $R = 2$, for which the optimum rate from Equation (4.2) is $D = 0.06$. Bit allocation is as in Table 4.3 (for $\Delta_d = 8$ allocating $R_s = 2$ and $R_o = 5$ resulted in a very small improvement for some values of ρ).

The best fractal coding performance in these experiments was obtained for $\rho \approx 0.8$ for the

[22] Although improved performance is achieved by quantising prior to calculating collage errors (see Section 3.5), this was not feasible because of the fast domain search used.

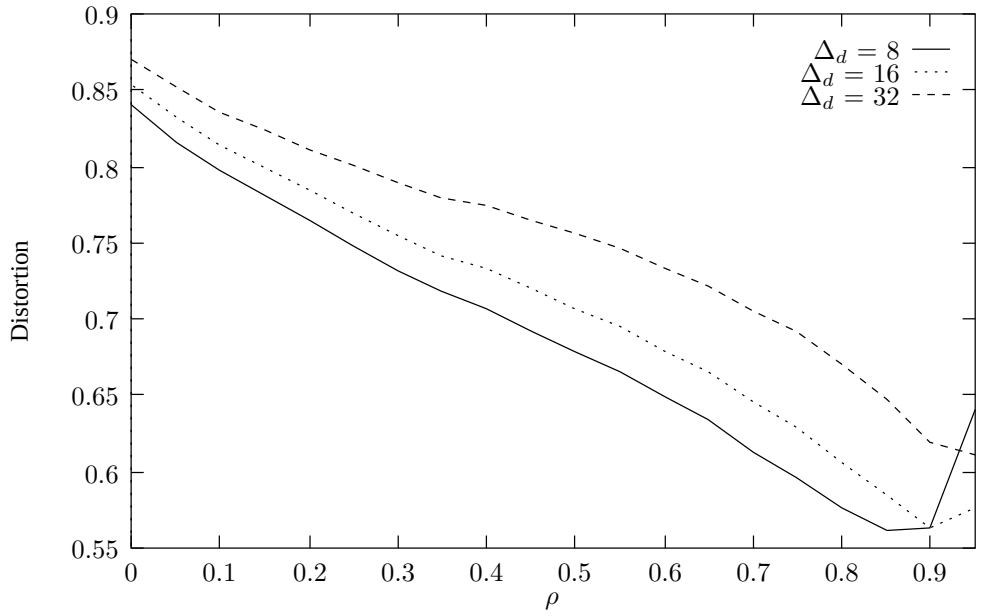


Figure 4.25: Fractal coding distortion at a rate of 1 bit/sample ($n = 4096$, $b = 16$, $s_{\max} = 1.0$). The optimum distortion is 0.25.

Table 4.2: Transform coefficient bit allocations at a rate of 1 bit/sample ($n = 4096$, $b = 16$, $s_{\max} = 1.0$).

	Bits Allocated		
Δ_d	Domain Position	Scaling	Offset
8	9	3	4
16	8	3	5
32	7	3	6

Table 4.3: Transform coefficient bit allocations at a rate of 2 bits/sample ($n = 4096$, $b = 8$, $s_{\max} = 1.0$).

	Bits Allocated		
Δ_d	Domain Position	Scaling	Offset
4	10	2	4
8	9	3	4
16	8	3	5

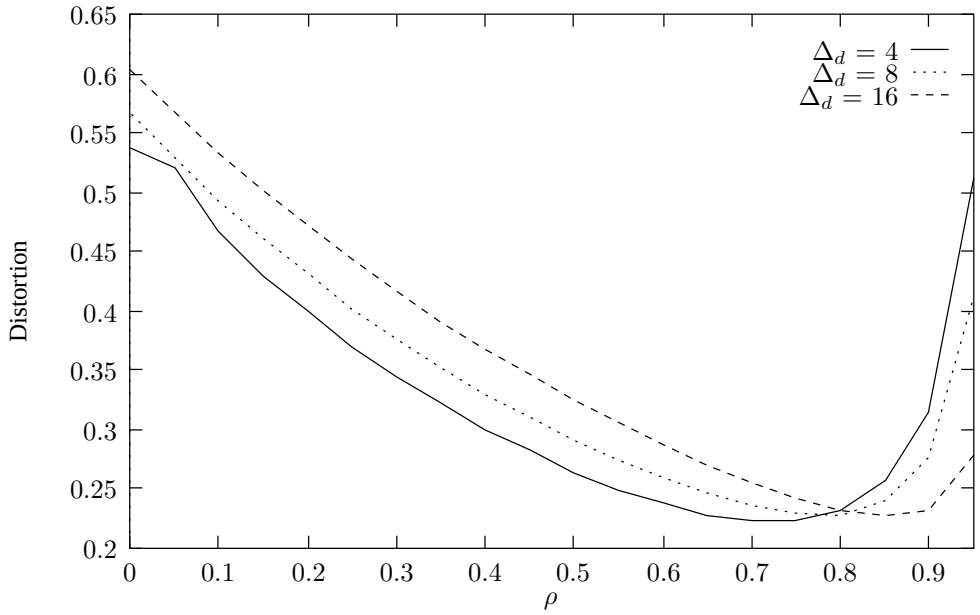


Figure 4.26: Fractal coding distortion at a rate of 2 bits/sample ($n = 4096$, $b = 8$, $s_{\max} = 1.0$). The optimum distortion is 0.06.

original quantisation strategy and $\rho \approx 0.9$ for the alternatives, since the variance of the offset coefficient grows with increasing correlation, making it more expensive to code. In particular, performance was extremely poor (as a result of very high offset variance) for the AR(1) model with $\rho = 1$, which generates stochastic fractal (Brownian motion) signals. In general the actual distortions obtained after quantising the fractal code coefficients were greater than the distortion rate optimum by more than a factor of 2 for the cases investigated, including the stochastic fractal signals generated when $\rho = 1.0$.

4.6 Conclusions

Any conclusions drawn from the results of this chapter are subject to a number of limitations. Most importantly, only a small variety of fractal coding schemes has been examined, which reflects on the precise forms of self-affinity examined. Evaluations of actual signal coding distortion were subject to additional limitations in the form of suboptimal parameter quantisation and the absence of collage optimisation [85]. Finally, these results are clearly only valid for signal generated by Gaussian AR(1) models.

Self-affinity may be considered to be significant when there is statistical dependence between range and domain blocks in a signal resulting in the domain blocks providing a particularly effective match to the range blocks under affine transforms. The existence of such dependence may be determined by a comparison of the domain pool efficiency measure with a suitable reference measure calculated for a random codebook of domains of the same distri-

bution, but independent of the signal from which the ranges are extracted. It is proposed here that signals for which the domain pool efficiency exceeds this reference be termed "weakly self-affine".

Inter-sample correlations in AR(1) models were found to result in a raised similarity between neighbouring domain and range blocks, in the sense that they are considerably more likely to have the same "shape" than two completely independent blocks with the same distribution. The influence of this form of "special relationship" within each signal is however, sufficiently small in codebooks of more than a few code vectors for AR(1) models to be weakly self-affine to a marginal extent only. It is reasonable to expect this borderline weak self-affinity to be a property of a broad range of signal models, since it depends only on the existence of correlations between neighbouring signal samples.

A similar reference value computed for an optimised fixed codebook was proposed in defining "strong self-affinity". The domain pool codebook efficiency was found to be significantly inferior to that of optimised codebooks - AR(1) models are not strongly self-affine. Clearly a considerably stronger relationship is required between domain and range blocks in the same signal for the presence of strong self-affinity. The performance of actual fractal coding of AR(1) signals was found to be significantly suboptimal in distortion rate terms. This may be ascribed to the absence of strong self-affinity in conjunction with the additional practical difficulties in using a domain pool as a codebook. The most significant of these is the contractivity requirement, which further degrades domain pool effectiveness by disallowing a match by any domain with a corresponding scaling coefficient exceeding the imposed bounds.

It is not clear how strongly self-affine a signal model may be without requiring statistics differing significantly from those usually measured for natural images. An indication of this compatibility is provided in the following chapter by the examination of the consequences of exact self-affinity on commonly measured signal statistics.

Chapter 5

Self-Affine Signal Models

An evaluation of the degree of self-affinity of a standard class of signal models is described in the preceding chapter. The opposite approach is adopted in this chapter, in which the consequences of exact self-affinity on signal statistics are examined. This is achieved by defining a class of signal models exhibiting exact self-affinity, and examining the statistical consequences of this construction. These exactly self-affine models may be considered the implicit signal models on which fractal compression is based, in the sense that there are no other models for which fractal coding is more appropriate.

The exactly self-affine models were constructed by considering the fractal code coefficients[1] as suitably distributed random variables driving the iterative decoding process, just as a random *innovations process* is passed through a linear filter to generate an AR model[2] [99, ch. 2]. There is considerable freedom in constructing these models, both in terms of the structural constraints of the fractal coding scheme used in their construction as well as in the statistical distributions of the associated fractal representation coefficients. The former reflect the particular form of self-affinity desired, and are restricted here to the same range partitions and block transforms evaluated in the previous chapter. As before, only one dimensional signal models are considered in this chapter, both to limit the model complexity and to reduce the computational requirements of Monte Carlo simulations where these are necessary. The choice of appropriate probability distributions for the transform coefficients is somewhat more arbitrary, but is guided as far as possible by practices prevalent in the literature, as well as by practical requirements.

5.1 A Generalised Lattice Vector Quantisation Interpretation

Despite the frequent comparison of the domain pool in fractal coding with the codebook of Mean Removed SGVQ, a theoretical analysis of fractal coding based on asymptotic VQ

[1] The set of domain block positions and scaling and offset coefficients for each range block.

[2] Since every signal decoded from a fractal representation is exactly self-affine, any signal which is *not* could not have been generated by an exactly self-affine model. In contrast, a signal may only be *unlikely* to be the product of a particular AR model.

results [72] [73] [210] is impossible since the domain pool "codebook" is not fixed for all signals. Although fractal coding is clearly unable to outperform optimum VQ on vectors of the size of each signal (i.e. in the large block limit), it is difficult to compare with VQ based on range block sized vectors[3]. A viable analysis in terms of VQ is achieved by a shift of perspective from VQ operating at the range block level to VQ operating on the signal as a whole, in which case the fractal coding mechanism may be viewed as a form of generalised Lattice[4] VQ on the signal as a whole (i.e. in the large block limit). Consider a metric space (X, d) with a set of parametrised[5] transforms

$$\mathcal{T} = \{T_{\mathbf{a}} : X \to X \mid \mathbf{a} \in \mathcal{A}\}$$

where the set of allowed parameters $\mathcal{A}$ is usually chosen such that $T_{\mathbf{a}}$ is eventually contractive for all $\mathbf{a} \in \mathcal{A}$. A set of code vectors may then be defined as

$$\mathcal{C} = \{\mathbf{x} \in X \mid \exists T \in \mathcal{T} \;\; T\mathbf{x} = \mathbf{x}\}$$

which induces a nearest neighbour quantisation[6]

$$Q(\mathbf{x}) = \min_{\mathbf{x}' \in \mathcal{C}}{}^{-1} d(\mathbf{x}, \mathbf{x}').$$

Even if $\mathcal{A}$ is a continuous set, the structural constraints [85] induce a partial quantisation on X, which becomes a full quantisation after quantisation of $\mathcal{A}$ (see Figure 5.1). The final generalised lattice is thus determined by the structural constraints in conjunction with the selected transform parameter quantisation.

The results of Cheng and Zhu [35] (see Section 3.9.5) imply that the lattice points may be made arbitrarily dense in any region by using sufficiently small range blocks. It is certainly *not* clear however, why the lattice point densities would be arranged so as to form an efficient codebook for compression purposes. In the high resolution case [73, pp. 338-340] optimal reconstruction vectors for a source $f_{\mathbf{X}}(\mathbf{x})$ of vectors of dimension k have distribution

$$f_{\mathbf{X}}^{\frac{k}{k+2}}(\mathbf{x})$$

[3]There is, however, clearly no advantage to the "virtual codebook", when compared with VQ at the same block size, for mutually *independent* range blocks, since in this case each range block has no "special relationship" with the rest of the signal from which it is extracted.

[4]The structure described here is not strictly speaking a lattice, but is similar in that the reconstruction vectors are determined by a mathematical construction rather than simply as a list of vectors.

[5]This is an exception to the usual usage in this dissertation, where the form of a particular fractal coding scheme is described by its "parameters", whereas the actual values coded are referred to as the representation "coefficients".

[6]The function $\min^{-1}$ is often written as $\arg\min$, and denotes the set element minimising the set of values to which is is applied, rather than the minimum value of that set, which is denoted by min.

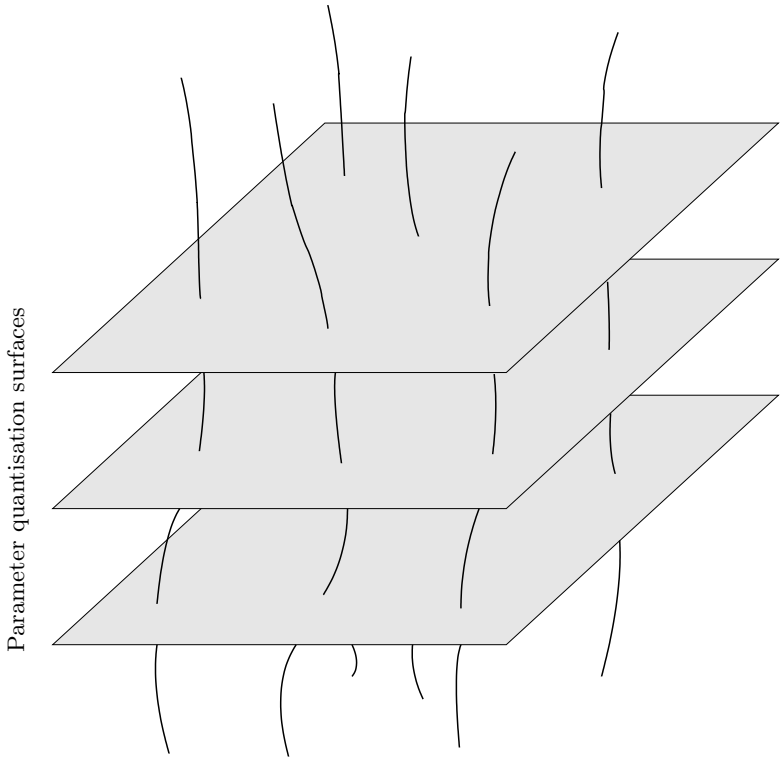


Figure 5.1: Abstract diagram of the generalised Lattice VQ view of fractal coding. Each individual "thread" corresponds to a different set of domain block choices for the range blocks, while the "threads" themselves are produced by continuous variation of the scaling and offset coefficients of the block transforms. The "threads" represent the partial quantisation due to the structural constraints, with full quantisation imposed by quantisation of the parameters varying along the length of the "threads". The diagram is highly abstract in that the "threads" represent surfaces in multidimensional space.

for the MSE distortion measure [73, pg. 471]. Therefore, for large k, the density of reconstruction points in an optimal codebook should correspond approximately to the probability density of the source. One may consequently evaluate the effectiveness of the lattice codebook by comparing its structure with that of the probability density of the source it is intended to quantise. There is a strong similarity between such an evaluation and the evaluation of the properties of the exactly self-affine models performed here, since the probability density of a self-affine model corresponds to the lattice density of a generalised lattice codebook in the limit of very high resolution quantisation of the transform parameters.

5.2 Autocorrelation

The autocorrelation (or autocovariance) is the most widely used statistic in characterising random signals. It is particularly important in transform coding, which specifically depends on the correlation properties, and in the case of a multivariate Gaussian it completely characterises the signal model. In addition, an upper bound on the rate distortion function is available in terms of the autocorrelation [99, app. D], and for a stationary model the expected signal spectral content (the power spectral density) is the Fourier transform of the autocorrelation.

An analytic estimate of the model autocorrelation may be obtained by utilising the closed form series expansion of Equation (3.2) for random vector $\mathbf{X}$ in terms of random matrix A and random vector[7] $\mathbf{b}$. The expansion results in

$$\mathbf{X}\mathbf{X}^T = (I + A + A^2 + \ldots)\mathbf{b}\mathbf{b}^T(I + A^T + (A^2)^T + \ldots),$$

implying

$$\begin{aligned} R_{\mathbf{X}} &= E[\mathbf{X}\mathbf{X}^T] \\ &= E[\mathbf{b}\mathbf{b}^T] + \\ &\quad E[A\mathbf{b}\mathbf{b}^T] + E[\mathbf{b}\mathbf{b}^T A^T] + \\ &\quad E[A^2\mathbf{b}\mathbf{b}^T] + E[A\mathbf{b}\mathbf{b}^T A^T] + E[\mathbf{b}\mathbf{b}^T(A^2)^T] + \\ &\quad E[A^3\mathbf{b}\mathbf{b}^T] + E[A^2\mathbf{b}\mathbf{b}^T A^T] + E[A\mathbf{b}\mathbf{b}^T(A^2)^T] + E[\mathbf{b}\mathbf{b}^T(A^3)^T] + \\ &\quad E[A^4\mathbf{b}\mathbf{b}^T] + \ldots \end{aligned} \tag{5.1}$$

where the terms are arranged[8] according to increasing order in A.

[7] The bold lower case symbol for the deterministic vector $\mathbf{b}$ is retained for the corresponding random vector for notational simplicity.

[8] Truncating the expansion in Equation (3.2) prior to calculating the autocorrelation results in a different ordering.

All self-affine models evaluated in this chapter are based on fixed block size partitions with range blocks of width b and domain blocks of width $2b$. A signal of n samples therefore has $N_r = n/b$ range blocks and $N_d = \lfloor (n-2b)/\Delta_d \rfloor + 1$ domain blocks, where Δ_d is the domain block increment. The index of the domain block selected for range block i (where $0 \leq i < N_r$) is denoted by p_i (where $0 \leq p_i < N_d$), and the scaling and offset coefficients for range block i are s_i and o_i respectively.

Unless otherwise stated, all scaling and offset coefficients are considered to be independent random variables (although this is a considerable simplification, it is an assumption commonly made for the quantisation of fractal code coefficients). In most of the experiments the offset coefficients were assigned a Gaussian distribution, since this is at least approximately the measured distribution for real signals. The scaling coefficients were assigned a uniform distribution on the interval (-1,1) since this choice avoided convergence problems, and uniform quantisation, which may in a weak sense be considered an implicit assumption of uniformity, is often applied to the scaling coefficients [65, ch. 3]. Monte Carlo simulation[9] was utilised where analytic derivation was intractable.

5.2.1 Analytic derivation of the autocorrelation

The $n \times n$ matrix A of Equation (3.2) may be expressed as $A = ((SP) \otimes I_b)D$, where D (an $bN_d \times bN_r$ matrix) extracts each domain from the signal, spatially contracts each of them, and then concatenates them into a single vector, I_b is the $b \times b$ identity matrix, P (an $N_r \times N_d$ matrix) is a domain selection matrix (since domain and range indices begin at zero, matrix entries are indexed similarly, the top left entry of any matrix having zero as the row and column index)

$$P = \begin{pmatrix} \delta_{0,p_0} & \delta_{1,p_0} & \cdots & \delta_{N_d-1,p_0} \\ \delta_{0,p_1} & \delta_{1,p_1} & \cdots & \delta_{N_d-1,p_1} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{0,p_{N_r-1}} & \delta_{1,p_{N_r-1}} & \cdots & \delta_{N_d-1,p_{N_r-1}} \end{pmatrix},$$

and S ($N_r \times N_r$) is a diagonal scaling matrix assigning a scaling coefficient to the selected domain for each range

$$S = \begin{pmatrix} s_0 & 0 & \cdots & 0 \\ 0 & s_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_{N_r-1} \end{pmatrix}.$$

[9] Random values with a Gaussian distribution were generated as in Chapter 4. Uniformly distributed values and discrete random numbers were generated using the same C++ class library.

The $bN_r \times bN_d$ matrix $((SP) \otimes I_b)$ may be expressed in terms of submatrix I_b as

$$((SP) \otimes I_b) = \begin{pmatrix} \delta_{0,p_0} s_0 I_b & \delta_{1,p_0} s_0 I_b & \dots & \delta_{N_d-1,p_0} s_0 I_b \\ \delta_{0,p_1} s_1 I_b & \delta_{1,p_1} s_1 I_b & \dots & \delta_{N_d-1,p_1} s_1 I_b \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{0,p_{N_r-1}} s_{N_r-1} I_b & \delta_{1,p_{N_r-1}} s_{N_r-1} I_b & \dots & \delta_{N_d-1,p_{N_r-1}} s_{N_r-1} I_b \end{pmatrix}. \quad (5.2)$$

The structure of D is determined by the fractal coding scheme parameters, such as domain increment, block size and the use of mean removal. As an example

$$D = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & \dots & 0 & 0 & 0 & \dots \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & \dots & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \dots \\ 0 & 0 & 0 & 0 & \dots & \frac{1}{2} & \frac{1}{2} & 0 & \dots \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & \dots & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \dots & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & \frac{1}{2} & \frac{1}{2} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

for $\Delta_d = 1$ and spatial contraction by averaging, with no mean removal.

The offset $\mathbf{b}$ may be written as

$$\mathbf{b} = \mathbf{o} \otimes \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \mathbf{o} \otimes \mathbf{1}$$

where $\mathbf{1}$ is a column b-vector with unit entries.

Expectation Terms

Evaluation of the individual terms of Equation (5.1) was simplified by the assumptions

$\forall i, j \in \{0, 1, \dots, N_r - 1\}$:

$\mu_{s_i} = \mu_s = 0 \quad \sigma_{s_i} = \sigma_s \quad E[s_i s_j] = \delta_{i,j}\sigma_s$

$\mu_{o_i} = \mu_o = 0 \quad \sigma_{o_i} = \sigma_o \quad E[o_i o_j] = \delta_{i,j}\sigma_o$

$E[s_i o_j] = 0$

and p_i was assumed to take on all values in $\{0, 1, \dots, N_d - 1\}$ with equal probability.

The individual terms in Equation (5.1) may be evaluated by utilising the expansion of A in terms of Equation (5.2), and the following results [82, pp. 243-244]:

1. If $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ then $\mathbf{x}\mathbf{y}^T = \mathbf{x} \otimes \mathbf{y}^T$
2. If A and B are arbitrary matrices then $(A \otimes B)^T = A^T \otimes B^T$
3. If products AC and BD of matrices A, B, C, and D exist then $(A \otimes B)(C \otimes D) = AC \otimes BD$

Order 0 term

The initial term

$$\begin{aligned}
\mathbf{b}\mathbf{b}^T &= \left(\mathbf{o} \otimes \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \right) \left(\mathbf{o}^T \otimes (1, 1, \ldots, 1) \right) \\
&= \mathbf{o}\mathbf{o}^T \otimes \begin{pmatrix} 1 & 1 & \ldots \\ 1 & 1 & \ldots \\ \vdots & \vdots & \ddots \end{pmatrix} \\
&= \mathbf{o}\mathbf{o}^T \otimes \mathbf{1}_{bb},
\end{aligned}$$

and therefore

$$\begin{aligned}
E[\mathbf{b}\mathbf{b}^T] &= E[\mathbf{o}\mathbf{o}^T] \otimes \mathbf{1}_{bb} \\
&= \sigma_o^2 \begin{pmatrix} \mathbf{1}_{bb} & \mathbf{0}_{bb} & \ldots \\ \mathbf{0}_{bb} & \mathbf{1}_{bb} & \ldots \\ \vdots & \vdots & \ddots \end{pmatrix},
\end{aligned}$$

where

$$\mathbf{0}_{bb} = \begin{pmatrix} 0 & 0 & \ldots \\ 0 & 0 & \ldots \\ \vdots & \vdots & \ddots \end{pmatrix} \text{ and } \mathbf{1}_{bb} = \begin{pmatrix} 1 & 1 & \ldots \\ 1 & 1 & \ldots \\ \vdots & \vdots & \ddots \end{pmatrix}.$$

Order 1 terms

The matrix A may be expressed as

$$\begin{aligned} A &= ((SP)\otimes I_b)D \\ &= \begin{pmatrix} \delta_{0,p_0}s_0I_b & \delta_{1,p_0}s_0I_b & \delta_{2,p_0}s_0I_b & \dots \\ \delta_{0,p_1}s_1I_b & \delta_{1,p_1}s_1I_b & \delta_{2,p_1}s_1I_b & \dots \\ \delta_{0,p_2}s_2I_b & \delta_{1,p_2}s_2I_b & \delta_{2,p_2}s_2I_b & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} D_{00} & D_{01} & D_{02} & \dots \\ D_{10} & D_{11} & D_{12} & \dots \\ D_{20} & D_{21} & D_{22} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \\ &= \begin{pmatrix} s_0D_{p_00} & s_0D_{p_01} & s_0D_{p_02} & \dots \\ s_1D_{p_10} & s_1D_{p_11} & s_1D_{p_12} & \dots \\ s_2D_{p_20} & s_2D_{p_21} & s_2D_{p_22} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \end{aligned}$$

where the D_{ij} are $b\times b$ submatrices of D with upper left entry having coordinates (bi, bj) in D. The expectation of $A\mathbf{b}\mathbf{b}^T$ is consequently

$$\begin{aligned} E[A\mathbf{b}\mathbf{b}^T] &= E[A]E[\mathbf{b}\mathbf{b}^T] \\ &= \mu_s\sigma_o^2 \begin{pmatrix} E[D_{p_00}] & E[D_{p_01}] & \dots \\ E[D_{p_10}] & E[D_{p_11}] & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \mathbf{1}_{bb} & \mathbf{0}_{bb} & \dots \\ \mathbf{0}_{bb} & \mathbf{1}_{bb} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \\ &= \mathbf{0}_{nn}, \end{aligned}$$

since $\mu_s = 0$. A similar derivation results in $E[\mathbf{b}\mathbf{b}^TA^T] = \mathbf{0}_{nn}$.

Order 2 terms

The square of the matrix A may be expressed as

$$\begin{aligned}
A^2 &= ((SP)\otimes I_b)D((SP)\otimes I_b)D \\
&= \begin{pmatrix} s_0D_{p_00} & s_0D_{p_01} & s_0D_{p_02} & \dots \\ s_1D_{p_10} & s_1D_{p_11} & s_1D_{p_12} & \dots \\ s_2D_{p_20} & s_2D_{p_21} & s_2D_{p_22} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} s_0D_{p_00} & s_0D_{p_01} & s_0D_{p_02} & \dots \\ s_1D_{p_10} & s_1D_{p_11} & s_1D_{p_12} & \dots \\ s_2D_{p_20} & s_2D_{p_21} & s_2D_{p_22} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \\
&= \begin{pmatrix} s_0\sum_{k=0}^{N_r-1} s_kD_{p_0k}D_{p_k0} & s_0\sum_{k=0}^{N_r-1} s_kD_{p_0k}D_{p_k1} & s_0\sum_{k=0}^{N_r-1} s_kD_{p_0k}D_{p_k2} & \dots \\ s_1\sum_{k=0}^{N_r-1} s_kD_{p_1k}D_{p_k0} & s_1\sum_{k=0}^{N_r-1} s_kD_{p_1k}D_{p_k1} & s_1\sum_{k=0}^{N_r-1} s_kD_{p_1k}D_{p_k2} & \dots \\ s_2\sum_{k=0}^{N_r-1} s_kD_{p_2k}D_{p_k0} & s_2\sum_{k=0}^{N_r-1} s_kD_{p_2k}D_{p_k1} & s_2\sum_{k=0}^{N_r-1} s_kD_{p_2k}D_{p_k2} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.
\end{aligned}$$

Since $\mu_s = 0$ and $E[s_is_j] = \delta_{i,j}\sigma_s^2$,

$$\begin{aligned}
E[A^2] &= \begin{pmatrix} E[s_0^2]E[D_{p_00}D_{p_00}] & E[s_0^2]E[D_{p_00}D_{p_01}] & E[s_0^2]E[D_{p_00}D_{p_02}] & \dots \\ E[s_1^2]E[D_{p_11}D_{p_10}] & E[s_1^2]E[D_{p_11}D_{p_11}] & E[s_1^2]E[D_{p_11}D_{p_12}] & \dots \\ E[s_2^2]E[D_{p_22}D_{p_20}] & E[s_2^2]E[D_{p_22}D_{p_21}] & E[s_2^2]E[D_{p_22}D_{p_22}] & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \\
&= \sigma_s^2 \begin{pmatrix} E[D_{p_00}D_{p_00}] & E[D_{p_00}D_{p_01}] & E[D_{p_00}D_{p_02}] & \dots \\ E[D_{p_11}D_{p_10}] & E[D_{p_11}D_{p_11}] & E[D_{p_11}D_{p_12}] & \dots \\ E[D_{p_22}D_{p_20}] & E[D_{p_22}D_{p_21}] & E[D_{p_22}D_{p_22}] & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix},
\end{aligned}$$

and

$$\begin{aligned}
E[A^2\mathbf{b}\mathbf{b}^T] &= E[A^2]E[\mathbf{b}\mathbf{b}^T] \\
&= \sigma_s^2\sigma_o^2 \begin{pmatrix} E[D_{p_00}D_{p_00}] & E[D_{p_00}D_{p_01}] & \dots \\ E[D_{p_11}D_{p_10}] & E[D_{p_11}D_{p_11}] & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \mathbf{1}_{bb} & \mathbf{0}_{bb} & \dots \\ \mathbf{0}_{bb} & \mathbf{1}_{bb} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \\
&= \frac{\sigma_s^2\sigma_o^2}{N_d} \begin{pmatrix} \sum_{i=0}^{N_d-1} D_{i0}^2\mathbf{1}_{bb} & \sum_{i=0}^{N_d-1} D_{i0}D_{i1}\mathbf{1}_{bb} & \dots \\ \sum_{i=0}^{N_d-1} D_{i1}D_{i0}\mathbf{1}_{bb} & \sum_{i=0}^{N_d-1} D_{i1}^2\mathbf{1}_{bb} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}.
\end{aligned}$$

Similarly

$$\begin{aligned} E[\mathbf{b}\mathbf{b}^T(A^2)^T] &= \frac{\sigma_s^2\sigma_o^2}{N_d}\begin{pmatrix} \sum_{i=0}^{N_d-1}\mathbf{1}_{bb}(D_{i0}^T)^2 & \sum_{i=0}^{N_d-1}\mathbf{1}_{bb}D_{i0}^TD_{i1}^T & \dots \\ \sum_{i=0}^{N_d-1}\mathbf{1}_{bb}D_{i1}^TD_{i0}^T & \sum_{i=0}^{N_d-1}\mathbf{1}_{bb}(D_{i1}^T)^2 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \\ &= E[A^2\mathbf{b}\mathbf{b}^T]^T. \end{aligned}$$

The following expression, in terms of an arbitrary $bN_d \times bN_d$ matrix Q, is helpful in determining $E[A\mathbf{b}\mathbf{b}^TA^T]$:

$$((SP)\otimes I_b)Q((SP)^T\otimes I_b^T) =$$

$$\begin{pmatrix} \delta_{0,p_0}s_0I_b & \delta_{1,p_0}s_0I_b & \dots \\ \delta_{0,p_1}s_1I_b & \delta_{1,p_1}s_1I_b & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}\begin{pmatrix} Q_{00} & Q_{01}\dots & \\ Q_{10} & Q_{11}\dots & \\ \vdots & \vdots & \ddots \end{pmatrix}\begin{pmatrix} \delta_{0,p_0}s_0I_b & \delta_{0,p_1}s_1I_b & \dots \\ \delta_{1,p_0}s_0I_b & \delta_{1,p_1}s_1I_b & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

$$= \begin{pmatrix} s_0^2Q_{p_0p_0} & s_0s_1Q_{p_0p_1} & \dots \\ s_1s_0Q_{p_1p_0} & s_1^2Q_{p_1p_1} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}. \tag{5.3}$$

Setting

$$\begin{aligned} Q &= D\mathbf{b}\mathbf{b}^TD^T \\ &= \begin{pmatrix} D_{00} & D_{01} & \dots \\ D_{10} & D_{11} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}\begin{pmatrix} o_0^2\mathbf{1}_{bb} & o_0o_1\mathbf{1}_{bb}\dots & \\ o_1o_0\mathbf{1}_{bb} & o_1^2\mathbf{1}_{bb}\dots & \\ \vdots & \vdots & \ddots \end{pmatrix}\begin{pmatrix} D_{00}^T & D_{10}^T & \dots \\ D_{01}^T & D_{11}^T & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \\ &= \begin{pmatrix} \sum_{j=0}^{N_r-1}\sum_{i=0}^{N_r-1}o_jo_iD_{0i}\mathbf{1}_{bb}D_{0j}^T & \sum_{j=0}^{N_r-1}\sum_{i=0}^{N_r-1}o_jo_iD_{0i}\mathbf{1}_{bb}D_{1j}^T & \dots \\ \sum_{j=0}^{N_r-1}\sum_{i=0}^{N_r-1}o_jo_iD_{1i}\mathbf{1}_{bb}D_{0j}^T & \sum_{j=0}^{N_r-1}\sum_{i=0}^{N_r-1}o_jo_iD_{1i}\mathbf{1}_{bb}D_{1j}^T & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}, \end{aligned}$$

and substituting into Equation (5.3) gives

$$E[A\mathbf{b}\mathbf{b}^TA^T] = \frac{\sigma_s^2\sigma_o^2}{N_d}\begin{pmatrix} \sum_{j=0}^{N_d-1}\sum_{i=0}^{N_r-1}D_{ji}\mathbf{1}_{bb}D_{ji}^T & \mathbf{0}_{bb} & \dots \\ \mathbf{0}_{bb} & \sum_{j=0}^{N_d-1}\sum_{i=0}^{N_r-1}D_{ji}\mathbf{1}_{bb}D_{ji}^T & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}.$$

Images representing the contributions of the individual terms for model parameters as in Table 5.1 are displayed in Figure 5.2. Note that the inter-block correlations resulting from the term $E[A\mathbf{b}\mathbf{b}^T] + E[\mathbf{b}\mathbf{b}^TA^T]$ are sufficiently small in magnitude to be undetectable in the sum displayed in Figure 5.3. Despite the truncation of expansion 5.1 in this evaluation, it is clear

from a comparison of Figures 5.3 and 5.4 that an accurate estimate of the autocorrelation is obtained. All further evaluations were based on Monte Carlo simulations, which are more easily applied to a wider range of parameters.

Table 5.1: Model parameters used in a number of examples. The affine transform does not remove the mean of the domain block and all domains are selected with equal probability. The scaling coefficients are uniformly distributed on the interval $(-1, 1)$, while the offset coefficients have a Gaussian distribution of zero mean and unit variance.

n	b	Δ_d	s pdf	o pdf
128	16	2	U(-1,1)	N(0,1)

5.2.2 Estimation of the autocorrelation by Monte Carlo simulation

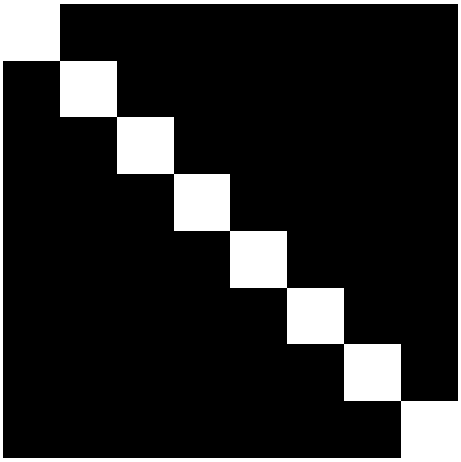
An ensemble of exactly self-affine signals was generated by applying the iterative decoding process to a corresponding ensemble of random domain to range transforms. Scaling and offset coefficients were assigned uniform and Gaussian distributions respectively, and domain positions were selected with equal probability from those available.

Figure 5.4 illustrates the low correlation between blocks[10], as well as the decreasing correlation away from the diagonal within each block. Although the autocorrelation is not stationary, one may measure the autocorrelation decay with increasing distance from the diagonal within the visible "blocks" in the autocorrelation of the whole signal. In this case fitting the expected decay function for an AR(1) model to the initial part of this curve resulted in a best fit at $\rho = 0.92$ for the model parameters as in Table 5.1, a value which is consistent with the findings of Chapter 4, where the best performance for fractal coding was found to occur at a similar correlation.

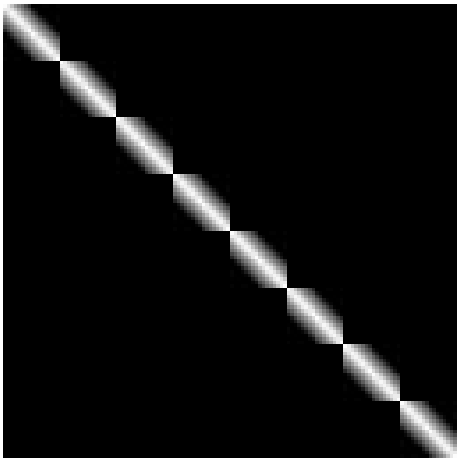
These gross features correspond approximately to the usual autocorrelation measurements for images [37, ch. 2], and the abrupt decay at block boundaries is a result of independent coding of range blocks[11]. Adjustment of the fine structure of the autocorrelation surface (in particular the rate of decay with increasing distance from the diagonal) by varying the distributions of the fractal representation coefficients and the parameters of the structural constraints proved to be rather difficult without introducing clearly undesirable features. Scaling of the surface was possible by varying σ_o^2, but practical values for σ_s^2 were restricted by the contractivity requirement. Very long distance correlations between blocks resulted from non-zero μ_o or μ_s, as well as from allowing correlations between the individual offsets o_i. The autocorrelation surface became rather irregular with the introduction of correlations between the scaling coefficients s_i. Increased irregularities also resulted from an increase in domain increment Δ_d, while the introduction of mean removal in the block transforms (see

[10] Self-affinity is therefore not necessarily associated with significant long-range correlation as has been conjectured [87].

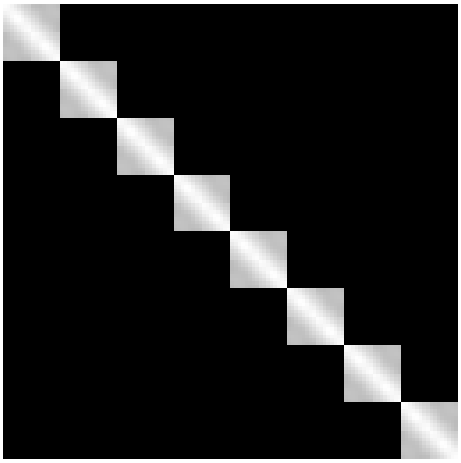
[11] A similar blocked autocorrelation would be expected for signals subsequent to blocked transform coding.



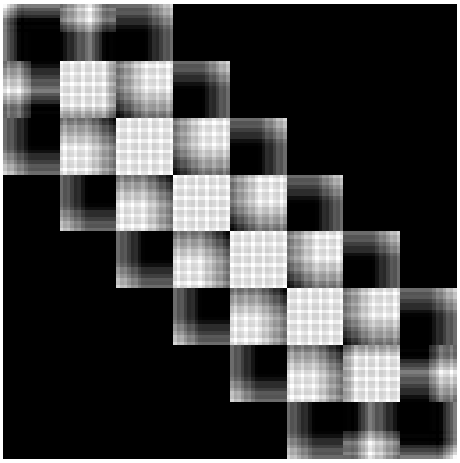
$E[\mathbf{b}\mathbf{b}^T]$



$E[A\mathbf{b}\mathbf{b}^T A^T]$



$E[\mathbf{b}\mathbf{b}^T] + E[A\mathbf{b}\mathbf{b}^T A^T]$



$E[A\mathbf{b}\mathbf{b}^T] + E[\mathbf{b}\mathbf{b}^T A^T]$

Figure 5.2: Individual terms in expansion of autocorrelation of exactly self-affine model (parameters as in Table 5.1). Correlation increases from dark to light.

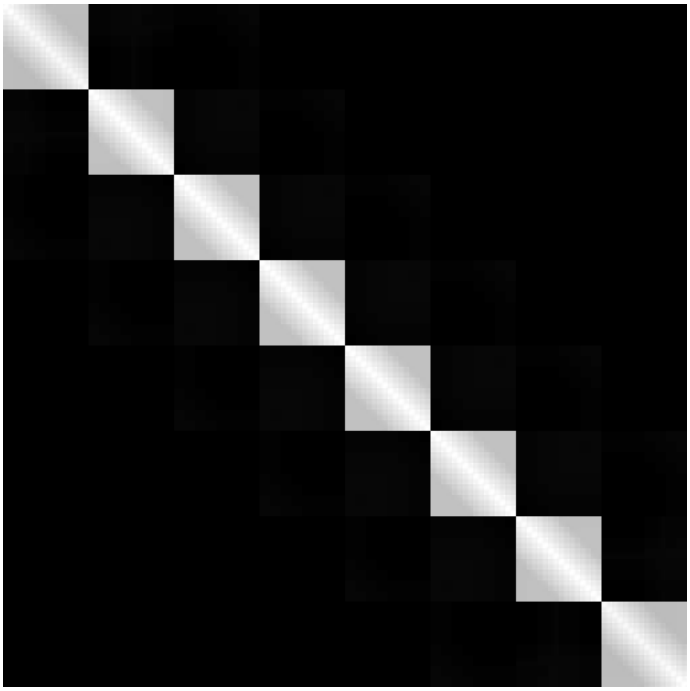


Figure 5.3: Analytic estimate of autocorrelation of exactly self-affine model (parameters as in Table 5.1).

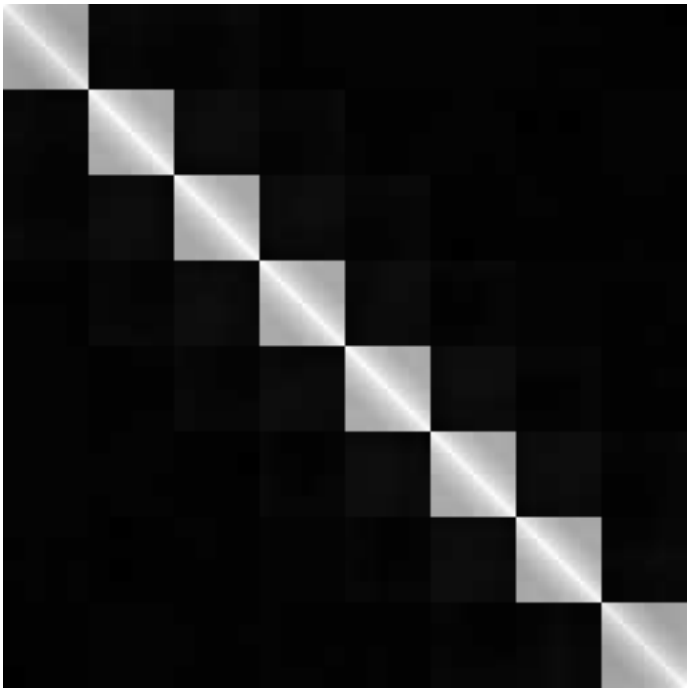


Figure 5.4: Autocorrelation of exactly self-affine model (parameters as in Table 5.1).

Section 3.2.2) did not result in significant differences. In general the form of matrix D plays an important role in determining the autocorrelation, as is clear from Section 5.2.1.

5.3 A Decorrelating Transform

It was shown in Section 3.9 that a relatively simple wavelet transform domain description is possible for a restricted set of fractal coding schemes. Since it is reasonable to expect this simplifying effect to extend to the statistical properties of the exactly self-affine signal models, these properties were examined in the wavelet transform domain. It will be shown in subsequent sections that the exactly self-affine models may also be *constructed* in the wavelet domain, as opposed to merely examining the wavelet domain properties of signals generated in the spatial domain.

5.3.1 Autocorrelation

The Haar transform domain autocorrelation for fractal representation coefficient distributions as in Table 5.1 is displayed in Figure 5.5 (gamma correction[12] has been applied to enhance visibility). The small off-diagonal correlations visible correspond mainly to correlations between detail coefficients at the same position and different resolutions[13]. The decorrelation efficiency [37, ch. 3] [140] of a transform T is

$$\eta_C = 1 - \frac{\sum_{i=0}^{n-1} \sum_{j>i}^{n-1} C_Y(i,j)}{\sum_{i=0}^{n-1} \sum_{j>i}^{n-1} C_X(i,j)}$$

where $\mathbf{Y} = T\mathbf{X}$. The decorrelation efficiency of the Haar transform for models as in Figure 5.5 is in excess of 99% (see Table 5.2). The Haar transform is thus a good approximation to the Karhunen-Loève Transform (KLT) for these models.

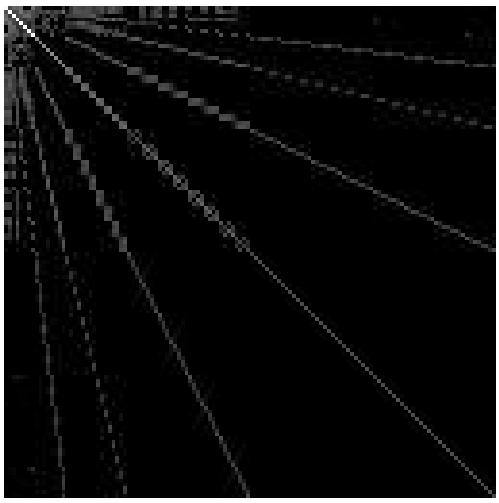
The rapid decay in variance with increasing coefficient index is displayed in Figure 5.6, in which it may be observed that the variances adhere to the tree structure of the wavelet decomposition, all coefficients at the same resolution having similar variances.

Table 5.2: Decorrelation efficiency of Haar transform for exactly self-affine models as in Figure 5.5.

	Mean removal	
Δ_d	without	with
2	99.34%	99.62%
16	99.31%	99.68%

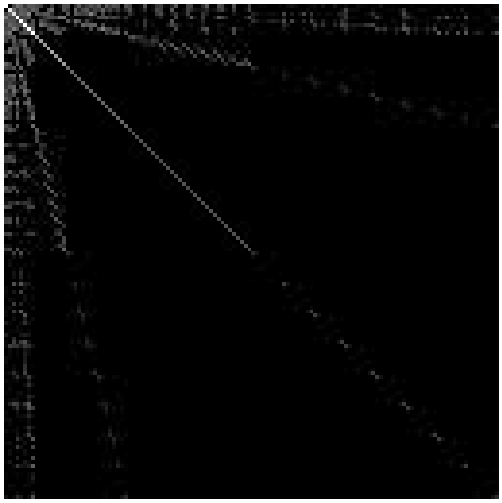
[12] Gamma correction by a factor γ consists of normalising all pixel values to the range $[0,1]$, applying the function $f(x) = x^{1/\gamma}$ and finally normalising back to the original range of pixel values.

[13] It is interesting to note that the positioning of these residual correlations is similar to those after a Daubechies basis wavelet transform of an AR(1) model [55].

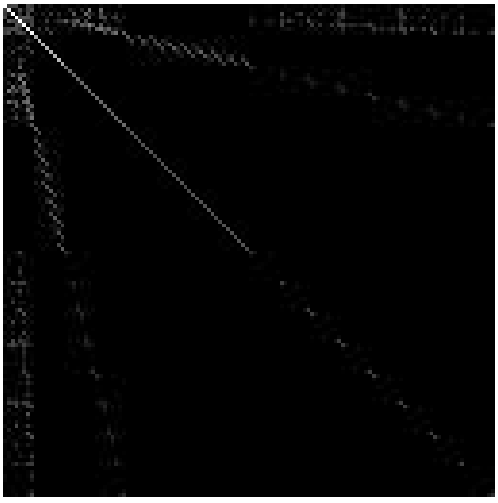


$\Delta_d = 2$, without mean removal

$\Delta_d = 2$, with mean removal



$\Delta_d = 16$, without mean removal



$\Delta_d = 16$, with mean removal

Figure 5.5: Haar transform domain autocorrelation of exactly self-affine model ($n = 128, b = 16$). A gamma correction factor of 4.0 has been applied to the images to emphasise small off-diagonal correlations, which would otherwise not be visible.

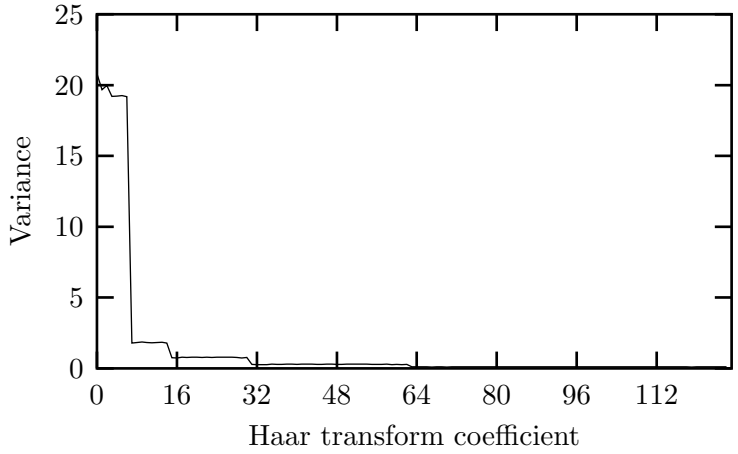


Figure 5.6: Haar transform coefficient variances for model parameters as in Table 5.1.

5.3.2 Marginal distributions

The marginal distributions of the Haar transform coefficients are reasonably well approximated by a spectrum of generalised Gaussian distributions, with shape parameter $\nu \approx 2$ (similar to a Gaussian) for the initial N_r low resolution coefficients[14], and $\nu \leq 1$ for the remaining $n - N_r$ high-resolution coefficients (see Figure 5.7), with ν becoming progressively smaller with increasing resolution. The Gaussian-like distributions may be attributed to the summing of independent random variables (the offset coefficients) generating the low-resolution coefficients, while the Laplacian-like distributions reflect the influence of multiple products of random variables [187] (the scaling coefficients).

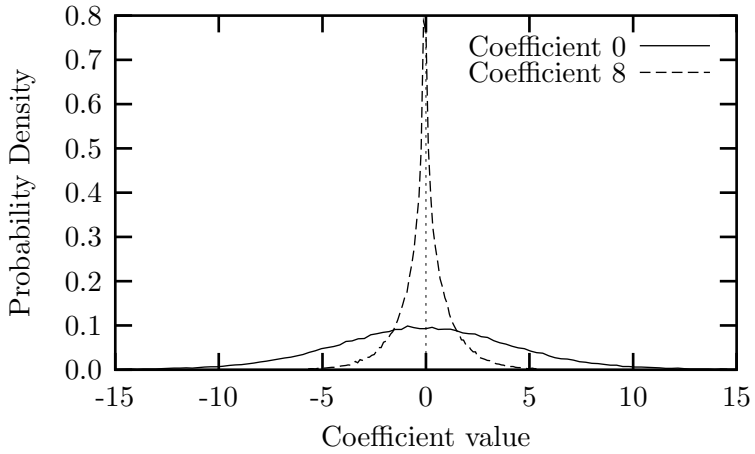


Figure 5.7: Haar transform coefficient marginal distributions in Figure 5.5.

[14]Where N_r is a power of 2.

5.4 Coding Experiments

Despite the minimal correlations between Haar transform coefficients for these models, one would expect a significant degree of dependence between these coefficients from inspection of the generating mechanism. Any dependence remaining after decorrelation is significantly non-linear[15], and is inadequately characterised by the autocorrelation. The significance of the residual dependence was measured by comparing the relative effectiveness of fractal coding (quantising and storing the self-affine representation coefficients) and transform coding (scalar quantising and storing the Haar transform coefficients) of signals generated by an exactly self-affine model. This comparison was achieved by generating an ensemble of self-affine signals (model parameters as in Table 5.1) which were encoded by both methods at an equivalent bit rate. All distortions are measured in terms of the MSE distortion measure.

5.4.1 Transform coding quantisation

Haar transform coefficients were quantised using optimal Gaussian and Laplacian quantisers [120] [131] [150], corresponding to the distributions observed (see Section 5.3.2) for low and high resolution coefficients respectively.

Bit allocation for transform coding was based on Equation (2.3). Since the optimum bit allocation R_k for coefficient k resulting from this allocation is not necessarily a positive integers, a heuristic procedure was adopted. Starting with coefficient 0, $\lceil R_k \rceil$ bits were allocated for coefficient k, allocating zero bits to the remaining coefficients once all of the available bits had been allocated.

5.4.2 Fractal coding quantisation

Since the analytic determination of the optimum bit allocation for quantisation of the fractal coefficients is not as straightforward as in the case of transform coding, this allocation was determined by Monte Carlo simulation. Figure 5.8 illustrates the distortions obtained for various bit allocations to each scaling and offset coefficient, while the optimum bit allocations[16] determined from the distortion surface are displayed in Figure 5.9. The distortions[17] obtained at these bit allocations are displayed in Figure 5.10. Optimal uniform and Gaussian quantisers were used for the scaling and offset coefficients respectively.

[15]The dependence present in a multivariate Gaussian is linear in the sense that the absence of correlation implies independence.

[16]These correspond closely to those resulting from the heuristic allocation procedure in Section 4.5.5. The general tendency to allocate more bits to the offset than the scaling is compatible with the results for test images discussed in Section 3.5.

[17]It is interesting to note that the distortion is approximately proportional to $2^{-14.4R}$, which represents a significantly faster decay with increasing rate than that for optimal coding of a multivariate Gaussian, which is proportional to 2^{-2R} for small distortions [99, pg. 644].

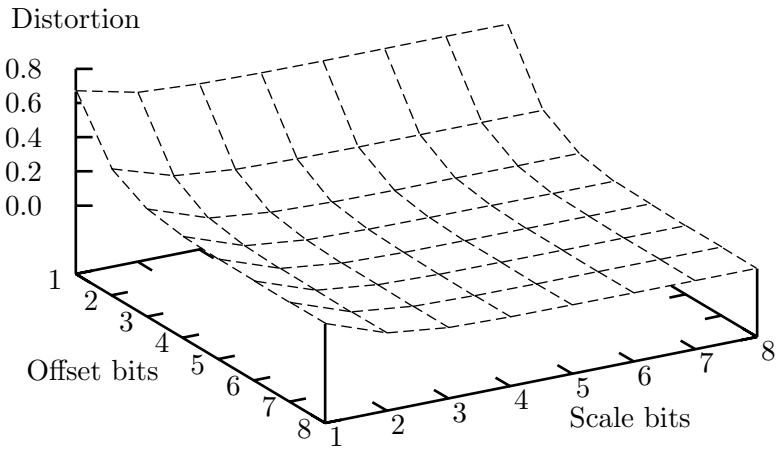


Figure 5.8: Distortion surface for varying scaling and offset bit allocations in coding of exactly self-affine signals (model parameters as in Table 5.1).

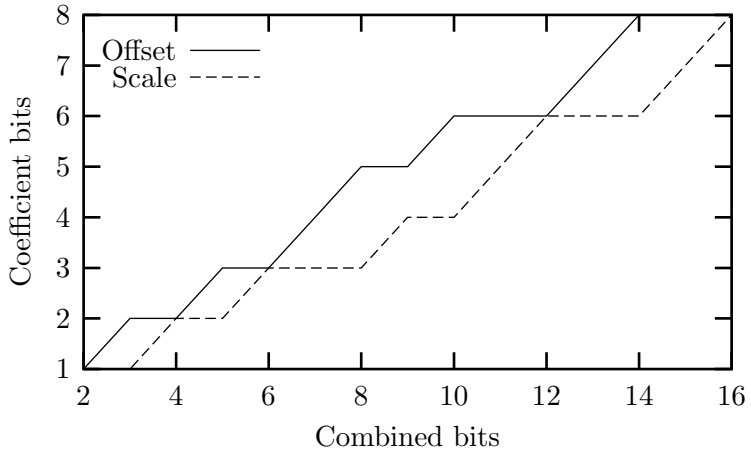


Figure 5.9: Optimum scaling and offset coefficient bit allocations selected from Figure 5.8.

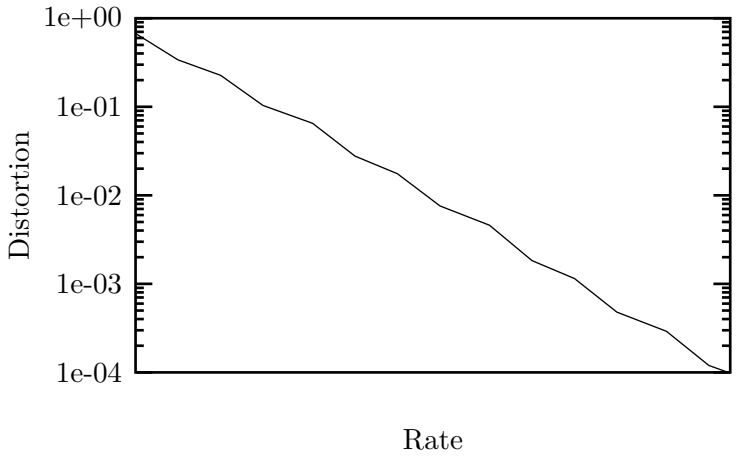


Figure 5.10: Distortion for optimum bit allocations to scaling and offset coefficients in coding of exactly self-affine signals (model parameters as in Table 5.1). The bit rate is obtained by dividing the total bit allocation per range block (the sum of the bits required to distinguish between each of the 49 possible domain blocks and the bits allocated to the scaling and offset coefficients) by the range block size.

5.4.3 Distortion comparisons

The results of the coding experiments are displayed in Figure 5.11. The considerable differences in distortions, especially at rates above 1 bit/sample are evidence for significant dependence between Haar transform coefficients. It is clear that, although the coefficients are efficiently decorrelated by the linear transform, they remain mutually dependent, which has a significant effect on the efficiency of their independent scalar quantisation. As a result, fractal coding would be expected to be considerably more effective than transform coding for a signal source sufficiently similar to an exactly self-affine model.

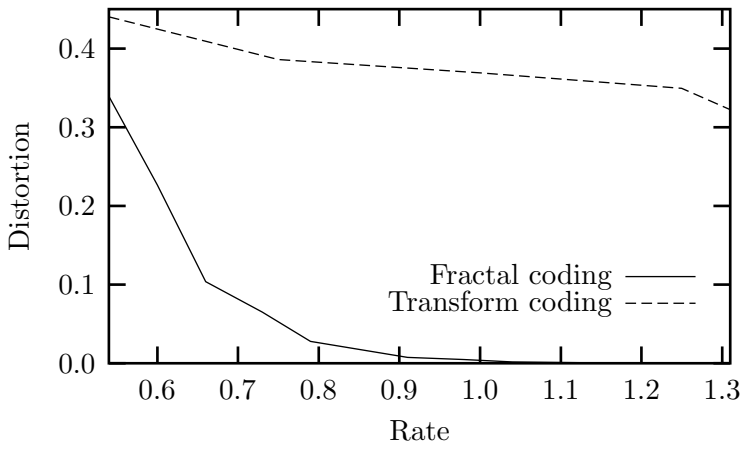


Figure 5.11: Distortion rate comparison for fractal and transform coding of exactly self-affine signals (model parameters as in Table 5.1).

5.5 Explicit Haar Domain Model

The exactly self-affine models may be explicitly constructed[18] in the Haar transform domain by considering the wavelet domain decoding operation described in Section 3.9 to operate on stochastic s_i, o_i and p_i. The subband extrapolation mechanism suggests a slight variation[19] on the model definition, in which the o_i become the actual coefficient values of domain subtree roots at resolution i_d. There are thus 2^{i_d} domain subtrees at resolution i_d in the signal wavelet decomposition and 2^{i_r} range subtrees at resolution $i_r = i_d+1$. Values of coefficients at levels[20] below i_d are determined by the scaling coefficient "paths" of which they form the termination (see Figure 5.12). The actual paths are, of course, determined by the particular assignment of domain subtrees to range subtrees. Given a path of length n, originating at domain block l_0 and subsequently passing through ranges $l_1, l_2, \ldots, l_n$, the value of the coefficient at the end of the path is $\left(\frac{1}{\sqrt{2}}\right)^n o_{l_0} s_{l_1} s_{l_2} \ldots s_{l_n}$. These inherently wavelet domain models are product models in the sense that the individual innovations process values generating the models are multiplied[21] together, rather than added as in an AR model.

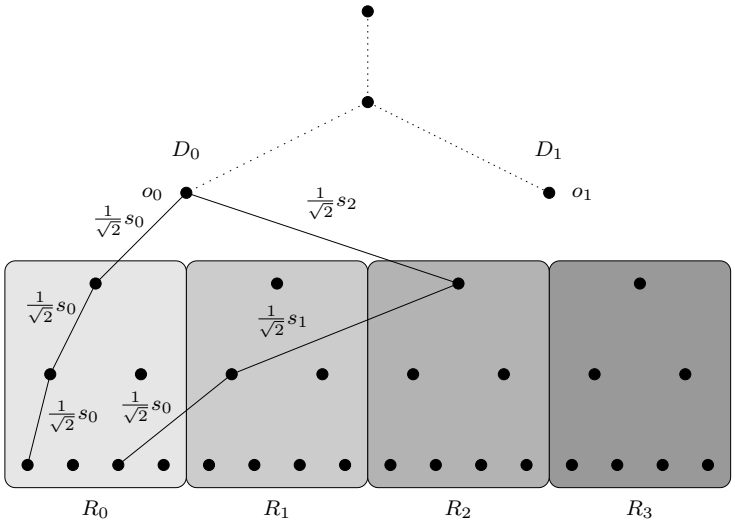


Figure 5.12: Paths of scaling factors for detail coefficients with values $\left(\frac{1}{\sqrt{2}}\right)^3 o_0 s_0^3$ and $\left(\frac{1}{\sqrt{2}}\right)^3 o_0 s_2 s_1 s_0$, for domain to range mappings as in Figure 3.4.

[18] The notation of Section 3.9 is adopted for the wavelet transform domain models, with domain and range indices starting at zero and the index of the domain mapped to range i denoted by p_i. The domain increment is assumed to be the same as the domain block width.

[19] This form of model leaves coefficients at resolutions below i_d unspecified.

[20] Higher resolutions correspond to lower "levels" in the decomposition tree.

[21] The models may be made additive by application of the log function, but complications arise when values are not strictly positive.

5.5.1 Variance decay with increasing resolution

The variation in detail coefficient variance with increasing resolution is an important statistical property in the wavelet transform domain, corresponding to the power spectral density in many respects (the power spectral density measures signal energy at each frequency, whereas the detail coefficient variance measures signal energy at each resolution). Wavelet domain statistical analysis has been found to be a powerful tool in the analysis of non-stationary signal models such as fractional Brownian motion [66] [67] [128] [191] [205]. Analytic evaluation of the detail coefficient variance for the wavelet domain self-affine models proved rather complicated, and although analytic bounds on the coefficient variance were obtained as described in this section, accurate estimates required Monte Carlo simulations.

Calculation of the average variance of a self-affine model at each resolution requires averaging over the variances of each coefficient at that resolution. Given a path of length n, originating at domain block l_0 and subsequently passing through ranges $l_1, l_2, \ldots, l_n$, the value of the square of the terminating coefficient is $\left(\frac{1}{2}\right)^n o_{l_0}^2 s_{l_1}^2 s_{l_2}^2 \ldots s_{l_n}^2$. Fixing the domain to range mapping and the specific path $(l_0, l_1, \ldots, l_n)$, the variance of that coefficient is $\frac{\sigma_o^2}{2^n} E[s_{l_1}^2 s_{l_2}^2 \ldots s_{l_n}^2]$, assuming the o_i and s_i are independent and zero-mean. Since this expectation is independent of the value of l_0, it may be excluded from the path for the purposes for which it is required here, the full path being considered to be $(l_1, l_2, \ldots, l_n)$. The average variance at a particular resolution is therefore the average of each of these expectations over all paths of the appropriate length, and for all possible domain to range assignments. Evaluation of this expectation is complicated by the possible presence of multiple occurrences of the same s_i in a path, with the result that the expectation is equal to the product of even central moments of the s_i, depending on how many multiple occurrences are in the path[22]. A log decay with increasing resolution would be obtained in the absence of this complicating factor. Such a decay is associated with the stochastic fractal fractional Brownian motion models [67] [124], which have been proposed as image texture models [154].

Since the variance of a coefficient terminating a particular path is dependent on the number of multiple occurrences of the same s_i in that path, a classification of the paths is required so that the average variance at a particular resolution may be calculated by appropriately weighting the variances of each class of path in averaging over all domain to range assignments. The natural classification in this situation is in terms of the occurrences of unique scaling factors. Consider a path of length n, where each entry is one of k scaling factors (corresponding to the $k = 2^{i_r}$ range subtrees), and h_0 is the number of the scaling

[22] Alternative, and somewhat simpler wavelet domain product models may be derived from the coding scheme described in Section 3.9.4, in which independent scaling coefficients are selected between each pair of levels and consequently the second order statistics of the models are not dependent on higher order statistics of the innovations process.

factors not appearing in the path, h_1 is the number of scaling factors appearing once only, and so on until h_n, which is the number of scaling factors appearing n times. Imposing the condition

$$\sum_{i=0}^{n} h_i = k$$

guarantees that each of the k scaling factors is accounted for, and

$$\sum_{i=0}^{n} i h_i = n$$

guarantees that the path contains n entries. Each path belonging to a particular class has the same associated variance of the terminating coefficient, which is

$$\frac{\sigma_o^2}{2^n} \prod_{i=1}^{n} m_{2i}^{h_i},$$

where m_i are the central moments of the distribution of the scaling coefficient.

The uniform and Laplacian distributions are considered here as possible distributions for the scaling coefficient. The Laplacian distribution is utilised for reasons revealed in Chapter 6, while the uniform distribution is useful for purposes of comparison, having been utilised for this purpose in the initial sections of this chapter. A zero-mean uniform distribution (see Appendix A) on the range $[-a, a]$ has pdf

$$f_X(x) = \begin{cases} \frac{1}{2a} & \text{if } x \in [-a, a] \\ 0 & \text{otherwise.} \end{cases}$$

Central moments m_i for even i are

$$m_i = \frac{a^i}{i+1},$$

and since $a = \sqrt{3}\,\sigma$,

$$\prod_{i=1}^{n} m_{2i}^{h_i} = \sigma_s^{2n} \prod_{i=1}^{n} \left(\frac{3^i}{2i+1} \right)^{h_i}.$$

The zero-mean Laplacian distribution (see Appendix A) is

$$f_X(x) = \frac{1}{\sigma\sqrt{2}} e^{-\frac{\sqrt{2}}{\sigma}|x|}.$$

Central moment i may be written

$$\begin{aligned} m_i &= \frac{1}{\sigma\sqrt{2}} \int_{-\infty}^{\infty} e^{-\frac{\sqrt{2}}{\sigma}|x|} x^i \, dx \\ &= \frac{2}{\sigma\sqrt{2}} \int_{0}^{\infty} e^{-\frac{\sqrt{2}}{\sigma}x} x^i \, dx \text{ for } i \text{ even.} \end{aligned}$$

Utilising the standard integral[23] [58, pg. 230]

$$\int_0^\infty x^n e^{-ax} \, dx = \frac{n!}{a^{n+1}} \text{ if } a > 0,\ n \in \mathbb{Z}^+,$$

the central moments for even i are

$$m_i = \frac{\sigma^i i!}{2^{i/2}},$$

and therefore

$$\prod_{i=1}^{n} m_{2i}^{h_i} = \sigma_s^{2n} \prod_{i=1}^{n} \left(\frac{(2i)!}{2^i} \right)^{h_i}.$$

In both cases then, the product may be expressed in terms of the variance of s and a function M of the class $(h_1, h_2, \ldots, h_n)$

$$\prod_{i=1}^{n} m_{2i}^{h_i} = \sigma_s^{2n} M(h_1, h_2, \ldots, h_n).$$

The average variance at a particular resolution is obtained by averaging over all possible paths in all domain to range mappings. If $N(l_1, \ldots, l_n)$ is the number of mappings in which a particular path occurs, $N_P = 2^{i_d+n} = 2^n \frac{k}{2}$ is the number of paths for each distinct mapping, and $N_M = N_d^{N_r} = \left(\frac{k}{2}\right)^k$ is the number of distinct mappings, the desired average is obtained by summing

$$\frac{N(l_1, \ldots, l_n)}{N_P N_M} \frac{\sigma_o^2}{2^n} E[s_{l_1}^2 s_{l_2}^2 \ldots s_{l_n}^2]$$

over all possible paths $(l_1, \ldots, l_n)$. Each of these terms may be expressed as

$$\frac{N(l_1, \ldots, l_n)}{N_P N_M} \frac{\sigma_o^2}{2^n} \prod_{i=1}^{n} m_{2i}^{h_i} = \frac{N(l_1, \ldots, l_n)}{N_P N_M} M(h_1, h_2, \ldots, h_n) \sigma_o^2 \left(\frac{\sigma_s^2}{2} \right)^n,$$

where $(h_1, h_2, \ldots, h_n)$ is the class of path $(l_1, \ldots, l_n)$.

The value of $N(l_1, \ldots, l_n)$ may be determined by considering the restrictions placed on the possible mappings by every transition $(l_i)(l_{i+1})$ in the path. The occurrence of transition $(l_i)(l_{i+1})$ in a path implies that $p_{l_{i+1}} = \lfloor l_i/2 \rfloor$, since otherwise the transition would not be possible. It is clear from this observation that $p_{l_{i+1}} = \lfloor l_i/2 \rfloor$ implies that transitions $(2\lfloor l_i/2 \rfloor)(l_{i+1})$ and $(2\lfloor l_i/2 \rfloor + 1)(l_{i+1})$ are possible, and they are therefore equivalent with respect to this criterion. Each unique transition, where uniqueness requires that an equivalent

[23] The positive integers are defined as $\mathbb{Z}^+ = \{n \in \mathbb{Z} \mid n > 0\}$.

transition may not occur elsewhere in the path, specifies one of the p_i. Denoting the number of remaining unspecified p_i by N_U,

$$N(l_1, \ldots, l_n) = N_D^{N_U} = \left(\frac{k}{2}\right)^{N_U}.$$

Since calculation for all possible paths $(l_1, \ldots, l_n)$ is impractical for all but the smallest path lengths, a more efficient method of summing

$$\frac{N(l_1, \ldots, l_n)}{N_P N_M} M(h_1, h_2, \ldots, h_n) \sigma_o^2 \left(\frac{\sigma_s^2}{2}\right)^n$$

over all paths is to enumerate the number of paths in each class and sum the above with the appropriate weighting for each class. Given the set of all paths of length n, where each entry is one of k scaling factors, the number of paths in each class $(h_0, h_1, \ldots, h_n)$ may be enumerated by application of the multinomial coefficient [110] as follows. Consider apportioning the h_i over the set of scaling factors. One way of doing so would be to have the first h_0 of the k scaling factors not occur, the next h_1 occur once and so on. There are

$$\frac{k!}{h_0! h_1! \ldots h_n!}$$

ways of apportioning the h_i over the set of scaling factors, and

$$\frac{n!}{(0!)^{h_0} (1!)^{h_1} \ldots (n!)^{h_n}}$$

different paths for each of the scaling factor utilisations. The total number of paths in class $(h_0, h_1, \ldots, h_n)$ is therefore

$$N_C(h_0, h_1, \ldots, h_n) = \frac{k! n!}{\prod_{i=0}^{n} h_i! (i!)^{h_i}}.$$

The value of $N(l_1, \ldots, l_n)$ is not constant for all paths in the same class, complicating calculation of the average variance for a particular resolution. Although no solution was found to this problem, it is clear by inspection that the value of $N(l_1, \ldots, l_n)$ is constant for all $(l_1, \ldots, l_n)$ in the same class, for classes $(k-1, 0, 0, \ldots, 0, 1)$ and $(k-n, n, 0, 0, \ldots, 0)$. In these cases the coefficient for the expectation of the *class* (in the average over all classes required to obtain the average variance at a particular resolution) is the number of paths in the class multiplied by the coefficient for each path. Evaluation of the expectations for these classes at least provides an indication of the behaviour of the average variance over all classes.

Since only one transition occurs for class $(k-1, 0, 0, \ldots, 0, 1)$ when $n > 1$, only one of the p_i is specified, and

$$N(l_1, \ldots, l_n) = N_D^{N_R - 1} = \left(\frac{k}{2}\right)^{k-1}.$$

Since

$$\frac{N(l_1,\ldots,l_n)}{N_P N_M} = \frac{2^{2-n}}{k^2}$$

and there are

$$N_C(k-1,0,0,\ldots,0,1) = \frac{k!n!}{(k-1)!n!} = k$$

paths in the class, the coefficient for $\frac{\sigma_o^2}{2^n}\prod_{i=1}^{n} m_{2i}^{h_i} = \frac{\sigma_o^2}{2^n} m_{2n}$ in this case is

$$\frac{N_C(k-1,0,0,\ldots,0,1)N(l_1,\ldots,l_n)}{N_P N_M} = \frac{2^{2-n}}{k}.$$

The coefficient for $\sigma_o^2\left(\frac{\sigma_s^2}{2}\right)^n$ for the uniform pdf is then

$$\frac{2^{2-n}3^n}{k(2n+1)}$$

whereas for the Laplacian pdf it is

$$\frac{(2n)!2^{2-2n}}{k}.$$

Since this class represents the paths consisting of multiple occurrences of a single s_i, and consequently represents the influence of the highest order moments possible, the greater these values the greater the deviation from simple log decay of the variance. The example values displayed in Table 5.3 indicate that the deviation from log decay is far greater for the Laplacian distribution of scaling coefficients than for the uniform distribution, and increases with increasing path length.

Table 5.3: Values of coefficients of $\sigma_o^2\left(\frac{\sigma_s^2}{2}\right)^n$ for class $(k-1,0,0,\ldots,0,1)$ and $k=16$.

n	1	2	3	4	5
Uniform	0.0000	0.1125	0.1205	0.1406	0.1726
Laplacian	0.0000	0.3750	2.8125	39.3750	885.9375

Since each s_i occurs only once for class $(k-n,n,0,0,\ldots,0)$, every transition is unique, and

$$N(l_1,\ldots,l_n) = N_D^{N_R-n+1} = \left(\frac{k}{2}\right)^{k-n+1}.$$

Since

$$\frac{N(l_1,\ldots,l_n)}{N_P N_M} = k^{-n}$$

and there are

$$\frac{k!n!}{(k-n)!n!} = \frac{k!}{(k-n)!}$$

paths in the class, the coefficient for $\frac{\sigma_o^2}{2^n}\prod_{i=1}^{n} m_{2i}^{h_i} = \frac{\sigma_o^2}{2^n} m_2^n$ in this case is

$$\frac{N_C(k-n,n,0,0,\ldots,0)N(l_1,\ldots,l_n)}{N_P N_M}\frac{k!}{(k-n)!k^n}.$$

The coefficient for $\sigma_o^2\left(\frac{\sigma_s^2}{2}\right)^n$ for both pdfs is

$$\frac{k!}{(k-n)!k^n}$$

since $M(h_1,h_2,\ldots,h_n) = 1$ in both cases. Since this class represents paths containing no multiple occurrences of scaling coefficients, which are therefore dependent only on the variance σ_s^2, the closer these values are to unity, the less the deviation from simple log decay of the variance. The example values displayed in Table 5.4 indicate increasing deviation from log decay with increasing path length.

Table 5.4: Values of coefficients of $\sigma_o^2\left(\frac{\sigma_s^2}{2}\right)^n$ for class $(k-n,n,0,0,\ldots,0)$ and $k=16$.

n	1	2	3	4	5
Coefficient	1.0000	0.9375	0.8203	0.6665	0.4999

Although these calculations allow an estimate of the behaviour of the variance decay with increasing resolution, exact computation is difficult, and Monte Carlo methods are consequently used for the results displayed in Figure 5.13, in which the considerably greater deviation from linearity on a log scale for the Laplacian[24] pdf is visible. The deviation from log decay is reduced with increasing k and decreasing path length, both of which result from an increase in i_d, which corresponds to a decreased range block size. Unlike an AR model for example, even the second order statistics of an exactly self-affine model are sensitive to the pdfs of the innovations process.

An interesting comparison is possible between a log variance decay and a theorem connecting the Lipschitz regularity[25] of a function with the decay of its wavelet transform maxima with increasing resolution. It may be shown [126] that if (and only if) a function $f(x)$ is

[24]Random values with a Laplacian distribution with mean μ and variance σ^2 were generated by passing values uniformly distributed on $(-1,0)\cup(0,1)$ through the function

$$f(x) = \begin{cases} \mu + \frac{\sigma}{\sqrt{2}}\ln|x| & \text{if } x \le 0 \\ \mu - \frac{\sigma}{\sqrt{2}}\ln|x| & \text{otherwise.} \end{cases}$$

[25]A function $f(x)$ is *uniformly Lipschitz* α (where $0 \le \alpha \le 1$) over an interval (a,b) iff $\exists K \in \mathbb{R}$ such that $\forall x, y \in (a,b)$

$$|f(x) - f(y)| \le K|x-y|^{\alpha}.$$

The Lipschitz uniform regularity [126] of $f(x)$ is the upper bound of all α for which $f(x)$ is uniformly Lipschitz α. It should be noted that the Lipschitz condition as defined here is often referred to as Hölder continuity [60, ch. 1] [100, ch. 2], in which case Lipschitz continuity is the special case of $\alpha = 1$.

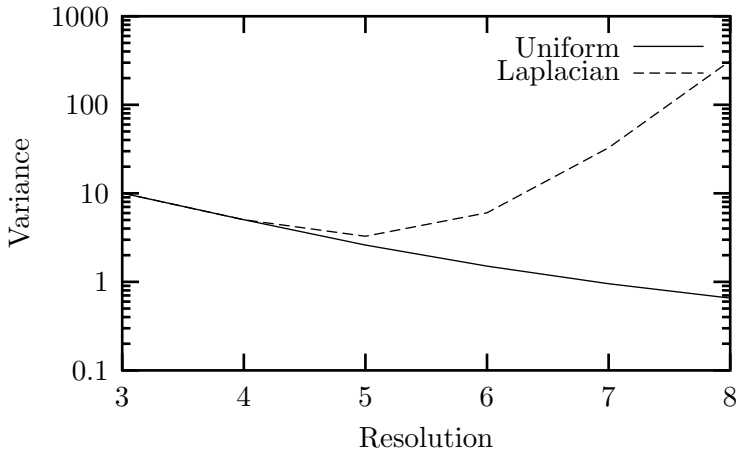


Figure 5.13: Comparison of variance decay for exactly self-affine models with uniform and Laplacian scaling coefficients ($i_d = 3$).

uniformly Lipschitz α over (a, b) then $\exists K \in \mathbb{R}$ such that $\forall x \in (a, b)$

$$|Q_i f(x)| \leq K(2^{-\alpha})^i,$$

where Q_i is the projection operator into wavelet detail space W_i (see Appendix C). Since the signal magnitude on an interval of a subband is bounded by the maximum of $|Q_i f(x)|$, this provides a relationship between the magnitude decay across subbands and the regularity of the signal (a detailed study of which is not pursued here).

5.5.2 Autocorrelations

A generalisation of the analysis above is required in calculating correlations between detail coefficients for a self-affine model. In this case *two* paths must be considered, one for each of the coefficients between which the correlation is desired. Since the origin of the paths is significant in this context, the l_0 must be included in each path.

Given a fixed domain to range assignment, the correlation between two coefficients is zero if any of the s_i occurs only once in either of the paths, or if the paths originate at different o_i (since the s_i and o_i are assumed independent and zero-mean). Non-zero correlations for fixed domain to range assignments depend on the number of multiple occurrences of scaling coefficients in the paths, resulting in dependence of the autocorrelation on the pdfs of the innovations process via their higher order moments.

The autocorrelation results presented here were generated by Monte Carlo methods due to the considerable complexity involved in an analytic derivation. A comparison of the autocorrelations for uniform and Laplacian scaling distributions is presented in Figures 5.14 and 5.15, and the corresponding average correlation coefficients between detail coefficient pairs at the same resolution and between child parent pairs are displayed in Table 5.5. Although the

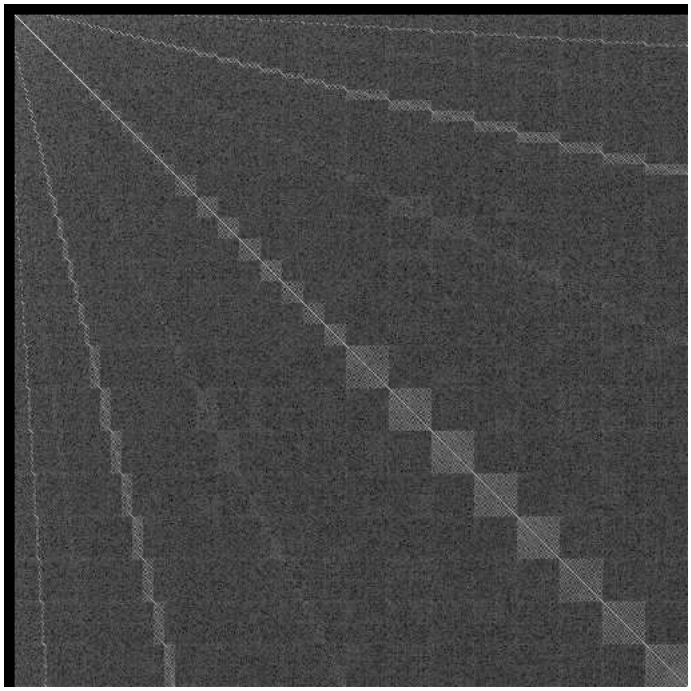


Figure 5.14: Wavelet domain autocorrelation of exactly self-affine model (uniform scaling coefficients, $i_d = 3$, $\mu_o = \mu_s = 0$, $\sigma_o^2 = 10$ and $\sigma_s^2 = 1$). Visibility has been enhanced by applying a gamma correction of 7 to the absolute values of the pixels.

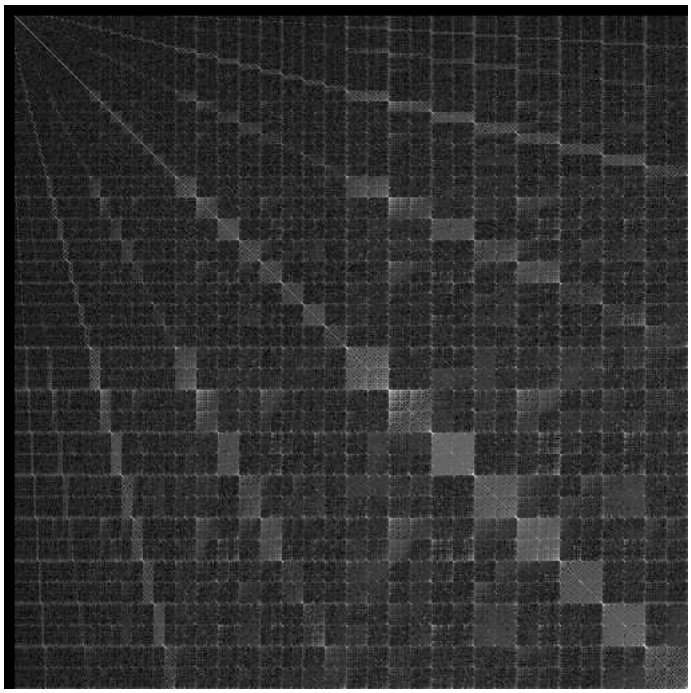


Figure 5.15: Wavelet domain autocorrelation of exactly self-affine model (Laplacian scaling coefficients, $i_d = 3$, $\mu_o = \mu_s = 0$, $\sigma_o^2 = 10$ and $\sigma_s^2 = 1$). Visibility has been enhanced by applying a gamma correction of 14 to the absolute values of the pixels.

Table 5.5: Average detail coefficient variances and correlation coefficients for self-affine models as in Figures 5.14 and 5.15. The child-parent correlation coefficients are attached to the parent resolution in the table.

	Uniform scaling			Laplacian scaling		
		Correlation			Correlation	
Resolution	Variance	Child	Same	Variance	Child	Same
3	10.00	0.00		10.00	0.00	
4	5.00	0.00	0.00	4.99	0.00	0.00
5	2.62	0.00	0.00	3.30	-0.01	0.00
6	1.50	0.00	0.01	5.78	-0.04	0.00
7	0.95	0.00	0.01	27.76	-0.10	0.00
8	0.66		0.02	227.56		0.00

visible patterns are similar, there are significant differences in the magnitudes. The low off-diagonal correlations[26] may be understood by noting that a significant fraction of the possible domain to range assignments are likely to result in the conditions for the zero correlations described above.

5.6 Conclusions

A class of exactly self-affine signal models is defined in this chapter, enabling an investigation of the statistical consequences of exact self-affinity by evaluating the properties of these signal models. A generalised Lattice VQ interpretation of fractal coding is proposed, and the structure of the resulting codebooks is associated with the properties of these models.

It is important to emphasise that the statistics of an exactly self-affine model are dependent on model parameters such as block partition, block transforms and statistical distributions for the transform coefficients, and significantly different results may be obtained for different model parameters. Nevertheless, the results for the limited range of parameters investigated provide some insight into the implicit statistical assumptions of fractal coding. While these assumptions need not be perfectly accurate in order for fractal coding to be effective, nor is the reduction in coding efficiency resulting from any observed mismatch in these statistics easily quantified, the more closely these assumptions are adhered to by a particular source, the more effective is fractal coding of that source.

The second order statistics of the exactly self-affine models, resulting from the affine self-similarity assumption, are in broad terms similar to those commonly assumed for transform coding; approximately uncorrelated blocks (under a range of assumptions on the distributions of the scaling and offset coefficients) and decreasing autocorrelation within each block. The finer details of the autocorrelation function are however, not easily tuned to desired statistics by varying the model parameters. Similar analysis in the wavelet domain indicates that the

[26] Considerable enhancement has been applied to the images to make these patterns visible.

rate of decay of variance with increasing resolution may be controlled to some extent by suitable choices of model parameters. This decay is compared with that measured for a set of test images in the following chapter. The wavelet domain correlations of the exactly self-affine models are small, but the actual pattern of these correlations is rather unusual.

The poor performance of transform coding of decorrelated signal samples is evidence for significant non-linear dependence between samples in the exactly self-affine models. Since these models are poorly described by their second order statistics, a characterisation of the non-linear dependence is desirable. Higher order statistics [132], which may be used to characterise dependence other than correlation, were not calculated since "standard" properties of these statistics are not well known for images, and their calculation by Monte Carlo simulation was considered computationally too demanding (and unstable, since pseudo-random number generators do not produce independent numbers). Since no suitable means of characterising the dependence was found, an evaluation of the compatibility of self-affinity with the properties of natural images was performed by direct measurement of the degree of self-affinity of a set of test images as, described in the following chapter.

The distributions of matching domains for each range across the signal were assumed uniform for all self-affine models described in this chapter. Although this assumption is justified to some extent by a corresponding assumption made in representing these values in many fractal coding schemes, the resulting long-distance dependence[27] is incompatible with entropy measurements for images [37, ch. 2] which suggest that the dependence between pixels falls off rapidly with increasing distance between them. The simplest means of removing this long-distance dependence from the model is to fix the matching domain for each range to be the spatially nearest domain, as in domain search free fractal coding schemes [135]. In this case however, the local dependence is unreasonably strong, since it results in a severe restriction on the possible range block "shapes". It appears as if exact self-affinity is necessarily associated with a form of dependence between samples that is incompatible with known image characteristics, although this has not been proven here since no adequate means of characterising this dependence was found.

Since *exact* self-affinity is not necessary for effective fractal coding, an understanding of the requirements for approximate but strongly present self-affinity would be productive. In particular, it is not clear how strongly self-affine a signal model may be without requiring the unnatural forms of dependence observed for exactly self-affine models. A possible approach, which has not been pursued, is to create an approximately self-affine signal model by adding varying degrees of "noise" derived from a multiscale AR model [22] [55] to an exactly self-affine model.

[27] This dependence is necessary for the domains and ranges to be able to "cooperate" so that there is a matching domain for each range.

Chapter 6

Image Properties in the Wavelet Transform Domain

The estimation of the self-affinity of "natural" images by an examination of the wavelet domain properties of a set of standard images (see Appendix E) is described in this chapter. Analysis in the wavelet transform domain is convenient, not only in terms of the simplified mechanism of fractal coding, but also in allowing an evaluation of the merits of the generalisation of fractal coding to a smooth wavelet basis.

The initial examination of distributions of detail coefficients and the ratios between them provides a basis for the selection of appropriate distributions for scaling and offset coefficients in a two-dimensional extension of the wavelet domain self-affine models of the previous chapter. This construction allows the detail coefficient variance decay associated with exact self-affinity to be compared with the decay measured in the set of standard images. The correlations between detail coefficients in the test images are measured, and provide valuable insights into the structure of the domain pool codebooks of these images. Finally, the degree of self-affinity of these images is quantified by comparisons with suitably constructed random and optimised codebooks.

The two-dimensional wavelet transform domain fractal coding scheme required both in the construction of an exactly self-affine model and in the self-affinity calculations utilises the non-standard construction of a two-dimensional basis (see Appendix C). The operations on subtrees are identical in two dimensions, except that each node in the coefficient tree has four children instead of two, and each subtree contains the three equivalent subtrees in the three directional subbands of the decomposition. The block isometry operations are not utilised since they are not an essential part of fractal coding[1] and contribute unnecessary additional complication.

The wavelet bases used in these evaluations were the Haar basis (in which wavelet domain

[1]These operations appear in fact to be counter-productive in a rate distortion sense (see Section 3.2.1).

fractal coding is equivalent to standard fractal coding) and the "Spline Variant with Less Dissimilar Lengths" basis of Antonini *et al.* [7], which was selected since it provided the best performance in the evaluation by Antonini *et al.*, and has been applied to fractal coding with considerable success [46].

6.1 Detail Coefficient Distributions

The properties of individual transform coefficients, and their dependence on their neighbours, is investigated in this section. Since practical constraints prevented the assembly of a large ensemble of images from which ensemble statistics could be calculated, stationarity was assumed and statistics were estimated from averages within individual images. Marginal distributions of detail coefficients, for example, were assumed to be the same for all coefficients at a particular resolution, and the distributions were estimated by assembling an ensemble of coefficients from all of the coefficients at that resolution in a single image, rather than from all coefficients at a fixed position across an ensemble of images.

6.1.1 Marginal distributions

The marginal distributions of detail coefficients (in both bases) in each subband were found experimentally to be well approximated by a generalised Gaussian distribution (see Appendix A). The shape parameter ν varied between 0.5 and 1.5 over all of the test images, with the majority close to 1.0, for which the generalised Gaussian is equivalent to a Laplacian distribution. These results are roughly in agreement with those of Antonini *et al.* [7], who found a best fit to a generalised Gaussian distribution (see Appendix A) with parameter $\nu = 0.7$. No tendency towards decreasing ν with increasing resolution was observed, which is not compatible with the properties of the exactly self-affine models described in the previous chapter.

Histograms of the detail coefficient values at resolution 8 of the horizontal directional subband of the "Lena" image are displayed in Figures 6.1 and 6.2, the peaks visible in Figure 6.1 being averaged out in Figure 6.2 by the use of a sufficiently large bin size. These peaks stem from the interaction between the rectangular shape of the Haar basis functions[2] and the discrete range of pixel values of the 8 bit image, and are barely discernible for smooth wavelet bases. This "pre-quantisation" effect may significantly influence the efficiency of a high bit rate scalar quantiser optimised for a pdf without pre-quantisation [130]. If σ^2 is the variance of the random variable to be scalar quantised, and $\Delta = \frac{1}{2}$ is the pre-quantisation interval, then a rate R quantiser optimised ignoring the pre-quantisation is efficient for $\frac{\sigma}{\Delta} > \frac{2^R}{4}$ [130], which is equivalent to requiring $R < \log_2 \sigma + 3$ for Haar basis detail coefficients. Depending on their construction, the offset coefficients of a fractal coding scheme may be subject to

[2]The peaks occur at intervals of $\frac{1}{2}$, which is the normalisation factor from one resolution to the next.

this effect, which should be taken into account in the design of a suitable quantiser. Since pre-quantisation is also visible in the ratios between child and parent coefficients in the Haar basis, the scaling coefficients may be subject to a similar effect. These effects need not be considered for a smooth wavelet basis based fractal coding scheme.

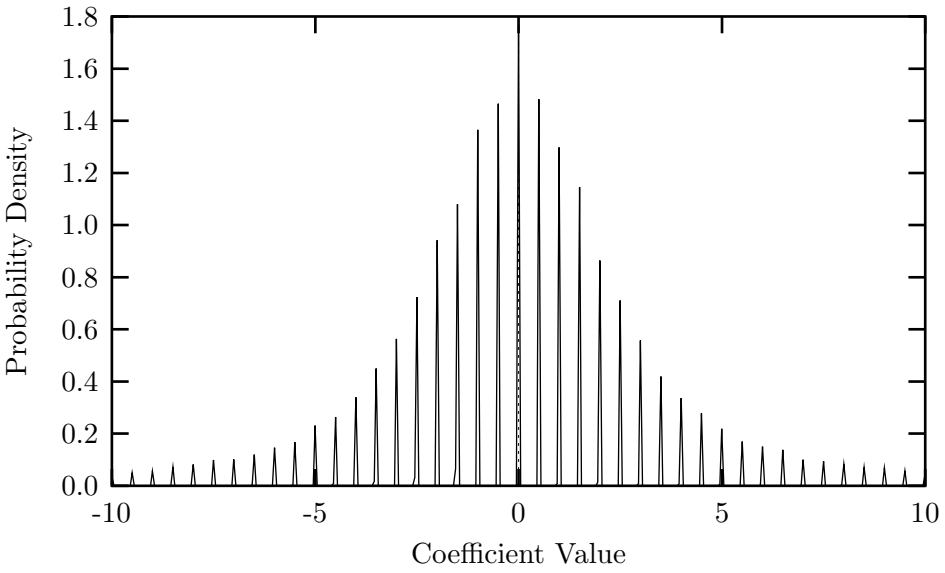


Figure 6.1: Histogram of Haar transform coefficient values at resolution 8 of the horizontal directional subband of the "Lena" image (the bin size is sufficiently small for the pre-quantisation effect to be visible).

6.1.2 Child-parent ratios

Each detail coefficient at a particular resolution in an exactly self-affine signal model is determined by a scaling factor multiplied by a detail coefficient one level up the detail coefficient tree. It is therefore reasonable to consider the histogram shape of the ratios of child and parent detail coefficients as an indication of a reasonable pdf for the scaling coefficients in these model. These histograms were found to be approximately Laplacian for both bases. Since a range subtree may be represented in terms of any domain subtree, an individual coefficient may be determined by an appropriate scaling of any coefficient in the parent subband. Similar ratios were therefore also calculated between each detail coefficient and all detail coefficients in the parent subband. As before, the histogram distributions were approximately Laplacian for both bases, providing some justification for using Laplacian scaling coefficients in the exactly self-affine models.

Additional calculations for the "Lena" image in the Haar basis reveal details of the magnitude decay of coefficients within each image, which is relevant to the contractivity of the representation of a range subtree by a domain subtree. The standard deviation of the child-parent ratio was approximately 0.80, with an average of 72% of the detail coefficients having

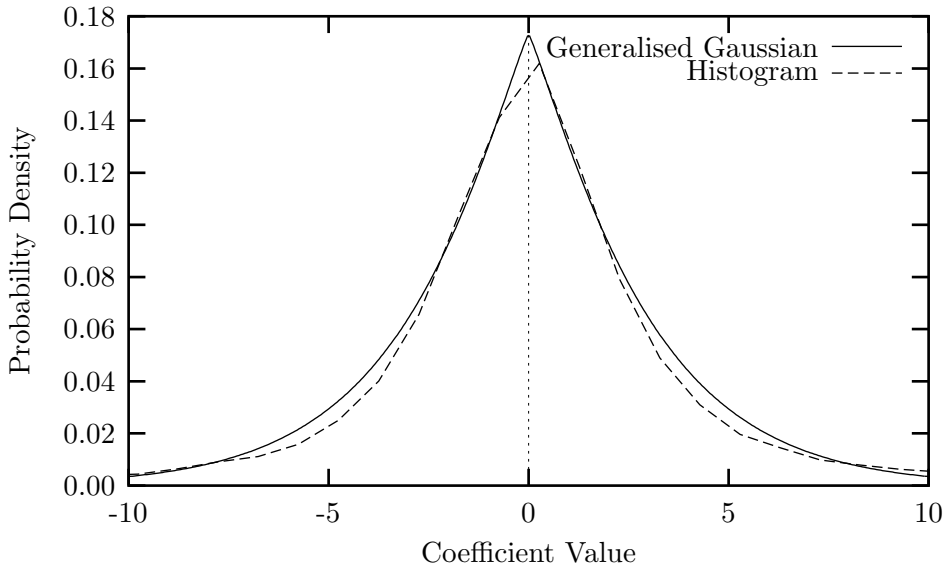


Figure 6.2: Histogram of Haar transform coefficient values at resolution 8 of the horizontal directional subband of the "Lena" image (the pre-quantisation effect is smoothed out by the use of a large bin size). The best fit generalised Gaussian has parameters $\sigma = 3.51$ and $\nu = 1.14$.

smaller magnitudes than their parents. The same average for the ratio between child coefficient and all coefficients in the parent subband was slightly lower, at 66%. Similar results were obtained in the spline basis. The high proportion of child coefficients of smaller magnitude than their parents suggest that the optimum scaling of a domain subtree in matching a range subtree would in most cases have magnitude less than unity, a requirement which is often imposed in order to guarantee contractivity.

6.1.3 Variance decay with increasing resolution

The variance[3] of detail coefficients at each resolution (averaged over all three directional subbands) is displayed on a log scale in Figure 6.3, in which the decay may be observed to be approximately linear for resolutions 3 and higher. If the generating process is assumed to be wide-sense stationary, the measured variance of detail coefficients within each subband of a single image may be taken as an approximation of the variance for that subband over an image ensemble.

Empirical results indicate that there is no large difference in the variance decays with increasing resolution between the three directional subbands (see Figure 6.4), except that the variance in the diagonal subband is consistently the lowest at each resolution. Similar results were obtained for the spline basis, although the measured variances were slightly lower.

The detail coefficient variance decay measured for the test images may be compared with

[3]Since the mean for each subband is close to zero, the measured variances are approximately equivalent to image *energy* [37].

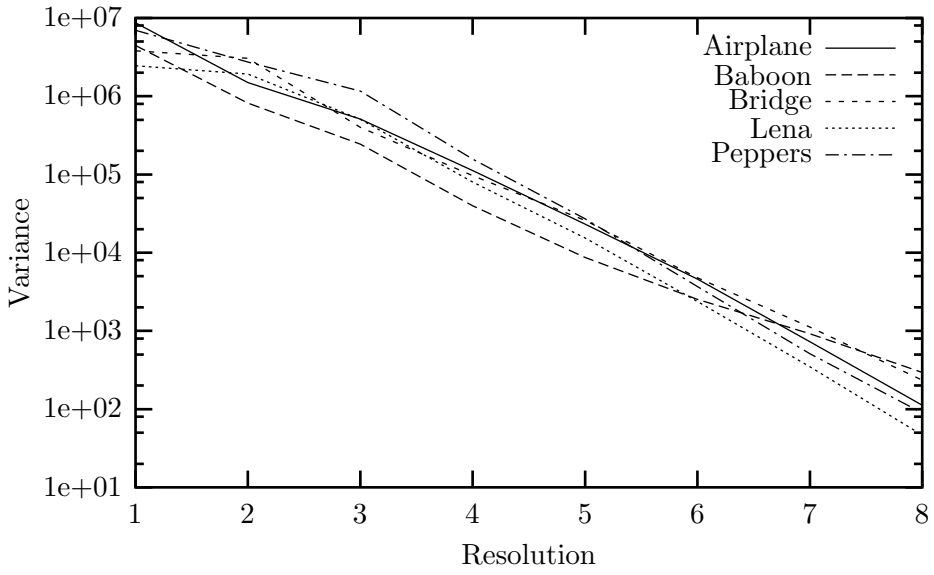


Figure 6.3: Comparison of average Haar basis detail coefficient variance for the set of standard images.

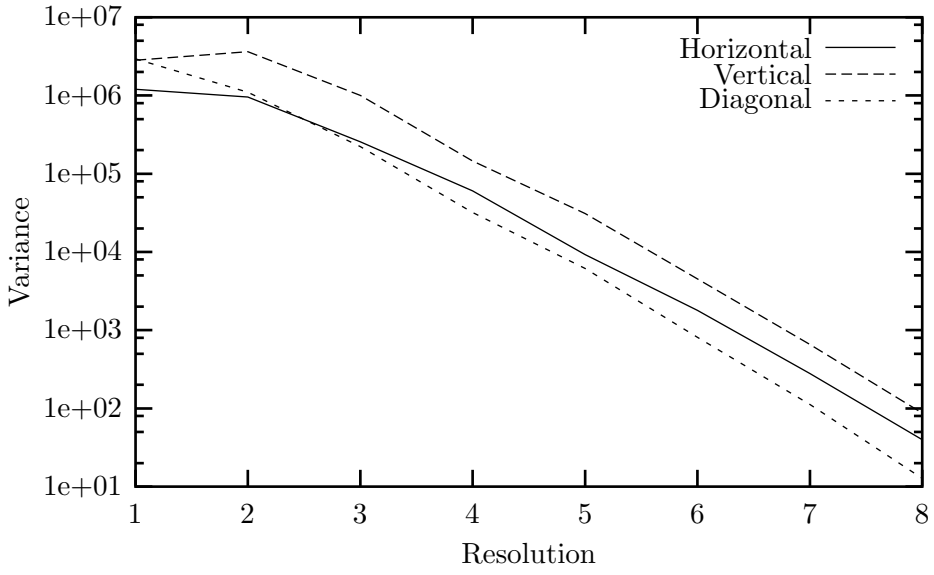


Figure 6.4: Comparison of average Haar basis detail coefficient variance in each of the directional subbands of the "Lena" image.

that resulting from the assumption of exact self-affinity by constructing a two-dimensional extension of a wavelet domain self-affine model. Since isometry operations are disallowed, there is no interaction between the three directional subbands of the non-standard decomposition, and it is therefore only necessary to model one of them, in which there are 4^{i_d} domain subtrees and 4^{i_r} range subtrees rooted at resolutions i_d and $i_r = i_d + 1$ respectively.

Since the normalisation factor between resolutions is 2 for the two-dimensional case, the value of a coefficient determined by a path of length n, originating at domain block l_0 and subsequently passing through ranges $l_1, l_2, \ldots, l_n$, is $\left(\frac{1}{2}\right)^n o_{l_0} s_{l_1} s_{l_2} \ldots s_{l_n}$, and its square is just $\left(\frac{1}{4}\right)^n o_{l_0}^2 s_{l_1}^2 s_{l_2}^2 \ldots s_{l_n}^2$. The expectation for a path belonging to a particular class is $\frac{\sigma_o^2}{4^n} \prod_{i=1}^n m_{2i}^{h_i}$, by an equivalent argument to that presented in Section 5.5.1.

The results of the previous chapter may be applied, although in this case $k = 4^{i_d+1}$, $N_P = 4^n \frac{k}{4}$, $N_M = \left(\frac{k}{4}\right)^k$ and there are 4 equivalent transitions in the enumeration N_U of unspecified p_i. The coefficients of $\frac{\sigma_o^2}{4^n} \prod_{i=1}^n m_{2i}^{h_i}$ for classes $(k-1, 0, 0, \ldots, 0, 1)$ and $(k-n, n, 0, 0, \ldots, 0)$ are

$$\frac{N_C(k-1, 0, \ldots, 0, 1) N(l_1, \ldots, l_n)}{N_P N_M} = \frac{4^{2-n}}{k}.$$

and

$$\frac{N_C(k-n, n, 0, \ldots, 0) N(l_1, \ldots, l_n)}{N_P N_M} = \frac{k!}{(k-n)! k^n}$$

respectively.

Since measurements on standard images indicate that the Laplacian distribution is reasonably appropriate for the marginal distributions of most wavelet coefficients, as well as for child-parent ratios, it is utilised here for the o_i and s_i. The coefficients of $\sigma_o^2 \left(\frac{\sigma_s^2}{4}\right)^n$ for classes $(k-1, 0, 0, \ldots, 0, 1)$ and $(k-n, n, 0, 0, \ldots, 0)$ are therefore

$$\frac{(2n)! 4^{2-n}}{2^n k}$$

and

$$\frac{k!}{(k-n)! k^n}$$

respectively. These coefficients provide bounds on the effects of higher-order moments on the variance at different resolutions of the exactly self-affine signal models. Values for these coefficients for $i_d = 4$ are displayed in Tables 6.1 and 6.2 respectively. The deviation from log decay of a two-dimensional model is considerably reduced from that of a one-dimensional version (see Tables 5.3 and 5.4) with the same values of i_d and n, as a result of the increased value of k associated with a particular value of i_d.

Figures 6.5 and 6.6 show the results of Monte Carlo based calculations of the variance decay for domain subtrees rooted as resolutions 3 and 4 respectively. Although there is a significant deviation from linearity for Laplacian scaling coefficients in Figure 6.5, the increase in k resulting from an increase in i_d from 3 to 4 is sufficient to render this deviation negligible,

Table 6.1: Values of coefficients of $\sigma_o^2 \left(\frac{\sigma_s^2}{4}\right)^n$ for class $(k-1, 0, 0, \ldots, 0, 1)$ and $k = 1024$.

n	1	2	3	4	5
Coefficient	0.0000	0.0059	0.0220	0.1538	1.7303

Table 6.2: Values of coefficients of $\sigma_o^2 \left(\frac{\sigma_s^2}{4}\right)^n$ for class $(k-n, n, 0, 0, \ldots, 0)$ and $k = 1024$.

n	1	2	3	4	5
Coefficient	1.0000	0.9990	0.9971	0.9942	0.9903

as may be observed in Figure 6.6. The decay for uniformly distributed scaling coefficients is also displayed for purposes of comparison.

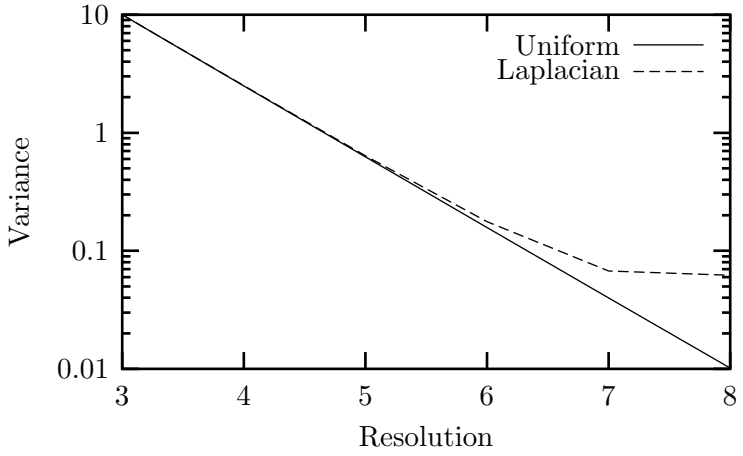


Figure 6.5: Comparison of variance decay for two-dimensional exactly self-affine models with uniform and Laplacian scaling coefficients ($i_d = 3$).

It is possible therefore, by the imposition of suitable restrictions, for the expected variance at resolution $i_d + i$ of an exactly self-affine model to be approximately $\sigma_o^2 \left(\frac{\sigma_s^2}{4}\right)^i$, which is linear in i on a log scale, and is compatible with the measured variance decay of the test images. The primary restriction required is that the domain subtree roots are at resolution 4 or higher, which is equivalent to setting a maximum range block size of 16×16 pixels in a 512×512 image. The best fit parameters for each of the images are displayed in Table 6.3, which illustrates the reduced variance of both coefficients in the spline basis.

The correspondence between measured variance decay for the test images and that of the self-affine models reflects favourably on the possibility of compatibility of the property of exact self-affinity with the statistics of natural images. It is however, difficult to quantify the degree of mismatch between a self-affine model and the test images based on their respective second order statistics, especially in terms of the consequences of this mismatch on image coding based on such a model. In addition, evidence was presented in Chapter 5 that an exactly

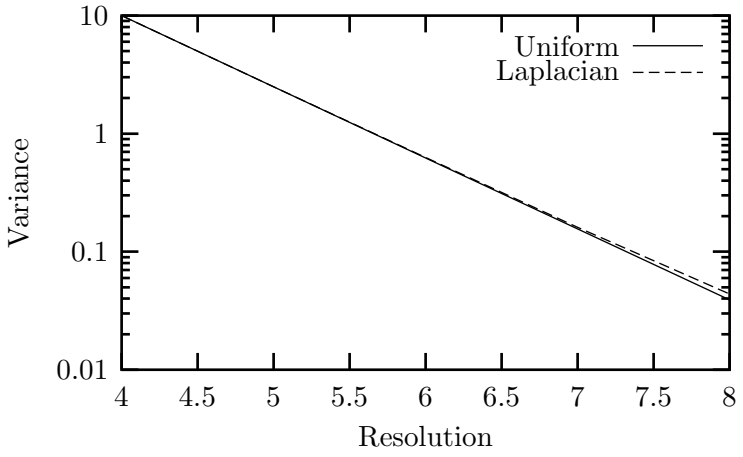


Figure 6.6: Comparison of variance decay for two-dimensional exactly self-affine models with uniform and Laplacian scaling coefficients ($i_d = 4$).

Table 6.3: Optimum parameters in fitting variance decay function $\sigma_o^2 \left(\frac{\sigma_s^2}{4}\right)^i$ to resolution $i_d + i$ of the set of standard images ($i_d = 4$).

Image	Haar basis		Spline basis	
	σ_o^2	σ_s^2	σ_o^2	σ_s^2
Airplane	1.12e5	0.82	9.50e4	0.79
Baboon	3.99e4	0.90	2.86e4	1.10
Bridge	9.70e4	1.03	8.50e4	0.92
Lena	7.98e4	0.76	8.19e4	0.49
Peppers	1.58e5	0.68	1.35e4	0.48

self-affine model is poorly characterised by second order statistics. Direct examination of the self-affinity of the test images is therefore required to determine the extent to which they exhibit this property. The measurement of correlations between detail coefficients is described in the following sections, in preparation for an evaluation of the domain pool structure and self-affinity of the test images.

6.1.4 Child-parent correlations

Correlation coefficient measurements between each child detail coefficient and its parent, averaged over all images in the test set, are displayed in Table 6.4. The correlations in the diagonal subbands may be observed to be significantly smaller than in the horizontal and vertical subbands, for both bases. The significantly smaller correlations in the spline basis are evidence for its improved decorrelation efficiency over that of the Haar basis.

Table 6.4: Child-parent correlation coefficients averaged over the set of standard images.

Parent Resolution	Haar basis			Spline basis		
	Correlation in subband			Correlation in subband		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
2	0.27	0.17	0.09	0.10	0.05	0.06
3	0.32	0.35	0.10	0.14	0.13	0.00
4	0.43	0.38	0.10	0.08	0.12	-0.02
5	0.41	0.41	0.10	0.12	0.10	-0.02
6	0.42	0.44	0.11	0.11	0.14	-0.01
7	0.46	0.47	0.12	0.13	0.17	0.00

These results are compatible with those of Shapiro [182], who found negligible correlations between parent and child detail coefficients for a smooth wavelet basis, but significant correlations between the *squares* of parent and child coefficients.

The similarities between subbands of the Haar decomposition of a scan line extracted from the image indicated in Figure 6.7 are clearly indicated in Figure 6.8. The similarity in the spline basis depicted in Figure 6.9 is considerably reduced, although the progression of maxima across resolutions [125] [126] is still visible.

6.1.5 Same-resolution correlations

Average correlation coefficients *within* each subband were calculated separately for the three families of directions depicted in Figure 6.10. Average results over all test images are tabulated in Tables 6.5 and 6.6. In the Haar basis, significant correlations may be observed in the horizontal direction in the horizontal subband and in the vertical direction in the vertical subband, while correlations in the diagonal direction are considerably smaller in all subbands. A similar pattern is evident for the spline basis, but with smaller correlations overall, and significant negative correlations in the vertical direction in the horizontal subband and in the horizontal direction in the vertical subband. Same-resolution correlations appear to be

Figure 6.7: Position of segment of scan line extracted from "Lena" image.

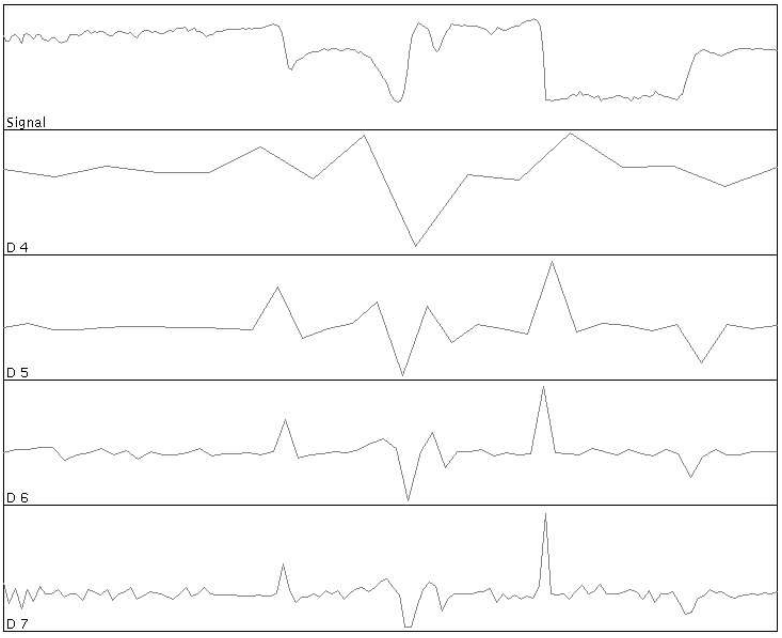


Figure 6.8: Haar basis decomposition of scan line indicated in Figure 6.7. Subbands at resolutions 4 to 7 are indicated by D4 to D7. Each subband has been scaled so that its extrema occupy the full vertical range.

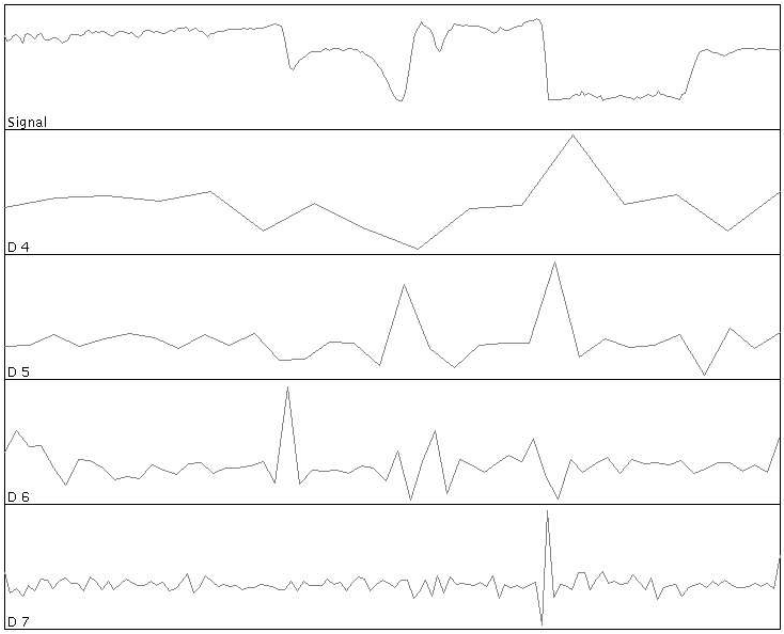


Figure 6.9: Spline basis decomposition of scan line indicated in Figure 6.7. Subbands at resolutions 4 to 7 are indicated by D4 to D7. Each subband has been scaled so that its extrema occupy the full vertical range.

of similar significance to child-parent correlations in the Haar basis, but the decorrelation efficiency of the spline basis is significantly greater for child-parent coefficients than for same-resolution coefficients.

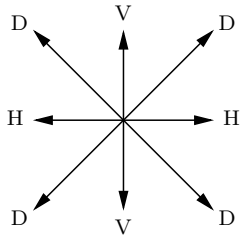


Figure 6.10: Directions of correlation measurement.

6.2 Self-Affinity and Domain Pool Structure

Since it is difficult to determine how compatible the assumption of self-affinity is with the properties of natural images based on standard statistical measures, it is necessary to directly estimate the degree of self-affinity of the set of test images. This is achieved by defining a subtree codebook performance measure similar to the subblock codebook performance measure defined in Section 4.4.

Table 6.5: Same-resolution (Haar basis) correlation coefficients averaged over the set of standard images.

Resolution	Horizontal Subband			Vertical Subband			Diagonal Subband		
	H	V	D	H	V	D	H	V	D
3	0.24	-0.23	-0.04	-0.35	0.47	-0.22	-0.24	-0.13	-0.05
4	0.43	-0.10	-0.01	-0.18	0.40	-0.06	-0.12	-0.18	0.02
5	0.46	-0.10	-0.01	-0.15	0.47	-0.04	-0.24	-0.21	0.09
6	0.42	-0.07	0.02	-0.15	0.44	-0.00	-0.21	-0.20	0.06
7	0.46	-0.08	0.02	-0.09	0.46	0.04	-0.21	-0.20	0.07
8	0.54	-0.05	0.04	-0.08	0.51	0.04	-0.16	-0.14	0.09

Table 6.6: Same-resolution (spline basis) correlation coefficients averaged over the set of standard images.

Resolution	Horizontal Subband			Vertical Subband			Diagonal Subband		
	H	V	D	H	V	D	H	V	D
3	0.28	-0.20	-0.12	-0.28	0.23	-0.10	0.05	-0.14	0.01
4	0.37	-0.23	-0.06	-0.25	0.35	-0.16	0.00	-0.07	0.02
5	0.32	-0.24	-0.07	-0.20	0.35	-0.09	-0.01	-0.13	0.05
6	0.30	-0.20	-0.08	-0.24	0.29	-0.11	-0.04	-0.06	0.03
7	0.32	-0.24	-0.08	-0.22	0.32	-0.09	-0.09	-0.11	0.04
8	0.43	-0.28	-0.11	-0.34	0.38	-0.14	-0.09	-0.09	0.09

6.2.1 Self-affinity

The degree to which a range subtree vector $\mathbf{r}$ is a multiple of a domain subtree vector $\mathbf{d}$ may be measured by the deterministic correlation coefficient[4]

$$\varrho(\mathbf{d}, \mathbf{r}) = \frac{\langle \mathbf{d}, \mathbf{r} \rangle}{\|\mathbf{d}\| \, \|\mathbf{r}\|},$$

which is equivalent to the cosine of the angle between the subtree vectors. The calculation of ϱ between a domain subtree at resolution 2 and a range subtree at resolution 3 of a one-dimensional signal is depicted in Figure 6.11. A shape codebook efficiency measure may be based on this quantity as in Chapter 4, but in this case, since such a measure is required for individual images, rather than over an ensemble, it is defined as

$$\mathcal{S} = \frac{1}{N_r} \sum_{j=0}^{N_r - 1} \max_{0 \le i \le N_d - 1} \varrho^2(\mathbf{d}_i, \mathbf{r}_j),$$

where $\mathbf{d}_i$ and $\mathbf{r}_j$ are the ith domain and jth range subtrees in an image respectively. The value of this measure for $\mathbf{d}_i$ extracted from the same image as the $\mathbf{r}_i$ is denoted $\mathcal{S}_{\text{self}}$. As described in

[4]In this case, since the domain vector is transformed by scaling alone, without the addition of an offset vector, this is the deterministic equivalent of the normalised correlation, rather than the normalised covariance as used in Chapter 4.

Section 4.4, the classifications of weak and strong self-affinity are based on comparisons with the values of $\mathcal{S}_{\text{random}}$ and $\mathcal{S}_{\text{VQ}}$ computed for appropriately constructed random and optimised sets of $\mathbf{d}_i$ respectively.

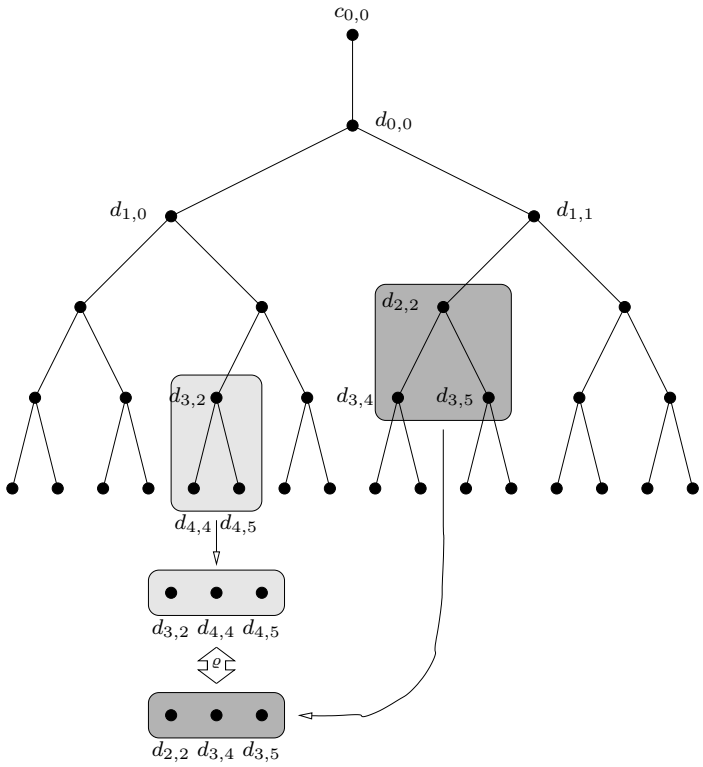


Figure 6.11: Calculation of deterministic correlation between subtrees for a one-dimensional signal.

A statistical analysis of self-affinity is complicated by the maximisation in the definition of the codebook efficiency $\mathcal{S}$, since requiring a high efficiency places constraints on the maximum rather than directly on the ϱ^2 values. A perverse relationship between subtrees as in an exactly self-affine model is possible, where all domains are mutually dependent, and "cooperate" so that one of them is a perfect match for each range. Such dependence may be completely transparent to any examination of the individual ϱ^2 values, making it difficult to detect other than by its effect in raising the maximum ϱ^2 value.

Given the evidence for decaying dependence with increasing distance between pixels [37, ch. 2], this form of domain pool structure appears highly unlikely, relying as it does on strong dependence between spatially remote domains. In the absence of evidence to the contrary, it is reasonable to assume a qualitatively similar domain pool structure to that of the AR(1) models, in which no "special relationship" exists except between nearby domains and ranges. Forms of inter-pixel dependence other than correlation may be present in images, and may play a role in determining the similarity of neighbouring domains and ranges. The domain pool structure of the test images was therefore examined for evidence contradicting this assumption.

6.2.2 Domain pool structure

The wavelet transform domain equivalent of standard fractal compression entails mappings between domain and range subtrees, consisting of the combined subtrees at the same position and resolution from each of the three directional subbands. Since different correlation properties were observed in the different directional subbands, the deterministic correlation between subtrees was investigated for subtrees extracted from separate directional subbands as well as for combined subtrees. These comparisons were performed for domain and range subtrees rooted at several resolutions, corresponding to different block sizes in standard fractal compression.

The dependence of ϱ and ϱ^2 on the distance between range and domain subtrees was investigated by computing a deterministic correlation surface for each range. This surface was constructed in a coordinate system with the range at the centre, the surface height at each position being the deterministic correlation (or square thereof) between the range subtree and the domain subtree (rooted one level up the tree from the range subtree) and at the position corresponding to the appropriate relative displacement from the position of the range subtree, i.e. the domain subtree with root at position (j_d, k_d) and the range subtree at position (j_r, k_r) determined the surface height at position $(j_r/2 - j_d, k_r/2 - k_d)$. An average surface over all range blocks was obtained by averaging each of these surfaces. Normalisation is required in the calculation of an average surface over all ranges since the ensemble sizes (see Figure 6.12) decrease with increasing offset from the range subtree (large offsets in all directions are only possible from range subtrees near the centre of an image because of the existence of image boundaries). The outer regions of these average surfaces are discarded in the surfaces displayed here, due to unstable averages resulting from the small ensemble sizes near the perimeter.

Average deterministic correlation surfaces for separate directional subbands reveal considerable structure in the horizontal direction in the horizontal directional subband (see Figure 6.13) and in the vertical direction in the vertical directional subband (see Figure 6.14), but not in the diagonal subband. This structure may be attributed to the corresponding same-resolution correlations described in Section 6.1.5, and is evidence of directional similarity between neighbouring subtrees. Similar structure was observed for average ϱ^2 surfaces, but the peaks at the origin are not as consistently significant for all images.

The structure visible in average ϱ^2 surfaces for combined directional subtrees appears to be a result of the combined effects of the structure in the individual directional subbands, although this not as clearly apparent for all cases as it is in Figure 6.15. The surface in Figure 6.15 is evidence of considerably raised similarity between a range subtree and its parent subtree, as well as less significant similarity between a range subtree and domain subtrees at small horizontal and vertical offsets. The mean and central peak values of ϱ and ϱ^2 for the "Airplane", "Lena" and "Peppers" images are summarised in Tables 6.7, 6.8

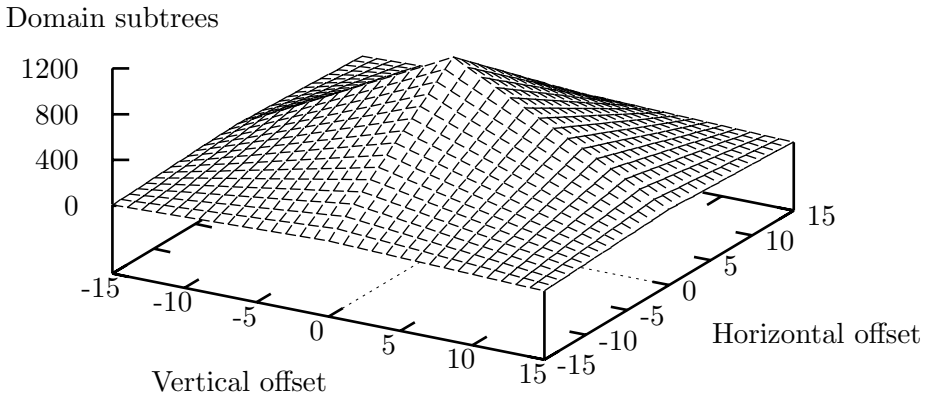


Figure 6.12: Variation with offset from the range subtree of the number of domain subtrees in the ensemble at resolution 4. The centre of the surface corresponds to a zero offset in both directions.

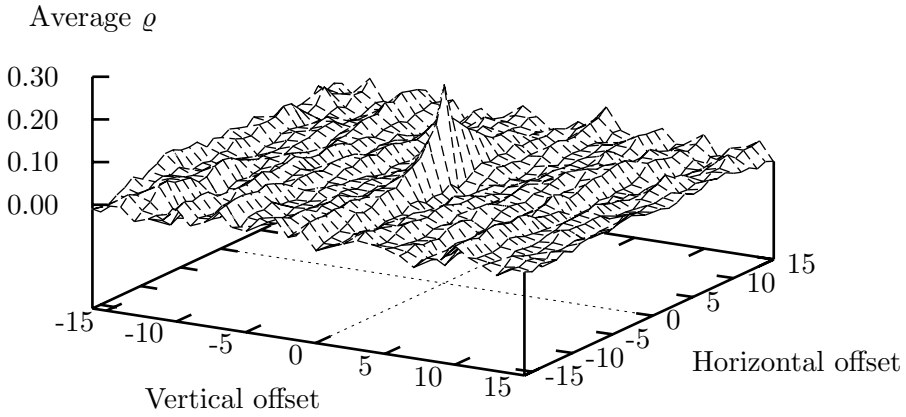


Figure 6.13: Average deterministic correlation surface for the horizontal subband at resolution 5 of the non-standard Haar basis decomposition of the "Airplane" image.

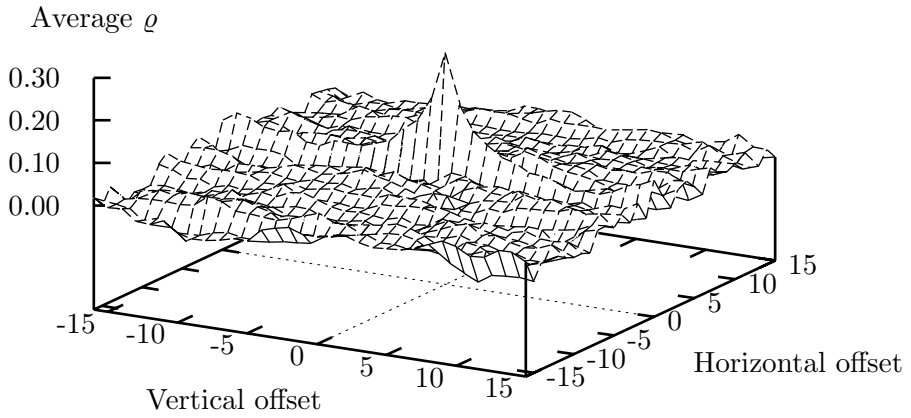


Figure 6.14: Average deterministic correlation surface for the vertical subband at resolution 5 of the non-standard Haar basis decomposition of the "Lena" image.

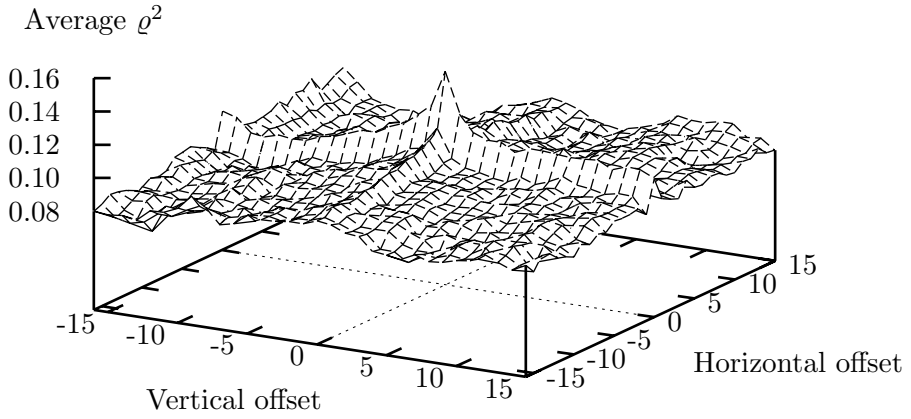


Figure 6.15: Average squared deterministic correlation surface for combined directional subtrees with domain subtrees rooted at resolution 5 of the non-standard spline basis decomposition of the "Airplane" image.

and 6.9 respectively. The ϱ average of approximately zero indicates that there is no average tendency for distant domains to have the same direction as ranges[5], while the significant positive value for the central peak indicates that the domain subtree actually containing a corresponding range subtree is likely to have a similar vector direction. The increase in peak and mean ϱ^2 values with increasing i_r is expected as a result of the corresponding decrease in subtree vector dimensionality. The significant difference between peak and mean values of ϱ^2 indicates that there is a raised probability of a range matching its parent domain in comparison with the probability of matching other domains. This central peak may be at least partially attributed to the child-parent correlations described in Section 6.1.4 (especially since the peaks are slightly less significant in the spline basis, for which child-parent correlations were lower than in the Haar basis), but other forms of dependence between detail coefficients may also play a role. There is no evidence to suggest that the domain pool structure is not qualitatively similar to that of the AR(1) models, with only those domains having raised average ϱ^2 having any significant "special relationship" with the ranges, although this is certainly not proved by the results presented here.

Table 6.7: Central peak and mean deterministic correlation surface values for range subtrees in the "Airplane" image.

	Haar basis				Spline basis			
	ϱ		ϱ^2		ϱ		ϱ^2	
i_r	Mean	Peak	Mean	Peak	Mean	Peak	Mean	Peak
4	0.01	0.17	0.06	0.12	0.01	0.10	0.06	0.12
5	0.00	0.20	0.08	0.15	0.00	0.06	0.07	0.14
6	0.00	0.21	0.09	0.16	0.00	0.04	0.09	0.15
7	0.00	0.24	0.12	0.21	0.00	0.02	0.11	0.18

Table 6.8: Central peak and mean deterministic correlation surface values for range subtrees in the "Lena" image.

	Haar basis				Spline basis			
	ϱ		ϱ^2		ϱ		ϱ^2	
i_r	Mean	Peak	Mean	Peak	Mean	Peak	Mean	Peak
4	0.02	0.21	0.10	0.16	0.01	0.10	0.11	0.17
5	0.01	0.25	0.12	0.20	0.00	0.10	0.11	0.20
6	0.01	0.28	0.13	0.23	0.00	0.05	0.10	0.18
7	0.00	0.30	0.13	0.27	0.00	0.04	0.10	0.17

Calculation of the distribution of best matching domain subtrees with spatial offset from the matched range revealed no large predominance (after taking into account the variation

[5]This does not preclude clustering in particular directions, but indicates that any clustering is symmetric with respect to negative scaling.

Table 6.9: Central peak and mean deterministic correlation surface values for range subtrees in the "Peppers" image.

	Haar basis				Spline basis			
	ϱ		ϱ^2		ϱ		ϱ^2	
i_r	Mean	Peak	Mean	Peak	Mean	Peak	Mean	Peak
4	0.01	0.19	0.09	0.13	0.01	0.04	0.08	0.11
5	0.01	0.28	0.11	0.20	0.00	0.06	0.09	0.16
6	0.01	0.34	0.12	0.24	0.00	0.05	0.08	0.15
7	0.01	0.32	0.11	0.24	0.00	0.04	0.08	0.15

in ensemble size with offset) of spatially close domains except for the "Lena" and "Peppers" images. There is no obvious explanation for this fact, but it should be noted that the average ϱ^2 surface peaks were most clearly visible for these two images[6].

6.3 Random Codebooks and Weak Self-Affinity

A random codebook suitable for use in determining weak self-affinity is required to have the same probability distribution as the individual domains in the source for which a comparison is desired. While construction of such a codebook was easily achieved in the case of AR(1) models, where the distribution is known analytically, it is considerably more difficult for sources for which the distribution is unknown and can only be estimated from an ensemble of signals.

6.3.1 Codebook construction

An approximation to a suitable random codebook may be constructed by randomly generating independent detail coefficients with a Laplacian distribution and variance in each subband as measured in the corresponding subband of the test images. This construction is suboptimal as a random codebook, since it does not take into account the correlations observed between detail coefficients, or any other form of dependence.

A conceptually similar random codebook was constructed by van de Walle [194], who generated random code vectors based on image second order statistics and compared the performance of these code vectors with that of a domain pool in representing smooth, textured and edge range blocks. The performance of the random codebook was found to match that of the domain pool exactly for smooth range blocks, while the domain pool was slightly superior for textured ranges and significantly superior for ranges containing edges. These results suggest that the statistical properties of images are well characterised by second order

[6] A similar highly peaked distribution of matching domain blocks is described by Hürtgen and Stiller [91, pg. 401]

statistics, except in the vicinity of edges where significant non-linear dependence is present [194].

An alternative approximation to the desired random codebook may be constructed by extracting domains from a different image than the one from which the ranges are extracted. This construction should provide a closer approximation to the domain probability distributions than that based on independent detail coefficients, although its use as a random codebook entails an implicit assumption of stationarity and ergodicity.

6.3.2 Codebook efficiency comparisons

Comparisons between the domain pool efficiency and that of the two approximations to a random codebook are presented in Tables 6.10 to 6.15. The domain pool details are labelled as "self codebook", the independent Laplacian codebook is labelled "variance codebook" and the separate image based random codebook is labelled with the name of that image. Domain and range subtrees were rooted at resolutions i_d and $i_r = i_d + 1$ respectively for all codebooks. The performance of the codebook constructed from a separate image was on average significantly better than that of the codebook constructed from independent detail coefficients, which suggests that the former codebook is a better approximation to the desired codebook, as expected.

Table 6.10: Comparison of the self codebook and two random codebooks for Haar basis range subtrees in the "Airplane" image.

	Self Codebook			"Baboon" Codebook			Variance Codebook		
	ϱ^2			ϱ^2			ϱ^2		
i_r	Mean	Max	MSE	Mean	Max	MSE	Mean	Max	MSE
4	0.06	0.38	634.50	0.04	0.24	748.23	0.04	0.20	801.16
5	0.08	0.46	306.66	0.04	0.35	394.37	0.04	0.26	486.13
6	0.09	0.57	99.77	0.06	0.51	132.03	0.06	0.43	202.24
7	0.12	0.79	16.60	0.09	0.77	23.48	0.10	0.74	41.96

Table 6.11: Comparison of the self codebook and two random codebooks for spline basis range subtrees in the "Airplane" image.

	Self Codebook			"Baboon" Codebook			Variance Codebook		
	ϱ^2			ϱ^2			ϱ^2		
i_r	Mean	Max	MSE	Mean	Max	MSE	Mean	Max	MSE
4	0.06	0.35	506.04	0.03	0.20	601.69	0.04	0.23	574.06
5	0.07	0.43	238.01	0.03	0.32	301.54	0.05	0.32	309.43
6	0.09	0.54	79.65	0.05	0.48	100.54	0.07	0.49	110.00
7	0.11	0.78	11.88	0.08	0.76	16.24	0.11	0.77	18.29

Table 6.12: Comparison of the self codebook and two random codebooks for Haar basis range subtrees in the "Lena" image.

	Self Codebook			"Peppers" Codebook			Variance Codebook		
	ϱ^2			ϱ^2			ϱ^2		
i_r	Mean	Max	MSE	Mean	Max	MSE	Mean	Max	MSE
4	0.10	0.45	327.25	0.09	0.42	346.41	0.06	0.28	436.35
5	0.12	0.53	131.56	0.11	0.52	140.76	0.06	0.33	233.66
6	0.13	0.57	46.67	0.13	0.57	47.52	0.08	0.44	90.94
7	0.13	0.76	9.79	0.13	0.76	10.47	0.10	0.72	19.64

Table 6.13: Comparison of the self codebook and two random codebooks for spline basis range subtrees in the "Lena" image.

	Self Codebook			"Peppers" Codebook			Variance Codebook		
	ϱ^2			ϱ^2			ϱ^2		
i_r	Mean	Max	MSE	Mean	Max	MSE	Mean	Max	MSE
4	0.11	0.42	254.92	0.08	0.39	276.95	0.08	0.37	278.27
5	0.11	0.49	103.38	0.10	0.49	105.93	0.08	0.40	132.49
6	0.10	0.51	36.23	0.10	0.50	38.03	0.08	0.46	48.73
7	0.10	0.73	7.19	0.09	0.72	7.50	0.09	0.72	8.79

Table 6.14: Comparison of the self codebook and two random codebooks for Haar basis range subtrees in the "Peppers" image.

	Self Codebook			"Airplane" Codebook			Variance Codebook		
	ϱ^2			ϱ^2			ϱ^2		
i_r	Mean	Max	MSE	Mean	Max	MSE	Mean	Max	MSE
4	0.09	0.43	570.82	0.06	0.34	682.79	0.06	0.30	739.50
5	0.11	0.54	195.70	0.08	0.47	252.61	0.06	0.33	378.77
6	0.12	0.55	58.15	0.09	0.52	72.97	0.07	0.42	142.50
7	0.11	0.71	14.11	0.10	0.70	15.64	0.08	0.66	33.54

Table 6.15: Comparison of the self codebook and two random codebooks for spline basis range subtrees in the "Peppers" image.

	Self Codebook			"Airplane" Codebook			Variance Codebook		
	ϱ^2			ϱ^2			ϱ^2		
i_r	Mean	Max	MSE	Mean	Max	MSE	Mean	Max	MSE
4	0.08	0.40	425.17	0.06	0.34	473.71	0.07	0.34	464.75
5	0.09	0.47	145.16	0.06	0.43	167.74	0.07	0.38	206.41
6	0.08	0.46	45.45	0.07	0.44	53.60	0.07	0.41	77.41
7	0.08	0.68	11.24	0.08	0.67	12.23	0.08	0.64	18.12

Mean ϱ^2 values are slightly higher for the domain pool than the other codebooks, but this may be due to the the central peak raising the average value of the domain pool ϱ^2 surfaces (the peak is obviously not present for the other codebooks), rather than a general superiority in matching ability for all self codebook domains. The maximum ϱ^2 for the self codebook is on average slightly higher than that for the separate image codebook, with particularly large differences for the lower resolution subtree roots. The superiority of the self codebook is likely to be due to to the similarity between neighbouring domains and ranges, but the extent to which it is observed may have been enhanced by the difficulty in constructing an appropriate random codebook.

Consider creating two sequences of shape codebooks of increasing size, one by optimising each codebook in the sequence, and the other by simply including an additional random vector. Although the former is always more efficient, their performance measured in terms of maximum ϱ^2 would both approach unity asymptotically[7], and one may therefore expect that the maximum ϱ^2 performance measures of these types of codebook converge with increasing codebook size. An additional effect in this case is the decreasing vector dimensionality with increasing subtree root resolution, which results in further convergence of the performance measure. It is not clear whether the observed convergence in performance of the three codebooks is largely due to this effect, or a result of the decreased influence of the local similarity between subtrees in larger codebooks. The increase in maximum ϱ^2 with increasing i_r for all three codebooks is due to the combined effects of increasing codebook size and decreasing vector dimensionality; the former increasing the set over which maximisation occurs and the latter increasing the minimum possible angle between any two vectors.

Average MSE distortion values were also calculated for the representation of each range subtree by the optimum scaling of the best domain subtree. The codebook effectiveness rankings in terms of the deterministic correlation correspond well with those in terms of this distortion, which is equivalent to spatial domain collage error for the Haar basis. Spatial domain distortions do not correspond to detail coefficient distortions in the spline basis, which is not orthogonal.

While conclusions are not as reliable as those for AR(1) models, due to practical limitations on ensemble sizes and in the construction of an appropriate random codebook, the test images appear to be weakly self-affine to a significant extent, especially for the lower resolution subtree roots. There is no convincing evidence however, that this self-affinity is due to any effect other than the statistical dependence between neighbouring regions of a signal, as observed for AR(1) models. Considerable further investigation is required to conclusively settle this issue. A comparison with a suitably optimised codebook is required to determine whether the test images are strongly self-affine.

[7]The larger the codebook therefore, the more difficult it is to determine the existence of significant self-affinity.

6.4 Vector Quantisation and Strong Self-Affinity

The degree to which the test images are strongly self-affine was evaluated by a comparison of the domain pool performance with that of a fixed subtree codebook, constructed by application of the Generalised Lloyd Algorithm (GLA) to subtrees rooted at the range resolution in the test images. Since domain subtrees are matched to range subtrees by scaling, Shape-Gain Vector Quantisation (SGVQ) [73, ch. 12] provides the codebooks appropriate for comparison.

Although comparisons with optimised codebooks have previously been performed [112, ch. 5], the experiments here differ in that

- Evaluation of the self-affinity is of primary interest, and transform coefficients are consequently not quantised.
- Since comparisons are performed in the wavelet domain, SGVQ is used rather than Mean Removed SGVQ [139].
- Comparisons are performed in Haar basis (equivalent to standard fractal coding) and smooth wavelet basis decompositions.

6.4.1 Codebook construction

Although the usual codebook design for SGVQ consists of simultaneous optimisation of the shape and gain codebooks [32] [167], the gain is not quantised in the experiments here (since evaluation of the structural constraints is the primary concern), and an alternative codebook design procedure is selected in order to optimise the shape codebook independent of the gain values.

Maximising average deterministic correlation

The first stage of each iteration of the GLA consists of identifying the training vectors in each Voronoi cell, while the second updates each of the code vectors to be the optimum vector for all of the training vectors in its Voronoi cell. The first stage may easily be modified to define Voronoi cells in terms of the maximum deterministic correlation, rather than the minimum MSE distortion. The second stage is more complex, as it is not immediately clear what the optimum vector for a set of training vectors is when using the deterministic correlation based measure.

Given the N training vectors $\mathbf{t}_0, \mathbf{t}_1, \ldots, \mathbf{t}_{N-1}$ in a Voronoi region, the desired optimum code vector $\mathbf{c}$ maximises the average deterministic correlation with the training vectors by maximising the function

$$\alpha(\mathbf{c}) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle}{\|\mathbf{c}\| \, \|\mathbf{t}_i\|}.$$

The maximum of $\alpha(\mathbf{c})$ occurs when $\nabla_{\mathbf{c}}\alpha(\mathbf{c}) = 0$, and since

$$\nabla_{\mathbf{c}} \frac{\langle \mathbf{c}, \mathbf{t} \rangle}{\|\mathbf{c}\| \, \|\mathbf{t}\|} = \frac{\mathbf{t}}{\|\mathbf{c}\| \, \|\mathbf{t}\|} - \frac{\langle \mathbf{c}, \mathbf{t} \rangle \mathbf{c}}{\|\mathbf{c}\|^3 \, \|\mathbf{t}\|},$$

this evaluates to

$$\sum_{i=0}^{N-1} \frac{\mathbf{t}_i}{\|\mathbf{c}\| \, \|\mathbf{t}_i\|} = \sum_{i=0}^{N-1} \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle \mathbf{c}}{\|\mathbf{c}\|^3 \, \|\mathbf{t}_i\|},$$

which may be written as

$$\sum_{i=0}^{N-1} \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|} = \frac{\mathbf{c}}{\|\mathbf{c}\|} \sum_{i=0}^{N-1} \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle}{\|\mathbf{c}\| \, \|\mathbf{t}_i\|}.$$

Rewriting the right hand side

$$\sum_{i=0}^{N-1} \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle}{\|\mathbf{c}\| \, \|\mathbf{t}_i\|} = \sum_{i=0}^{N-1} \langle \frac{\mathbf{c}}{\|\mathbf{c}\|}, \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|} \rangle = \langle \frac{\mathbf{c}}{\|\mathbf{c}\|}, \sum_{i=0}^{N-1} \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|} \rangle$$

reveals the solution

$$\mathbf{c} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|}$$

to be simply the usual centroid of the set of normalised training vectors. Unfortunately, however, this solution ignores the possibility of a negative scaling coefficient for a subtree, in which case maximising the deterministic correlation between vectors is suboptimal. The appropriate criterion for a match is to maximise the absolute value or the square of the deterministic correlation.

Maximising average absolute value of deterministic correlation

Addressing first the absolute value deterministic correlation criteria, the function to be maximised for the training vectors in a Voronoi region is

$$\alpha(\mathbf{c}) = \frac{1}{N} \sum_{i=0}^{N-1} \left| \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle}{\|\mathbf{c}\| \, \|\mathbf{t}_i\|} \right|.$$

Proceeding as before, since[8]

$$\nabla_{\mathbf{c}} \left| \frac{\langle \mathbf{c}, \mathbf{t} \rangle}{\|\mathbf{c}\| \, \|\mathbf{t}\|} \right| = \operatorname{sign}\langle \mathbf{c}, \mathbf{t} \rangle \left(\frac{\mathbf{t}}{\|\mathbf{c}\| \, \|\mathbf{t}\|} - \frac{\langle \mathbf{c}, \mathbf{t} \rangle \mathbf{c}}{\|\mathbf{c}\|^3 \, \|\mathbf{t}\|} \right),$$

[8]Where

$$\operatorname{sign} x = \begin{cases} -1 & \text{if } x < 0 \\ 1 & \text{otherwise.} \end{cases}$$

the requirement $\nabla_{\mathbf{c}}\alpha(\mathbf{c}) = 0$ for a maximum implies

$$\sum_{i=0}^{N-1} \text{sign}\langle \mathbf{c}, \mathbf{t}_i \rangle \frac{\mathbf{t}_i}{\|\mathbf{c}\| \, \|\mathbf{t}_i\|} = \sum_{i=0}^{N-1} \text{sign}\langle \mathbf{c}, \mathbf{t}_i \rangle \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle \mathbf{c}}{\|\mathbf{c}\|^3 \, \|\mathbf{t}_i\|},$$

which reduces to

$$\sum_{i=0}^{N-1} \text{sign}\langle \mathbf{c}, \mathbf{t}_i \rangle \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|} = \frac{\mathbf{c}}{\|\mathbf{c}\|} \sum_{i=0}^{N-1} \text{sign}\langle \mathbf{c}, \mathbf{t}_i \rangle \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle}{\|\mathbf{c}\| \, \|\mathbf{t}_i\|}.$$

Rewriting the right hand side as before

$$\sum_{i=0}^{N-1} \text{sign}\langle \mathbf{c}, \mathbf{t}_i \rangle \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle}{\|\mathbf{c}\| \, \|\mathbf{t}_i\|} = \sum_{i=0}^{N-1} \langle \frac{\mathbf{c}}{\|\mathbf{c}\|}, \text{sign}\langle \mathbf{c}, \mathbf{t}_i \rangle \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|} \rangle = \langle \frac{\mathbf{c}}{\|\mathbf{c}\|}, \sum_{i=0}^{N-1} \text{sign}\langle \mathbf{c}, \mathbf{t}_i \rangle \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|} \rangle$$

reveals the solution to be

$$\mathbf{c} = \frac{1}{N} \sum_{i=0}^{N-1} \text{sign}\langle \mathbf{c}, \mathbf{t}_i \rangle \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|}.$$

This is unfortunately not a closed form equation, the only obvious technique of finding a solution being to test all possible values of $\text{sign}\langle \mathbf{c}, \mathbf{t}_i \rangle$ for consistency, which is a time-consuming task which entails testing each of 2^N different sign assignments.

An iterative approach may be considered as an alternative solution. One would expect the existing code vector $\mathbf{c}_j$ for a Voronoi region to be reasonably close to the centroid after a few iterations of the GLA, in which case most of the centroids, or new code vectors $\mathbf{c}_{j+1}$, would be such that $\text{sign}\langle \mathbf{c}_j, \mathbf{t}_i \rangle = \text{sign}\langle \mathbf{c}_{j+1}, \mathbf{t}_i \rangle$. A reasonable approach based on this assumption is to define[9]

$$\mathbf{c}_{j+1} = \frac{1}{N} \sum_{i=0}^{N-1} \text{sign}\langle \mathbf{c}_j, \mathbf{t}_i \rangle \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|}.$$

Now

$$\sqrt{\langle \mathbf{c}_{j+1}, \mathbf{c}_{j+1} \rangle} = \frac{1}{N} \sum_{i=0}^{N-1} \text{sign}\langle \mathbf{c}_j, \mathbf{t}_i \rangle \frac{\langle \mathbf{c}_{j+1}, \mathbf{t}_i \rangle}{\|\mathbf{c}_{j+1}\| \, \|\mathbf{t}_i\|},$$

and by comparison with

$$\alpha(\mathbf{c}_{j+1}) = \frac{1}{N} \sum_{i=0}^{N-1} \text{sign}\langle \mathbf{c}_{j+1}, \mathbf{t}_i \rangle \frac{\langle \mathbf{c}_{j+1}, \mathbf{t}_i \rangle}{\|\mathbf{c}_{j+1}\| \, \|\mathbf{t}_i\|}$$

it is clear that

$$\sqrt{\langle \mathbf{c}_{j+1}, \mathbf{c}_{j+1} \rangle} \leq \alpha(\mathbf{c}_{j+1}).$$

[9]This is in fact equivalent to the solution adopted by Lepsøy [112, ch. 3], based on a rather different derivation. Lepsøy's approach however, did not lead to a simple solution for the squared deterministic correlation criterion as presented here.

In addition

$$\langle \mathbf{c}_{j+1}, \mathbf{c}_j \rangle = \frac{1}{N} \sum_{i=0}^{N-1} \text{sign}\langle \mathbf{c}_j, \mathbf{t}_i \rangle \frac{\langle \mathbf{c}_j, \mathbf{t}_i \rangle}{\|\mathbf{t}_i\|}$$

which implies

$$\frac{\langle \mathbf{c}_{j+1}, \mathbf{c}_j \rangle}{\sqrt{\langle \mathbf{c}_j, \mathbf{c}_j \rangle}} = \alpha(\mathbf{c}_j).$$

Applying the Cauchy-Schwarz inequality [186, pg. 185]

$$\langle \mathbf{c}_{j+1}, \mathbf{c}_j \rangle^2 \leq \langle \mathbf{c}_{j+1}, \mathbf{c}_{j+1} \rangle \langle \mathbf{c}_j, \mathbf{c}_j \rangle$$

gives

$$\alpha(\mathbf{c}_{j+1}) \geq \alpha(\mathbf{c}_j).$$

Since each iteration moves the codebook vector closer to its optimum for a particular set of training vectors, the inclusion of this step in the GLA, rather than the assignment of the exact centroid, results in a convergent algorithm.

Maximising average squared deterministic correlation

In the case of the squared deterministic correlation criterion the function to be maximised is

$$\alpha(\mathbf{c}) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle^2}{\|\mathbf{c}\|^2 \, \|\mathbf{t}_i\|^2}.$$

Once again requiring $\nabla_{\mathbf{c}} \alpha(\mathbf{c}) = 0$, and noting

$$\nabla_{\mathbf{c}} \frac{\langle \mathbf{c}, \mathbf{t} \rangle^2}{\|\mathbf{c}\|^2 \, \|\mathbf{t}\|^2} = \frac{2\langle \mathbf{c}, \mathbf{t} \rangle \mathbf{t}}{\|\mathbf{c}\|^2 \, \|\mathbf{t}\|^2} - \frac{2\langle \mathbf{c}, \mathbf{t} \rangle^2 \mathbf{c}}{\|\mathbf{c}\|^4 \, \|\mathbf{t}\|^2},$$

results in the condition

$$\sum_{i=0}^{N-1} \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle \mathbf{t}_i}{\|\mathbf{c}\|^2 \, \|\mathbf{t}_i\|^2} = \sum_{i=0}^{N-1} \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle^2 \mathbf{c}}{\|\mathbf{c}\|^4 \, \|\mathbf{t}_i\|^2}$$

for an optimum $\mathbf{c}$. Expressing this condition as

$$\sum_{i=0}^{N-1} \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle}{\|\mathbf{c}\| \, \|\mathbf{t}_i\|} \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|} = \frac{\mathbf{c}}{\|\mathbf{c}\|} \sum_{i=0}^{N-1} \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle^2}{\|\mathbf{c}\|^2 \, \|\mathbf{t}_i\|^2},$$

and observing that

$$\sum_{i=0}^{N-1} \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle^2}{\|\mathbf{c}\|^2 \, \|\mathbf{t}_i\|^2} = \sum_{i=0}^{N-1} \langle \frac{\mathbf{c}}{\|\mathbf{c}\|}, \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle}{\|\mathbf{c}\| \, \|\mathbf{t}_i\|} \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|} \rangle = \langle \frac{\mathbf{c}}{\|\mathbf{c}\|}, \sum_{i=0}^{N-1} \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle}{\|\mathbf{c}\| \, \|\mathbf{t}_i\|} \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|} \rangle,$$

the solution may be shown to be

$$\mathbf{c} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{\langle \mathbf{c}, \mathbf{t}_i \rangle}{\|\mathbf{c}\| \, \|\mathbf{t}_i\|} \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|}.$$

Since this solution is not in closed form, an iterative approach

$$\mathbf{c}_{j+1} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{\langle \mathbf{c}_j, \mathbf{t}_i \rangle}{\|\mathbf{c}_j\| \, \|\mathbf{t}_i\|} \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|}$$

is adopted as before. The proof of convergence proceeds in a similar fashion to that for the previous case. As before, it is easily shown that

$$\alpha(\mathbf{c}_j) = \frac{\langle \mathbf{c}_{j+1}, \mathbf{c}_j \rangle}{\sqrt{\langle \mathbf{c}_j, \mathbf{c}_j \rangle}}.$$

Similarly

$$\langle \mathbf{c}_{j+1}, \mathbf{c}_{j+1} \rangle = \frac{\langle \mathbf{c}_{j+1}, \mathbf{c}_{j+1} \rangle}{\sqrt{\langle \mathbf{c}_{j+1}, \mathbf{c}_{j+1} \rangle}} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{\langle \mathbf{c}_j, \mathbf{t}_i \rangle}{\|\mathbf{c}_j\| \, \|\mathbf{t}_i\|} \frac{\langle \mathbf{c}_{j+1}, \mathbf{t}_i \rangle}{\|\mathbf{c}_{j+1}\| \, \|\mathbf{t}_i\|},$$

$$\alpha(\mathbf{c}_j) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{\langle \mathbf{c}_j, \mathbf{t}_i \rangle^2}{\|\mathbf{c}_j\|^2 \, \|\mathbf{t}_i\|^2},$$

and

$$\alpha(\mathbf{c}_{j+1}) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{\langle \mathbf{c}_{j+1}, \mathbf{t}_i \rangle^2}{\|\mathbf{c}_{j+1}\|^2 \, \|\mathbf{t}_i\|^2}.$$

Now writing

$$a_i = \frac{\langle \mathbf{c}_j, \mathbf{t}_i \rangle}{\|\mathbf{c}_j\| \, \|\mathbf{t}_i\|} \quad b_i = \frac{\langle \mathbf{c}_{j+1}, \mathbf{t}_i \rangle}{\|\mathbf{c}_{j+1}\| \, \|\mathbf{t}_i\|}$$

and considering the a_i and b_i each as the components of an N-vector it is clear that

$$\langle \mathbf{c}_{j+1}, \mathbf{c}_{j+1} \rangle = \langle \mathbf{a}, \mathbf{b} \rangle^2, \quad \alpha(\mathbf{c}_j) = \langle \mathbf{a}, \mathbf{a} \rangle \quad \text{and} \quad \alpha(\mathbf{c}_{j+1}) = \langle \mathbf{b}, \mathbf{b} \rangle.$$

Since by the Cauchy-Schwarz inequality $\langle \mathbf{a}, \mathbf{b} \rangle^2 \leq \langle \mathbf{a}, \mathbf{a} \rangle \langle \mathbf{b}, \mathbf{b} \rangle$,

$$\langle \mathbf{c}_{j+1}, \mathbf{c}_{j+1} \rangle \leq \alpha(\mathbf{c}_j) \alpha(\mathbf{c}_{j+1}).$$

A further application of the Cauchy-Schwarz inequality $\langle \mathbf{c}_{j+1}, \mathbf{c}_j \rangle^2 \leq \langle \mathbf{c}_j, \mathbf{c}_j \rangle \langle \mathbf{c}_{j+1}, \mathbf{c}_{j+1} \rangle$ implies

$$(\alpha(\mathbf{c}_j))^2 = \frac{\langle \mathbf{c}_{j+1}, \mathbf{c}_j \rangle^2}{\langle \mathbf{c}_j, \mathbf{c}_j \rangle} \leq \langle \mathbf{c}_{j+1}, \mathbf{c}_{j+1} \rangle \leq \alpha(\mathbf{c}_j) \alpha(\mathbf{c}_{j+1})$$

and therefore

$$\alpha(\mathbf{c}_j) \leq \alpha(\mathbf{c}_{j+1}).$$

Once again, this step may therefore safely be included in the GLA.

6.4.2 Codebook efficiency comparisons

Codebook efficiency comparisons for the "Airplane","Lena" and "Peppers" images are presented in Tables 6.16, 6.17 and 6.18 respectively. Domain and range subtrees were rooted at resolutions i_d and $i_r = i_d + 1$ respectively, and all fixed codebooks were optimised using the GLA with squared deterministic correlation distortion criterion, which consistently delivered a slightly better codebook than the absolute value criterion. In each case the codebook training set was constructed from subtrees rooted at resolution i_r in the set of test images (excluding the image from which the range subtrees were extracted), and the subtrees for the codebook required to initialise the GLA were selected from this training set.

The maximum ϱ^2 values are, in the majority of cases, higher for the fixed codebook than the self codebook, often even for a fixed codebook one quarter the size of the corresponding self codebook. The two codebook performances are equal however, for the highest resolution subtrees in the "Lena" and "Peppers" images[10]. The expected convergence of random and optimised codebook performance with increasing codebook size [210] may be observed, for these cases, in the random codebook results of the previous section[11]. Since local similarity between subtrees is able to confer an advantage over a random codebook on a self codebook, it is possible that, where random and optimised codebook performances are sufficiently similar, this advantage might slightly outweigh that of optimisation.

There is no observable improvement of the self codebook relative to the fixed codebook in changing from the Haar to the spline basis. This suggests that the improved performance of fractal coding in the smooth wavelet basis [46] is not a result of the exploitation of a form of self-affinity which is present to a greater extent than in the Haar basis; a corresponding comparison for an equivalent form of VQ is likely to exhibit a similar performance improvement in the spline basis.

Although the self codebook enjoys the advantage of code vectors which are statistically related to the domains (resulting in the central ϱ^2 peaks), the disadvantage of the absence of an optimisation procedure appears to outweigh it in most of the cases examined here. While borderline strong self-affinity may be present for some large domain pools, there is no evidence for any significant advantage for the self codebook, and one may reasonably claim that, on average, the test images are not strongly self-affine.

The relative merits of self and optimised codebooks as described here are compatible with Lepsøy's comparison [112, ch. 5] of fractal coding with an equivalent VQ scheme, in which VQ

[10] These are also the images with large domain locality effects referred to in Section 6.2.2, but this may well be coincidental, given the small test set.

[11] It is interesting that a random codebook with approximately the distribution of the domain subtrees has similar performance to a codebook optimised for the range subtrees. This observation is evidence for a stochastic fractal nature of natural images, since it appears that their statistics are approximately invariant to a shift in resolution.

Table 6.16: Comparison of self and optimised codebooks for range subtrees in the "Airplane" image.

	Haar basis					Spline basis				
	Self Codebook		Fixed Codebook			Self Codebook		Fixed Codebook		
i_r	ϱ^2	MSE	Size	ϱ^2	MSE	ϱ^2	MSE	Size	ϱ^2	MSE
4	0.38	634.50	64	0.42	564.83	0.35	506.04	64	0.42	444.32
			32	0.41	584.48			32	0.40	458.71
			16	0.39	612.03			16	0.36	473.11
5	0.46	306.66	256	0.50	264.07	0.43	238.01	256	0.49	200.12
			128	0.49	271.20			128	0.48	206.04
			64	0.47	285.60			64	0.47	211.17
6	0.57	99.77	1024	0.58	90.35	0.54	79.65	1024	0.56	67.33
			512	0.57	94.41			512	0.55	69.57
			256	0.56	97.90			256	0.54	71.73

Table 6.17: Comparison of self and optimised codebooks for range subtrees in the "Lena" image.

	Haar basis					Spline basis				
	Self Codebook		Fixed Codebook			Self Codebook		Fixed Codebook		
i_r	ϱ^2	MSE	Size	ϱ^2	MSE	ϱ^2	MSE	Size	ϱ^2	MSE
4	0.45	327.25	64	0.49	305.70	0.42	254.92	64	0.47	230.65
			32	0.47	314.50			32	0.46	236.06
			16	0.44	332.97			16	0.44	243.88
5	0.53	131.56	256	0.54	132.06	0.49	103.38	256	0.51	99.61
			128	0.53	137.84			128	0.50	101.22
			64	0.51	143.47			64	0.49	105.88
6	0.57	46.67	1024	0.57	47.52	0.51	36.23	1024	0.51	36.27
			512	0.56	49.00			512	0.50	37.92
			256	0.55	52.54			256	0.49	39.24

Table 6.18: Comparison of self and optimised codebooks for range subtrees in the "Peppers" image.

	Haar basis					Spline basis				
	Self Codebook		Fixed Codebook			Self Codebook		Fixed Codebook		
i_r	ϱ^2	MSE	Size	ϱ^2	MSE	ϱ^2	MSE	Size	ϱ^2	MSE
4	0.43	570.82	64	0.50	485.70	0.40	425.17	64	0.47	358.06
			32	0.47	508.56			32	0.46	366.25
			16	0.45	539.99			16	0.43	381.26
5	0.54	195.70	256	0.56	179.06	0.47	145.16	256	0.50	132.81
			128	0.55	189.29			128	0.49	135.18
			64	0.53	200.64			64	0.48	140.41
6	0.55	58.15	1024	0.55	59.59	0.46	45.45	1024	0.46	45.73
			512	0.54	61.80			512	0.45	47.40
			256	0.53	65.44			256	0.44	49.05

was found to be superior in a rate distortion sense. A coding scheme constructed as a hybrid of fractal coding and VQ [80] was found to perform significantly better than a standard fractal scheme, and similarly to a comparable VQ scheme, which further supports the superiority of VQ.

Jacquin reported that fractal coding performed slightly better than VQ for edge blocks [95]. If this were the case, one would expect a comparison of self and fixed codebooks with the same deterministic correlation based effectiveness to reveal an advantage in distortion terms for the domain pool, since the magnitudes of edge range subtrees are expected to be larger than average. Although a small effect of this type is observed in the results presented here (for example, $i_r = 5$ and Haar basis in Table 6.17) it is not significant enough to provide convincing evidence. Such an effect is not necessarily incompatible with the conjecture that there is only local dependence between subtrees, since it might be a consequence of particularly high expected similarity between an edge range subtree and its parent domain subtree.

6.5 Conclusions

The consequences of exact self-affinity on image second order statistics were evaluated by constructing an exactly self-affine model in the wavelet domain. Comparisons with measured statistics for a set of test images indicated that the variance decay with increasing resolution associated with an exactly self-affine model may be made compatible with the measured decay by suitable restrictions on the self-affine model parameters, corresponding to restrictions on the parameters of a fractal coding scheme. The off-diagonal correlations of the exactly self-affine models were small, and although the observed pattern was rather unusual, this does not necessarily constitute a significant incompatibility since test image correlations were also

small in the smooth wavelet basis. Despite this apparent compatibility there is evidence for additional non-linear dependence which appears incompatible with image properties.

A direct examination of the self-affinity, similar to that for AR(1) models in Chapter 4, was performed for the set of test images. The domain pool structure of the test images exhibited increased similarity between spatially close range and domain subtrees, decaying rapidly with increasing distance in a similar fashion to that observed for AR(1) models. A comparison with approximations to random domain subtree codebooks suggested that the test images are weakly self-affine to a significant extent, but there was no compelling evidence that this is due to any "special relationship" other than the locally increased similarity between subtrees. While this local similarity may be sufficient for the existence of borderline strong self-affinity in large codebooks, where the optimisation does not constitute a large advantage over a random codebook, the test images are, on average, not strongly self-affine.

It is important to emphasise that as a result of practical difficulties in this evaluation, the results are not as reliable as those for AR(1) models. Since in some cases the differences in codebook performances are rather small, the corresponding classifications in terms of weak and strong self-affinity are similarly unreliable. Considerable further investigation of these issues is therefore required to confirm the tentative conclusions arrived at here.

It appears from the results presented here that there is a significant difference between domains in the neighbourhood of a range, with which they are statistically related, and distant domains with which it is not statistically related, but which act as random code vectors. This distinction may, in simplified form, be encapsulated in an interpretation of fractal coding as a "predict-or-quantise" strategy, in which representation by nearby domains constitutes vector prediction, as a result of the statistical dependence, while representation by remote domains constitutes vector quantisation with a random codebook.

Chapter 7

Conclusions

Despite the apparent absence of any rational foundation in information theory, fractal compression appears to be capable of performance comparable with that of considerably more theoretically sound compression techniques. The aim of the research described here has been to provide a theoretical basis for an understanding of the statistical signal properties required for effective fractal compression.

The fundamental requirement for effective fractal compression is that each signal to be compressed may be efficiently represented by the parameters of a set of affine transforms under which the signal is invariant, and which collectively comprise a contractive mapping in a suitably defined space. The existence of this "self-affinity" property is thus the basic assumption underlying fractal compression, and on which any understanding of fractal compression hinges. This is a complicated assumption, the rationale for which is certainly not obvious[1], although it appears to rest on the existence of a "special relationship" between regions in the same signal, resulting in their being more similar to each other than to regions in separate signals, even if the signals are similarly distributed. In particular, it is not immediately clear whether this entails some hitherto undiscovered property of "natural" images, or is merely a consequence of known image properties. Although other aspects of fractal compression are addressed, three primary questions are posed in examining this self-affinity property:

1. Is self-affinity a natural consequence of common or weak statistical conditions?
2. What are the consequences of self-affinity for signals exhibiting this property?
3. To what extent do "natural" images exhibit this property?

7.1 Restrictions and Limitations of Scope

Fractal compression is a blanket description for a wide variety of coding schemes, as should be clear from Chapter 3. Since many of these differences are not easily represented by different

[1] Barnsley argues [16, ch. 2] that certain affine transforms of subregions of "natural" images are themselves "natural" images, but this does not imply the existence of the required "self-affinity" within a particular image.

parameter choices of a single inclusive model, and in most cases evaluation of these coding schemes is, by practical necessity, performed by numerical experiment rather than symbolic analysis, only a small subset of the possible schemes is examined here. Furthermore, while the more complicated of these schemes are usually the most effective, their complexity represents considerable additional difficulty in obtaining a clear understanding of the underlying principles, and evaluation has consequently been limited to the simplest cases.

The primary restrictions lie in the range partition (fixed size blocks, square where applicable), block transforms (affine), in the limited variety of domain pools (in particular excluding the domain search free types of fractal coding [134] [135]), and in the omission of block isometry operations (see Section 3.2.1) and collage optimisation (see Section 3.7.3). In addition, many of the results were generated for one-dimensional signals only, and in many cases assumptions were made with respect to the transform parameter probability densities.

While any conclusions drawn from the results presented here are obviously subject to these limitations, there is sufficient commonality between fractal coding schemes for one to suspect, in the absence of evidence to the contrary, that they are relevant to a considerably wider variety of schemes than those examined here.

7.2 Description and Interpretation

The term "fractal compression" is derived from the similarity between the signal representation utilised and the Iterated Functions Systems used in generating deterministic fractal images, as well as the potential of the representation for decoding at arbitrary resolution, with synthesis of detail at all resolutions. The prevalence of this term is unfortunate, emphasising as it does one of the least significant aspects of the technique, as well as creating the impression that the success of fractal compression is a result of exploitation of the deterministic fractal properties of natural images, which do not appear to exist to any appreciable extent. In addition, the algorithm of Rinaldo and Calvagno [163] is not associated with fractals, yet is sufficiently conceptually close to fractal compression that they should share a common classification. Alternatives such as "Iterated Transformation Theory" based coding [93], "attractor coding" [112] and "self-quantisation" [47] have been proposed, but have not superseded the original description.

An interesting interpretation of fractal coding as long-range non-causal prediction has been proposed [118], but although there are distinct similarities, the representation of an image block by a distant domain block can not justifiably be viewed as prediction[2], which requires statistical dependence between the blocks, for which there is no evidence in the case

[2]In addition, the prediction is usually based on samples at a fixed location, and is not constructed by locating the best predictor deterministically for each signal. An analogy might be drawn however, with the relationship between transform coding and Singular Value Decomposition (SVD) coding [5] [71]. While transform coding is optimised for optimal energy packing in a statistical sense, the SVD performs a similar function deterministically for each signal to which it is applied.

of images. This is however, a reasonable interpretation of the domain search free types of fractal coding [134] [135].

The usual interpretation in terms of VQ utilising a "self codebook" coincides well with the actual coding procedure of fractal compression, but does not adequately describe the statistical dependence between neighbouring domains and ranges. While any range block may, in principle, be represented to an arbitrary accuracy by a transform on a remote domain block, the "shape" of the range block is far more strongly constrained by requiring it to be equal to a transform on a domain block with which it overlaps, which introduces deterministic constraints on the range block. The domain pool is thus a codebook in the way it is used, but not in the motivation for its use.

It is proposed here that fractal coding be interpreted as a "predict-or-quantise" strategy in the wavelet transform domain. To simplify somewhat, if a range subtree cannot be sufficiently accurately predicted from its parent subtree, it is vector quantised in terms of the random codebook consisting of the remaining domain subtrees. The option of quantisation in the event of inadequate prediction accuracy replaces the usual coding of residual error after prediction. The actual fractal coding process is usually more complex, since there is not such an abrupt transition between prediction and quantisation.

7.3 Self-Affinity of a Standard Class of Signal Models

The self-affinity of a class of standard signal models, consisting of AR(1) models with correlation varying between 0 and 1, was examined in order to determine whether self-affinity is a natural consequence of standard statistical characterisations in terms of second order statistics. Significant self-affinity requires range blocks to be more similar to domain blocks in the same signal than to statistically independent, but similarly distributed, domain blocks. The correlation between samples of an AR(1) model results in such dependence between neighbouring domain and range blocks, but to an extent sufficient only for the presence of marginally significant self-affinity. It is therefore conjectured that strong self-affinity is not a natural consequence of simple statistical restrictions.

7.4 Statistical Consequences of the Self-Affinity Assumption

Since fractal compression performance improves with increasing self-affinity (essentially by definition), it is reasonable to enquire what the consequences of exact self-affinity are for the standard signal statistics, in particular the second order statistics, which for a variety of reasons are used predominantly in stochastic signal modelling. This question is answered by constructing a signal model generating exactly self-affine signals (subject to a number of assumptions, since many such models are possible).

A decaying autocorrelation with initial high correlation, which is compatible with measured image statistics [37, ch. 2], is found to result from the assumption of exact self-affinity.

In the wavelet transform domain, with appropriate restrictions on the model parameters, the decay of average detail coefficient variance with increasing resolution is a reasonable match for the behaviour observed for "natural" images. The average correlations between detail coefficients are small, which is also the case for "natural" images in a number of wavelet bases. The pattern of correlations between detail coefficients is however rather unusual, and it is doubtful whether appropriate adjustments to the model parameters are capable of creating a particularly good match with the corresponding statistics for "natural" images.

Although the Haar transform is found to effectively decorrelate the signal samples of exactly self-affine models, a comparison between the relative efficiencies of transform coding of the transform coefficients, and direct coding of the fractal representation coefficients, reveals the existence of considerable residual dependence subsequent to decorrelation. Thus, although in a broad sense the consequences for second order statistics of exact self-affinity are reasonable (although difficult to adapt as desired by tuning model parameters), there are additional consequences of the assumption which are not revealed by the second order statistics. An appropriate statistical characterisation of these consequences was not found.

7.5 Self-Affinity of Natural Images

The self-affinity of a set of standard images was evaluated in the wavelet transform domain for Haar and spline bases. It should be emphasised that this investigation is considerably less reliable in determining the properties of natural images than the corresponding investigation for AR(1) models, for which a precise characterisation is available. Nevertheless, it appears as if the domain pool structure of natural images is qualitatively similar to that of high correlation AR(1) models, the only significantly enhanced similarity being observed between neighbouring domains and ranges. The domain pool was generally considerably less effective than an optimised codebook, although in a few cases the local similarity was sufficient for the domain pool to be as effective. While natural images may be self-affine to a small but significant extent, there is no evidence that this self-affinity is sufficiently strongly present for the domain pool to have any appreciable advantage over an optimised codebook. The observed degree of self-affinity of natural images appears simply to be a consequence of dependence between neighbouring pixels, rather than some mysterious relationship between all parts of an image.

7.6 Rate Distortion Performance of Fractal Image Coding

Some caution is required in discussing the effectiveness of fractal compression for image coding, since the wide variety of available schemes are responsible for an equally wide variety of performance levels. While a favourable comparison with JPEG transform coding at low bit rates is common [63], this comparison is somewhat biased since this transform coding scheme is not designed for low bit rates [182]. A far more fair comparison may be made with the EZW

algorithm, against which many fractal coding systems compare rather poorly (see Section 3.10). Nevertheless, the existence of a few systems which do offer comparable performance to the state of the art EZW algorithm suggests that the assumptions underlying the fractal coding framework are, at least to some extent, accurate.

The results presented in this dissertation suggest that the effectiveness of fractal coding is the result of the ability of the codebook derived from the domain pool to be almost as effective as a fixed optimised codebook at the same block size. Although the domain pool is similar to a random codebook in structure, the disadvantage of the absence of optimisation is compensated for to some extent by the dependence between neighbouring domain and range blocks, constituting a form of prediction of the range from a domain. Additional constraints necessary for utilising the domain pool in fractal coding, such as those required to guarantee contractivity, further reduce the performance of fractal compression relative to VQ. As a result, it is reasonable to claim that the effectiveness of fractal coding is largely due to its ability to emulate, rather than outperform VQ.

The wavelet transform domain perspective (see Section 3.9) is helpful in explaining why the form of VQ it emulates, operating on detail coefficient subtrees, is effective. Recent high performance compression algorithms [182] have been designed based on assumption of the importance of utilising the combined space-frequency localisation of image energy provided by the wavelet transform, in the sense that the majority of image energy is conjectured to be concentrated at low frequencies (coarse resolution coefficients) as well as spatially localised about edges and texture in the high frequencies (fine resolution coefficients) [208]. The wavelet subtree is the natural structure for representing such combined localisation, and it it thus reasonable to expect VQ on these subtrees to be effective [43]. In addition, since a domain block does not have to be identified when the relevant scaling coefficient is zero, this represents a primitive form of zerotree coding[3], and as the distortion threshold is raised in order to decrease the bit rate, such a zerotree equivalent becomes more likely.

7.7 Advantages and Disadvantages

While the primary criterion of rate distortion behaviour of fractal compression has already been discussed, there are often additional criteria of a more practical nature that are also significant in selecting an appropriate compression algorithm, the most common of which are the encoding and decoding times. Rapid decoding has been claimed [6] as an advantage of fractal compression over transform coding, although there do not appear to have been any comparisons between decoding times for fractal compression and other schemes such as the EZW algorithm. There is no reason however, to expect decoding of the fractal representation to be any more rapid than the equivalent VQ scheme.

[3]More similar to that of Lewis and Knowles [114] than the zerotrees used in coding significance maps in the EZW coder [182].

Encoding times are often not as critical as decoding times[4], and although fractal compression has traditionally required excessive encoding times, significant advances have recently been made in addressing this problem (see Section 3.4). Once again, there is no reason to expect slower encoding for an equivalent VQ system, since any preprocessing required for an efficient search of a fixed codebook need not be performed for each image to be coded.

Another oft-cited advantage of fractal compression is so called "resolution independence" (see Section 3.6.2). Although it is true that the structure of the image representation may be designed to have no explicit reference to a particular image size, this representation is implicitly dependent on the resolution of the image during its construction (i.e. at encoding), and it is more accurate to consider the decoder as being capable of a form of interpolation naturally associated with the image representation. This ability is only partially relevant to actual image compression, since any other compression algorithm could also incorporate interpolation at the decoder if desired[5]. Since empirical evidence suggests that there is no significant advantage to representing image subtrees in terms of subtrees rooted at coarser resolutions in the same image (other than for neighbouring subtrees), it is natural to enquire whether equally effective interpolation is possible for a fixed codebook by truncating the highest resolution coefficients of codebook subtrees for normal operation, and restoring them where interpolation is required. It is, in any event, unlikely that either form of interpolation would be as effective as one based on a more careful analysis of the progression of wavelet maxima across resolutions [33] [34].

The absence of a firm theoretical basis for fractal compression leads to a number of difficulties in practical implementation, as well as in the development of specific coding schemes, where it often results in *ad hoc* design choices, with little understanding of the reasons for their effectiveness. While relaxing the constraints on scaling coefficients has been shown to result in improved reconstruction fidelity [62], guaranteeing eventual contractivity poses a significant problem for all but the simplest cases [90]. A heuristic approach which works in the majority of cases may be acceptable in a research setting, but it is certainly not acceptable in a commercial implementation. An additional disadvantage of fractal compression is that the actual reconstruction distortion is not easily available during the encoding phase, with the result that it is difficult to encode with a target distortion limit. These complications all contribute to increased difficulty in designing a robust compression system which does not require impractical levels of operator tuning[6].

It appears therefore (bearing in mind the restrictions in scope of the results presented here), that fractal coding has no significant advantages over the equivalent (i.e. same block size and product codebook) form of VQ. With the additional consideration of the broad patent

[4]Especially in applications where multiple decode cycles are performed for each encode cycle.

[5]To be fair, the fractal representation may be a more time efficient means of achieving this in circumstances where images are frequently desired to be reproduced at a variety of sizes.

[6]One must assume that these difficulties have been solved in the implementation of the commercially available fractal compression software.

coverage of fractal compression [16], as well as the more developed theoretical background of VQ, this observation weighs heavily in favour of VQ based systems in any evaluation expected to lead to a commercial implementation.

In defence of fractal compression it might be claimed that it is based on rather general statistical assumptions, and is thus suitable for a broad class of signals (as is the EZW algorithm [182], which requires only that each wavelet detail coefficient is likely to have smaller magnitude than its parent). Thus, while VQ may enjoy an advantage when an accurate statistical characterisation of the source is available[7], fractal compression might provide superior performance to VQ when applied to signals differing significantly from those in the VQ training set, since the self codebook of fractal compression provides an adaptation mechanism, which is absent in standard VQ. However, since the self codebook is not equally efficient for all classes of signals (not all statistical sources exhibit significant self-affinity), the existence of such an advantage for fractal compression of "natural" images depends on the similarity between the class of "natural" images and the class of self-affine images. While the comparisons between domain pool and fixed optimised codebooks were performed on a set of very dissimilar images (and the fixed codebook training sets excluded the test images), which suggests that there is no such advantage, the evidence presented here is certainly not sufficient to entirely exclude this possibility.

In conclusion, the evidence presented here suggests that fractal compression is effective due to its ability to emulate comparable forms of VQ, but appears not to offer any substantial advantages over this technique[8], while simultaneously introducing a number of disadvantages. It should however once again be strongly emphasised that these conclusions are subject to the restrictions in scope described at the beginning of this chapter, and are in particular not applicable to partitions such as the HV partition, which is constructed to enhance self-similarity (although empirical evidence presented in Section 3.10 suggests that this does not result in improved performance).

7.8 Further Research

While the research described in this dissertation has contributed to an improved understanding of the implicit statistical assumptions underlying fractal compression, many significant questions remain unresolved. The most significant of these are presented here as issues particularly deserving of further research.

An extension of the evaluations described in this dissertation to a wider range of fractal coding schemes is clearly desirable, and evaluations based on a significantly larger ensemble

[7] A VQ codebook may be designed for any source statistics, whereas fractal compression is based on the assumption of some form of self-affinity, which is not a property shared by all sources.

[8] There may, in some circumstances, be an advantage to fractal compression in not requiring the storage of the codebook at the encoder and decoder, but the actual disk space required for these programs is not particularly critical in most applications.

of test images are required to confirm the tentative conclusions reached here with respect to the self-affinity of natural images. Further examination of the self-affine signal models is also proposed, in order to discover how strongly self-affine a signal model may be without exhibiting clearly unnatural characteristics. It is suggested that comparisons between the rate distortion performance of fractal coders and other techniques would benefit from the inclusion in the comparison of a fixed codebook VQ scheme equivalent to the fractal scheme in as many respects as possible. This would provide an indication whether superior performance by the fractal coder is a result of the exploitation of some strongly present form of self-affinity, or more prosaically, a well designed equivalent VQ algorithm. Since superior performance of the self codebook has been claimed specifically for edge blocks, individual comparisons between self and fixed codebooks are required for blocks of different classifications.

The "predict-or-quantise" interpretation suggests two avenues for generalisation. First, the existing "prediction" stage may be retained, either by considering only the parent domain subtree of a range subtree, or its immediate neighbours as well, but replacing the remainder of the domain pool with a fixed optimised codebook. Second, one might also[9] consider alternative forms of prediction of the range subtree, particularly since there is no apparent reason to expect this to constitute a particularly effective form of prediction, and improved understanding of this aspect is required.

[9]Since evidence suggests there is no advantage whatsoever in the use of spatially remote domains, there is no point in retaining them.

Appendix A

Probability Theory

Probability theory utilises the language of set theory in addressing the analysis of nondeterministic events in a rigorous fashion. Any process which produces results nondeterministically is considered to be an *experiment* with a fixed set of possible outcomes, each of which may be assigned a probability.

A.1 Probability

The fixed set S of all possible outcomes of an experiment is termed the *sample space* [110]. The sample space may be continuous or discrete, depending on the problem at hand. A discrete sample space $S = \{1, 2, 3, 4, 5, 6\}$ for example, would be appropriate to describe the outcome of a single throw of a die, whereas a continuous sample space $S = [50, 300]$ would be appropriate to describe the result of measuring the height in centimetres of a randomly selected individual.

An *event* is a subset of a sample space and is considered to have occurred for an experiment if any of its elements is an outcome of that experiment [110]. Possible events defined on the samples spaces in the preceding examples might be the event "an odd number is thrown" $A = \{1, 3, 5\}$, or the event "a very short person is measured" $B = [50, 100]$.

The purpose of defining a sample space and events of interest is to enable calculation of the probability of these events. The simplest and most easily comprehended interpretation[1] of the probability of an event is that of *relative frequency*. In this interpretation, if an experiment is repeated n times and the event A occurs k times, the probability of event A is approximately k/n, the estimate becoming more accurate as n increases.

The following axioms for a probability measure in a sample space are defined in accordance with the requirements of the intuitive notion of probability discussed above [52] [110]:

Definition 1 *A probability measure on a sample space S is a function*[2] $P : \mathcal{P}(S) \rightarrow \mathbb{R}$ *such*

[1] Although one with some technical difficulties.

[2] The *power set* of A, $\mathcal{P}(A) = \{X | X \subset A\}$ is the set of all subsets of A.

that

1. $P(S) = 1$
2. $P(A) \geq 0 \quad \forall A \subset S$
3. $P(\cup_i A_i) = \sum_i P(A_i) \quad if \quad A_i \cap A_j = \varnothing \quad \forall i \neq j$

These axioms define conditions necessary for a function to be a valid probability measure, but do not suggest how to construct this measure in order to represent an accurate description of the physical situation or experiment of interest. A number of useful results may be derived from these axioms [110, pg. 27]:

1. $P(\varnothing) = 0$
2. $P(S - A) = 1 - P(A)$
3. $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

In many cases knowledge of one event may allow a re-evaluation of the probability of another. The *conditional probability* $P(B|A)$ of event B given that A has occurred is obtained by re-normalising such that A becomes the new sample space

$$P(B|A) = \frac{P(B \cap A)}{P(A)} \quad \text{if} \quad P(A) \neq 0.$$

Returning to the example of the die, if event $A = \{1, 3, 5\}$ and event $B = \{6\}$, it is clear that knowing A has occurred influences the probability of B (in this case fixing it at 0, since A and B are mutually exclusive).

Two events A and B are *independent* if $P(B|A) = P(B)$, which is equivalent to

$$P(A \cap B) = P(A)P(B).$$

Independent events provide no information about each other in the sense of conditional probability.

A.2 Univariate Random Variables

The appropriate model for the result of an experiment often involves a numerical description of the outcome. Random variables provide a formal mechanism for modelling such experiments when the behaviour of the underlying processes is nondeterministic. Just as in the case of deterministic variables, random variables may be manipulated algebraically, although with considerably greater difficulty [187]. Formally, a random variable is defined as follows [110, pg. 76]:

Definition 2 *A (univariate) random variable X on a sample space S is a function $X : S \to \mathbb{R}$.*

Returning to the previous example of random height measurements, a random variable X might be defined as $X(s) = 10s$ with $s \in S$, representing the height in millimetres, while the sample space consists of heights measured in centimetres. The standard methods of dealing with random variables are dependent on the type of range of the random variable.

A.2.1 Discrete random variables

A random variable X is discrete if its range is a discrete set, that is, it consist of only countably many elements in $\mathbb{R}$ [110, pg. 76]. The *probability mass function* (pmf) of a random variable X defined on sample space S

$$p_X(x) = P(\{s \in S \mid X(s) = x\})$$

assigns a probability to each element of the range of X in terms of the probabilities of the underlying events in S. Since the notation above is rather cumbersome, it is often abbreviated as

$$\begin{aligned} P(X = x) &\equiv P(\{s \in S \mid X(s) = x\}) \\ P(X < x) &\equiv P(\{s \in S \mid X(s) < x\}) \end{aligned}$$

and so forth. The probability of an event $\{s \in S \mid a < X(s) \leq b\}$ in the range of the random variable X is

$$P(a < X \leq b) = \sum_{a<x\leq b} p_X(x).$$

The *cumulative distribution function* (cdf) [110, pg. 86] of a random variable X is defined as

$$F_X(t) = P(X \leq t),$$

which may, for a discrete random variable X, be expressed in terms of the pmf as

$$F_X(t) = \sum_{x\leq t} p_X(x).$$

The probability of events in the form introduced above may instead be expressed as [110, pg. 84]

$$P(a < X \leq b) = F_X(b) - F_X(a).$$

A.2.2 Continuous random variables

A *continuous* random variable X has a range consisting of an interval or union of intervals in $\mathbb{R}$, with the probability of any single element of the range equal to zero [110, pg. 76]. The *probability density function* (pdf) f_X is defined such that [110, pp. 87-88]

$$P(a < X \leq b) = \int_a^b f_X(x)\,dx,$$

equivalent to requiring that

$$F_X(t) = \int_{-\infty}^{t} f_X(x)\,dx.$$

The probability of an event may be visualised as the area under the curve integrated over all regions of the x axis within the event of interest (see Figure A.1). Subject to a few technical requirements which will not be dealt with here

$$f_X(t) = \frac{d}{dt}F_X(t).$$

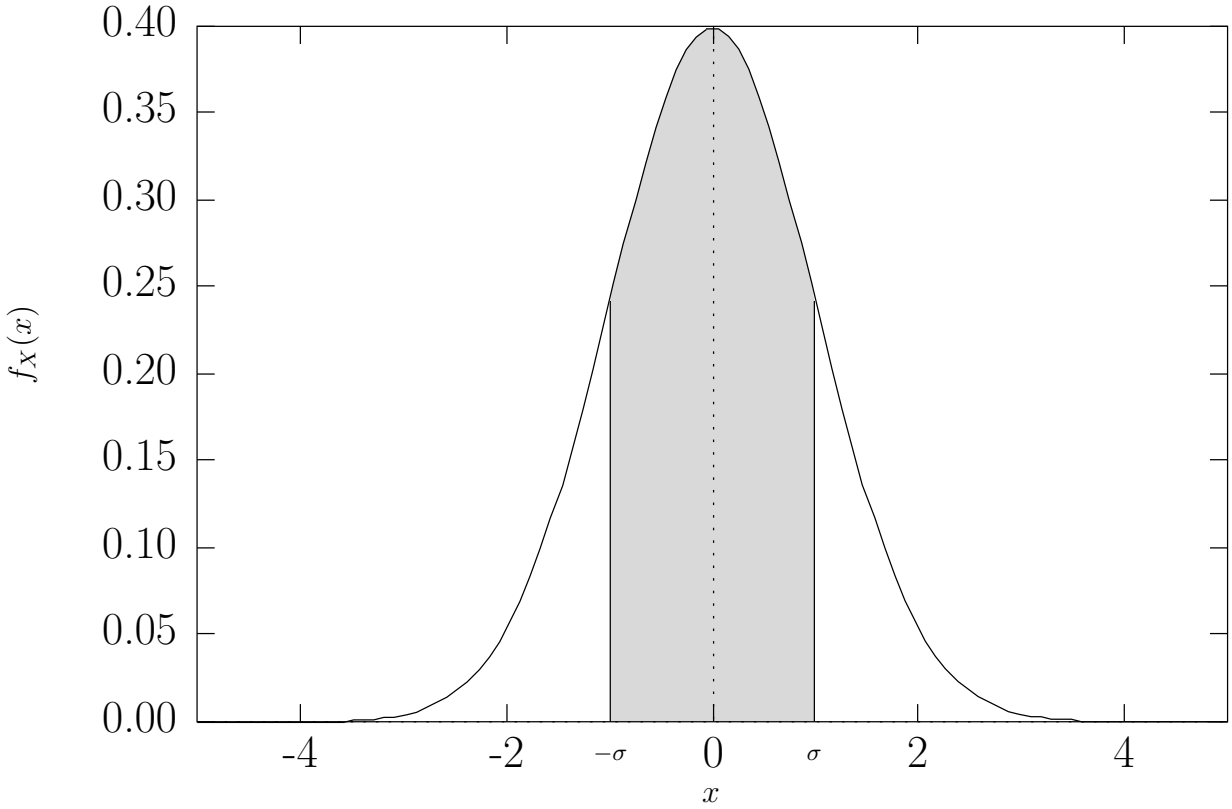


Figure A.1: Probability $P(-\sigma < X < \sigma) \approx 0.68$ as area under the curve for a Gaussian pdf with $\mu = 0$ and $\sigma = 1$.

A.2.3 Expected values

The *expected value* of a function $g(X)$ of random variable X is the average value of the function over all x in the range of X (weighted by the probability of x), and is defined[3] as [110, pp. 95-96]

$$E[g(X)] = \sum_x g(x)p_X(x)$$

for a discrete random variable and

$$E[g(X)] = \int_{-\infty}^{\infty} g(x)f_X(x)\,dx$$

for a continuous random variable. It may be shown that for $a, b \in \mathbb{R}$ and functions g, h that

$$E[ag(X) + bh(X)] = aE[g(X)] + bE[h(X)].$$

The expected values of standard functions $g_k(x) = x^k$, the *moments* of a random variable, are useful in summarising many of its properties. The kth moment of random variable X is $E[X^k]$. The *mean*

$$\mu_X = E[X],$$

the first moment of X, measures the "centroid" of the distribution. The kth *central moment* is defined as $E[(X - \mu_X)^k]$. The variance

$$\sigma_X^2 = E[(X - \mu_X)^2]$$

is the second central moment of X, and is a measure of the "spread" in the distribution. The square root of the variance is the *standard deviation* σ_X.

A.2.4 Standard distributions

Processes are often modelled by random variables selected from a set of standard distributions. The simplest distribution is the *uniform* pdf (see Figure A.2), defined as [110, pp. 152-153]

$$f_X(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & \text{otherwise,} \end{cases}$$

where a and b are parameters determining the extent of the distribution. The mean and variance are $\mu_X = \frac{b+a}{2}$ and $\sigma_X^2 = \frac{(b-a)^2}{12}$ respectively.

The most commonly encountered distribution is the Normal or Gaussian pdf (see Figure

[3]The expected values may not exist if the sum or integral is not convergent.

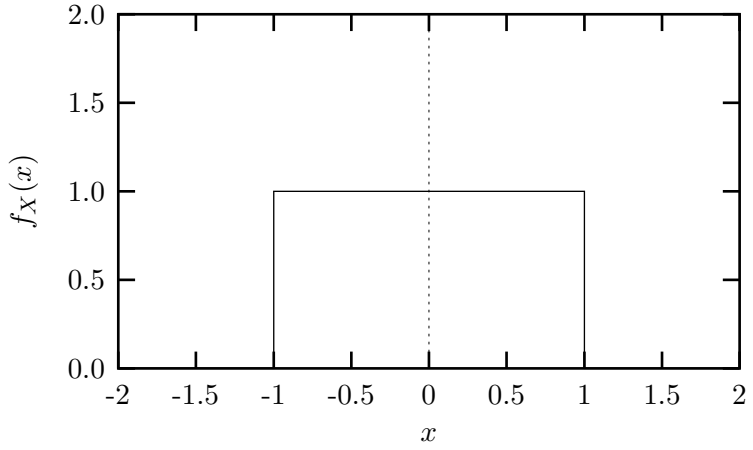


Figure A.2: Uniform pdf on the interval $[-1, 1]$.

A.3), defined as [110]

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}.$$

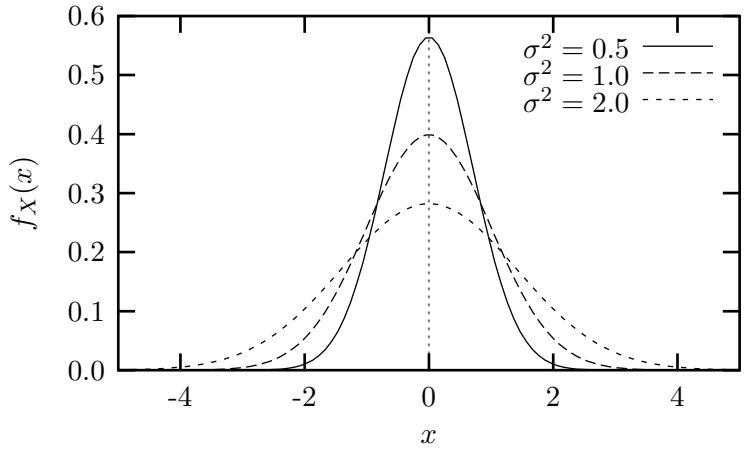


Figure A.3: Gaussian pdfs with $\mu = 0$.

The Laplace pdf (see Figure A.4), defined as [57]

$$f_X(x) = \frac{1}{\sigma\sqrt{2}} e^{-\frac{\sqrt{2}}{\sigma}|x-\mu|}$$

is useful in signal modelling applications, as is the *generalised* Gaussian pdf (see Figure A.5), which is defined as[4] [37, ch. 4] [190]

$$f_X(x) = \frac{\nu\alpha(\sigma,\nu)}{2\Gamma(1/\nu)} e^{-(\alpha(\sigma,\nu)|x-\mu|)^\nu}$$

[4]The gamma function is defined as $\Gamma(x) = \int_0^\infty e^{-t} t^{x-1}\, dt$.

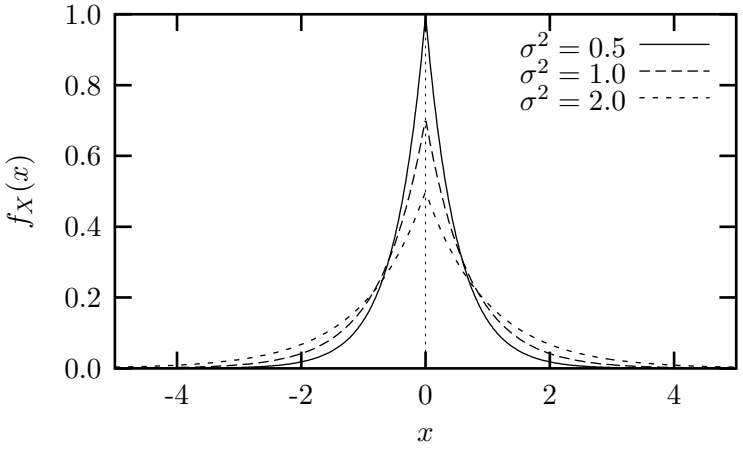


Figure A.4: Laplacian pdfs with $\mu = 0$.

where $\nu \in (0, \infty)$ and

$$\alpha(\sigma, \nu) = \frac{1}{\sigma}\sqrt{\frac{\Gamma(3/\nu)}{\Gamma(1/\nu)}}.$$

The parameter ν controls the form of the pdf, which is the same as the Gaussian pdf for $\nu = 2$, the Laplace pdf for $\nu = 1$, and approaches the uniform pdf as $\nu \to \infty$ [190].

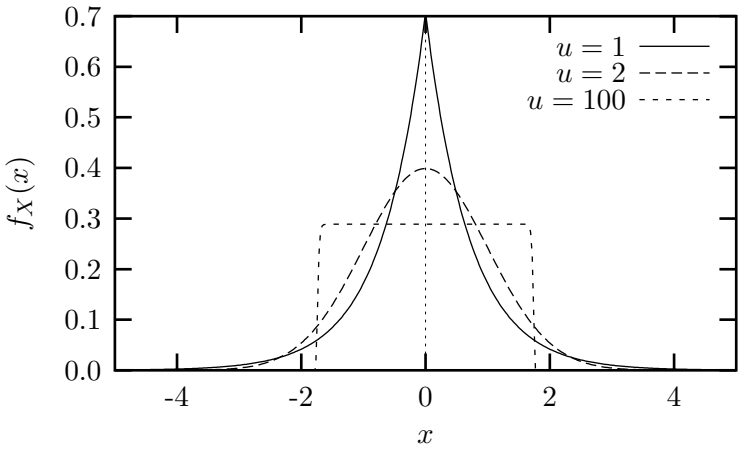


Figure A.5: Generalised Gaussian pdfs with $\mu = 0$ and $\sigma = 1$.

A.3 Multivariate Random Variables

A univariate random variable is inadequate for modelling processes generating vector valued results, such as a simultaneous throw of three dice, or measurement of the height and weight of a randomly selected individual. A multivariate random variable is used where a vector valued range is desired.

Definition 3 *An n-dimensional multivariate random variable* $\mathbf{X}$ *on a sample space* S *is a function* $\mathbf{X} : S \to \mathbb{R}^n$.

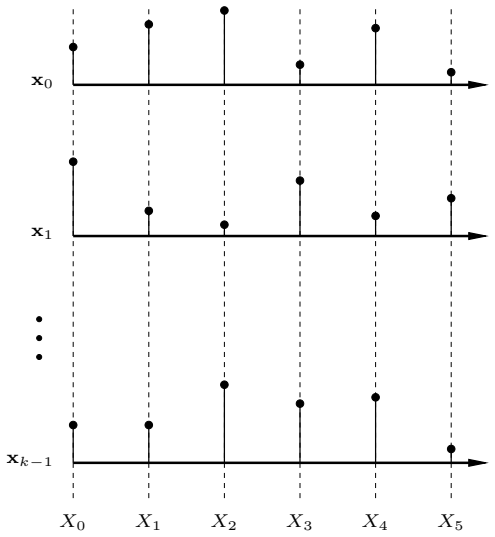


Figure A.6: Ensemble of k vectors $\mathbf{x}_0 \dots \mathbf{x}_{k-1}$. The statistics of element i across the ensemble are represented by the random variable X_i.

The random variable $\mathbf{X} = (X_0, X_1, \dots, X_{n-1})^T$ may be visualised as representing the statistical properties of a large ensemble of individual vectors $\mathbf{x}$ as displayed in Figure A.6.

A.3.1 Discrete random variables

The pmf of a discrete multivariate random variable $\mathbf{X}$ on sample space S is defined analogously to the univariate case as

$$p_{\mathbf{X}}(\mathbf{x}) = P(\{s \in S \mid \mathbf{X}(s) = \mathbf{x}\}),$$

or $p_{\mathbf{X}}(\mathbf{x}) = P(\mathbf{X} = \mathbf{x})$ in the abbreviated notation. As in the univariate case, the probability of an event is obtained by summing the pmf over all elements of the range of $\mathbf{X}$ in that event

$$P(A) = \sum_{\mathbf{x} \in A} p_{\mathbf{X}}(\mathbf{x}).$$

The cdf of a multivariate random variable is defined as

$$F_{\mathbf{X}}(\mathbf{t}) = F_{X_0, X_1, \dots, X_{n-1}}(t_0, t_1, \dots, t_{n-1}) = P(X_0 \leq t_0, X_1 \leq t_1, \dots, X_{n-1} \leq t_{n-1}).$$

A.3.2 Continuous random variables

The pdf $f_{\mathbf{X}}(\mathbf{t})$ of a multivariate continuous random variable $\mathbf{X}$ is defined such that the probability of an event A is

$$P(A) = \int_A f_{\mathbf{X}}(\mathbf{t})\, d\mathbf{t}.$$

Subject to a few technical requirements which will not be discussed here,

$$f_{\mathbf{X}}(\mathbf{t}) = \frac{\partial^n}{\partial t_0 \partial t_1 \dots \partial t_{n-1}} F_{\mathbf{X}}(\mathbf{t}).$$

A.3.3 Marginal distributions

The marginal distributions [52] [110] of a bivariate distribution (X, Y) are the individual distributions of random variables X and Y. The marginal distribution of X is obtained by summing

$$p_X(x) = P(X = x) = \sum_y p_{X,Y}(x, y)$$

or integrating

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x, y)\, dy$$

over all possible values of Y (and *vice versa* for the marginal distribution of Y). When dealing with multivariate distributions of higher dimensionality one may also define the marginal distributions of any subset of the components by integrating or summing over the remainder of the components [52, pp. 143-144].

The individual random variables X_i making up the vector random variable $\mathbf{X}$ are *independent* [81] iff

$$p_{\mathbf{X}}(\mathbf{x}) = \prod_i p_{X_i}(x_i)$$

for a discrete pdf, or

$$f_{\mathbf{X}}(\mathbf{x}) = \prod_i f_{X_i}(x_i)$$

for a continuous distribution.

A.3.4 Conditional distributions

The conditional distributions [52] [110] of a bivariate distribution (X, Y) are the distributions of one of the component random variables given that the value taken on by the other random variable is known. The conditional distribution of X, given that $Y = y$ is defined as

$$p_{X|Y}(x|y) = \frac{p_{X,Y}(x, y)}{p_Y(y)} \quad p_Y(y) > 0$$

for discrete random variables, and as

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)} \quad f_Y(y) > 0$$

for continuous random variables, where the conditional distribution of Y given that $X = x$ is defined analogously. Conditional distributions of a subset of the components of a multivariate distribution of higher dimensionality may also be defined in an analogous fashion [52, pp. 146-148].

A.3.5 Expected values

Expected values are defined by an obvious extension of the univariate definition;

$$E[g(\mathbf{X})] = \sum_{\mathbf{x}} g(\mathbf{x}) p_{\mathbf{X}}(\mathbf{x})$$

for a discrete random variable and

$$E[g(\mathbf{X})] = \int g(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}$$

for a continuous random variable. The function $\mathbf{g}(\mathbf{X})$ may be vector valued, resulting in a vector valued expectation. In particular, the mean vector $\boldsymbol{\mu}_{\boldsymbol{X}} = (\mu_{X_0}, \mu_{X_1}, \ldots, \mu_{X_{n-1}})^T$ is

$$\boldsymbol{\mu}_{\boldsymbol{X}} = E[\mathbf{X}].$$

A.3.6 Correlation and covariance

Given a bivariate distribution (X, Y), the covariance of X and Y [110, ch. 5]

$$\sigma_{XY} = E[(X - \mu_X)(Y - \mu_Y)]$$

is a measure of the linear dependence between X and Y. Since this value is dependent on the individual variances of X and Y, a better indication of the linearity of the relationship is the correlation coefficient [110, ch. 5] (also called the normalised covariance [99, ch. 2])

$$\rho_{XY} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y},$$

which may be shown to be restricted to the range $\rho_{XY} \in [-1, 1]$.

The autocorrelation of $\mathbf{X}$

$$R_{\mathbf{X}} = E[\mathbf{X}\mathbf{X}^T]$$

and autocovariance of $\mathbf{X}$

$$C_{\mathbf{X}} = E[(\mathbf{X} - \boldsymbol{\mu}_{\boldsymbol{X}})(\mathbf{X} - \boldsymbol{\mu}_{\boldsymbol{X}})^T]$$

are a measure of linear dependence between components of the multivariate random variable $\mathbf{X}$. The diagonal element at position (i,i) of $C_{\mathbf{X}}$ is the variance $\sigma^2_{X_i}$ and the off diagonal element at (i,j) is the covariance $\sigma_{X_iX_j}$.

It is often convenient to express the autocorrelation and autocovariance as the functions

$$R_X(i,j) = E[X_iX_j]$$

and

$$C_X(i,j) = E[(X_i - \mu_{X_i})(X_j - \mu_{X_j})]$$

respectively. A *wide-sense stationary* random variable $\mathbf{X}$ has

$$\mu_{X_j} = \mu_{X_{j+k}} = \mu_X \text{ and } \sigma^2_{X_j} = \sigma^2_{X_{j+k}} = \sigma^2_X \quad \forall j,k$$

in which case the autocovariance and autocorrelation are functions of relative displacement $k = n - m$ only, i.e.

$$R_X(k) = E[X_jX_{j+k}]$$

and

$$C_X(k) = E[(X_j - \mu_X)(X_{j+k} - \mu_X)].$$

In this case one may define the power spectral density function

$$S_X(\omega) = \sum_k R_X(k)e^{-\imath\omega k}$$

as the Fourier transform of the autocorrelation function [73, ch. 2], and the variance normalised autocorrelation function as

$$\rho_X(k) = \frac{R_X(k)}{R_X(0)}.$$

Appendix B

Functional Analysis

Functional analysis plays an important role in signal processing, forming the mathematical basis of tools such as spectral analysis and Fourier transform theory, as well as the more recently developed wavelet theory and multiresolution analysis. In addition, concepts from functional analysis are fundamental to fractal coding.

B.1 Vector Spaces

Vectors and their associated vector spaces play a fundamental role in functional analysis. A vector space is a non-empty set upon the elements of which operations of addition and scalar multiplication with desirable properties are defined [108]:

Definition 4 *A vector space over a field*[1] K *is a set* $V \neq \varnothing$ *together with the two algebraic operations:*
Vector addition: $V \times V \rightarrow V$ *satisfying*

1. $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u} \quad \forall \mathbf{u}, \mathbf{v} \in V$
2. $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w} \quad \forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in V$
3. $\exists \mathbf{0} \in V$ *such that* $\mathbf{u} + \mathbf{0} = \mathbf{u} \quad \forall \mathbf{u} \in V$
4. $\forall \mathbf{u} \in V \;\; \exists (-\mathbf{u}) \in V$ *such that* $\mathbf{u} + (-\mathbf{u}) = \mathbf{0}$

and
Scalar multiplication: $K \times V \rightarrow V$ *satisfying*

1. $k(\mathbf{u} + \mathbf{v}) = k\mathbf{u} + k\mathbf{v} \quad \forall k \in K, \; \mathbf{u}, \mathbf{v} \in V$
2. $(k + l)\mathbf{u} = k\mathbf{u} + l\mathbf{u} \quad \forall k, l \in K, \; \mathbf{u} \in V$

[1] See Rudin [166] for example, for the definition of a field. Little is lost in this context in assuming that K is either $\mathbb{R}$ or $\mathbb{C}$.

3. $(kl)\mathbf{u} = k(l\mathbf{u}) \quad \forall k, l \in K,\ \mathbf{u} \in V$

4. $\exists 1 \in K$ *such that* $1\mathbf{u} = \mathbf{u} \quad \forall \mathbf{u} \in V$

The simplest example of a vector space is $\mathbb{R}^n$, the space of n-dimensional Euclidean vectors, where a geometric interpretation of the individual vectors as directed line segments is possible.

A subspace of a vector space is a subset of that vector space which is closed under addition and scalar multiplication, i.e. all vector additions and scalar multiplications involving elements of the subset are also elements of the subset. These requirements may be encapsulated in the following definition [108]:

Definition 5 *A subspace U of vector space V is a set $U \subset V$ such that*

1. $U \neq \varnothing$

2. $\mathbf{u}, \mathbf{v} \in U \Rightarrow k\mathbf{u} + l\mathbf{v} \in U \quad \forall k, l \in K$

Any plane containing the origin is a subspace of $\mathbb{R}^3$, for example.

The orthogonal complement of a subspace of a vector space is the set of all vectors in the vector space which are orthogonal to every vector in the subspace [108]:

Definition 6 *The orthogonal complement $U^\perp$ of U in V, where U is a subspace of V, is*

$$U^\perp = \{\mathbf{v} \in V \mid \langle \mathbf{u}, \mathbf{v} \rangle = 0 \ \ \forall \mathbf{u} \in U\}.$$

The orthogonal complement in $\mathbb{R}^2$ of the subspace represented by a line through the origin is the line through the origin at right angles to the first.

A vector space is the sum of two subspaces if any vector in the space may be expressed as the sum of a vector from each of the subspaces. If, in addition, the intersection of the subspaces contains only the $\mathbf{0}$ vector, uniqueness of the expansion in terms of vectors of the subspaces is guaranteed and the sum is a direct sum [108]:

Definition 7 *A vector space U is the direct sum $U = V \oplus W$ of two subspaces V and W if every $u \in U$ may be expressed* uniquely *as $\mathbf{u} = \mathbf{v} + \mathbf{w}$ where $\mathbf{v} \in V$ and $\mathbf{w} \in W$.*

The vector space $\mathbb{R}^2$, for example, may be expressed as the direct sum of the subspaces represented by non-parallel lines through the origin. A projection is an operator taking a vector space into one of its subspaces:

Definition 8 *The operator $P : U \to U$ is the projection from U onto V and parallel to W if $U = V \oplus W$, and $P\mathbf{u} = \mathbf{v}$ where $\mathbf{u} = \mathbf{v} + \mathbf{w}$ for $\mathbf{u} \in U$, $\mathbf{v} \in V$ and $\mathbf{w} \in W$.*

A projection operator P is necessarily linear and idempotent (i.e. $P^2 = P$).

If the subspaces involved in a direct sum decomposition are orthogonal, the sum is termed an orthogonal sum.

Definition 9 *A vector space U is the orthogonal sum*[2] *of two subspaces V and W if $U = V \oplus W$ and $W = V^{\perp}$.*

In the example involving $\mathbb{R}^2$ above, the orthogonal sum decomposition corresponds to two orthogonal lines. The orthogonal projection from U onto a subspace V is the projection onto V and parallel to $V^{\perp}$.

Subsequent notation and definitions will be simplified by assuming that all vector spaces are over the field $K = \mathbb{R}$.

B.2 Metric Spaces

A vector space imposes structure on a set by defining operators which constitute rules for generating set members from other set members. In contrast to this, a metric space need have no such operations defined, imposing a different type of structure by defining the notion of a *distance* or *metric* between pairs of set members. In order to qualify as a distance measure, a function should conform to the following definition [108]:

Definition 10 *A* metric *on a set X is a function $d : X \times X \to \mathbb{R}$ such that*

1. $d(x, y) = 0 \Leftrightarrow x = y \quad \forall x, y \in X$
2. $d(x, y) = d(y, x) \quad \forall x, y \in X$
3. $d(x, y) \leq d(x, z) + d(z, y) \quad \forall x, y, z \in X$

The definition of a metric *space* follows immediately:

Definition 11 *A* metric space (X, d) *is a non-empty set X together with a metric d.*

Examples of metric spaces include $(\mathbb{R}, d_E)$, the real numbers with the Euclidean metric $d_E(x, y) = \mid x - y \mid \ \forall x, y \in \mathbb{R}$, and (X, d_D) for arbitrary non-empty set X with the discrete metric

$$\forall x, y \in X \quad d_D(x, y) = \begin{cases} 0 & \text{if} \quad x = y \\ 1 & \text{if} \quad x \neq y. \end{cases}$$

While the structure imposed on a finite set by a distance measure may not be particularly interesting, concepts such as continuity become useful for infinite sets. The behaviour of sequences within these infinite sets is a useful way of characterising the structure of the set under the chosen metric.

Definition 12 *A sequence in (X, d) is a function*[3] *$S : \mathbb{N} \to X$. The members of the sequence are denoted $x_0, x_1, x_2, \ldots$, i.e. $x_n = S(n)$.*

---

[2] The orthogonal sum may be denoted by $\overset{\perp}{\oplus}$. Note, however, that Chui [36] utilises $\dot{+}$ for the direct sum, and $\oplus$ for the orthogonal sum.

[3] The natural numbers are defined here as $\mathbb{N} = \{n \in \mathbb{Z} \,|\, n \geq 0\}$. Note that a common alternative definition excludes 0 from this set.

It is also possible to define a bi-infinite sequence, which is a function $S : \mathbb{Z} \to X$ having members $\ldots, x_{-1}, x_0, x_1, \ldots$. When referring to a sequence as a whole, it is conventional[4] to write $\{x_n\}$ as opposed to x_n, which is just the nth element of the sequence. The behaviour of a sequence as it approaches infinity is of particular interest, leading to the notion of convergence.

Definition 13 *A sequence* $\{x_n\} \subset (X, d)$ *is a* Cauchy sequence *if* $\forall \epsilon > 0\ \exists N \in \mathbb{N}$ *such that* $n, m > N \Rightarrow d(x_n, x_m) < \epsilon$.

A Cauchy sequence has the property that $d(x_n, x_m) \to 0$ as $m, n \to \infty$. This does not necessarily imply that the sequence is convergent in the following sense,

Definition 14 *A sequence* $\{x_n\} \subset (X, d)$ converges *to a point* $x \in X$ *if* $\forall \epsilon > 0\ \exists N > 0$ *such that* $n > N \Rightarrow d(x_n, x) < \epsilon$,

although all convergent sequences are Cauchy sequences. A Cauchy sequence in (X, d) is not a convergent sequence if it "attempts to converge" to a limit which is not in X, e.g. the sequence

$$x_n = 1 + 1/1! + 1/2! + 1/3! + \ldots + 1/n!$$

in $\mathbb{Q}$ converges to a limit in $\mathbb{R}$, but not in $\mathbb{Q}$ [155, pg. 66] [166, pp. 63-65].

Definition 15 *A metric space* (X, d) *is* complete *if every Cauchy sequence in* (X, d) *converges to a limit in* X.

While $\mathbb{R}$ is complete, the counterexample above indicates that $\mathbb{Q}$ is not.

Associated with a set such as $(0, 1) \subset \mathbb{R}$ are boundary points ("accumulation points") which are not in the set. The closure of a set is the original set together with the accumulation points, which are 0 and 1 in the example above.

Definition 16 *The closure [10] [166]* $\overline{X}$ *in* Y *of a subset* X *of metric space* Y *is*

$$\overline{X} = \{x \in Y \mid \exists \{x_i\} \subset X \textit{such that} \lim_{i \to \infty} x_i = x\}$$

A set is closed if it is equal to its closure. A closed space is necessarily complete, but not vice versa.

Definition 17 *A subset* X *of the metric space* Y *is dense [108] in* Y *if* $\overline{X} = Y$.

If X is dense in Y, every member of Y is the limit of a sequence in X.

[4] A common alternative notation is (x_n) for the sequence as a whole.

B.3 Normed and Inner Product Spaces

While metric spaces were defined for an arbitrary set in the previous section, there is no reason why that set should not be a vector space if a suitable distance is defined between any two vectors in that space. Instead of defining a distance on a vector space, one may define the norm [108]:

Definition 18 *A norm on a vector space V over $\mathbb{R}$ is a function $\|\cdot\| : V \to \mathbb{R}$ such that*

1. $\|\mathbf{v}\| \geq 0 \quad \forall \mathbf{v} \in V$
2. $\|a\mathbf{v}\| = |a| \, \|\mathbf{v}\| \quad \forall a \in \mathbb{R}, \mathbf{v} \in V$
3. $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\| \quad \forall \mathbf{u}, \mathbf{v} \in V$
4. $\|\mathbf{v}\| = 0 \Leftrightarrow \mathbf{v} = \mathbf{0} \quad \forall \mathbf{v} \in V$

which corresponds to the notion of vector magnitude for Euclidean vectors. A norm on a vector space immediately induces a metric on that space defined by

$$d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\| \quad \forall \mathbf{u}, \mathbf{v} \in V,$$

although not every metric on a vector space is induced by a norm.

The definition of a normed space follows immediately

Definition 19 *A normed space is a vector space V together with a norm $\|\cdot\|$.*

Since every norm induces a metric, completeness is defined for a normed space, allowing the following definition

Definition 20 *A Banach space is a complete normed space.*

An alternative function which may be defined on a vector space is the inner product [108]:

Definition 21 *An inner product on a vector space V over $\mathbb{R}$ is a function $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$ such that*

1. $\langle \mathbf{u} + \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{u}, \mathbf{w} \rangle + \langle \mathbf{v}, \mathbf{w} \rangle \quad \forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in V$
2. $\langle a\mathbf{u}, \mathbf{v} \rangle = a\langle \mathbf{u}, \mathbf{v} \rangle \quad \forall a \in \mathbb{R}, \mathbf{u}, \mathbf{v} \in V$
3. $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{v}, \mathbf{u} \rangle \quad \forall \mathbf{u}, \mathbf{v} \in V$
4. $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$ *and* $\langle \mathbf{u}, \mathbf{u} \rangle = 0 \Leftrightarrow \mathbf{u} = \mathbf{0} \quad \forall \mathbf{u} \in V$

Definition 22 *An inner product space is a vector space V together with an inner product $\langle \cdot, \cdot \rangle$.*

A commonly encountered example of an inner product space is $\mathbb{R}^n$ with the inner product

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=0}^{n-1} u_i v_i.$$

An inner product space is also a normed space since the inner product induces a norm

$$\|\mathbf{u}\| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}.$$

Once again, not every norm is induced by an inner product. The relationship between the various spaces introduced here is displayed in Venn diagram form in Figure B.1.

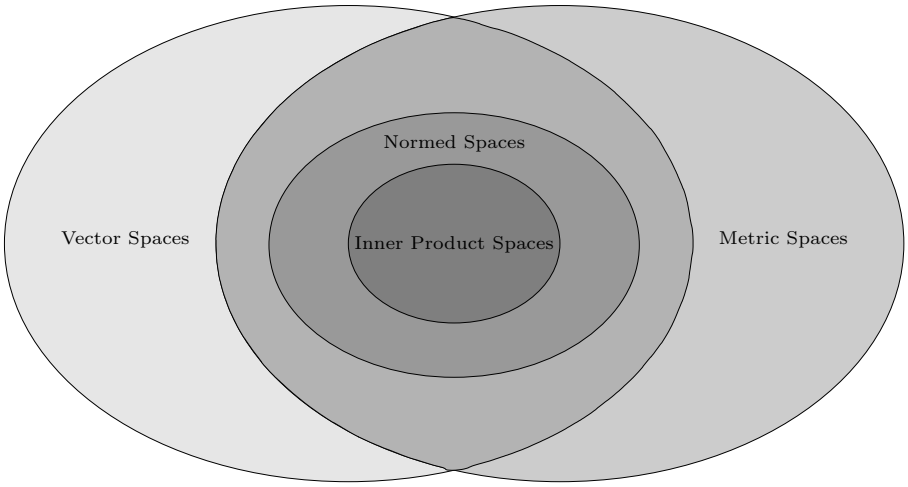


Figure B.1: Relationship between vector, metric, normed and inner product spaces.

Definition 23 *A Hilbert space is a complete inner product space.*

Hilbert spaces of particular interest here are $L^2(\mathbb{R})$ and $l^2(\mathbb{Z})$, the spaces of square-integrable functions, and square-summable sequences respectively.

Definition 24 *The Hilbert space $L^2(\mathbb{R})$ is defined as*[5]

$$L^2(\mathbb{R}) = \left\{ f : \mathbb{R} \to \mathbb{R} \;\middle|\; \int_{-\infty}^{\infty} |f(x)|^2 \, dx < \infty \right\}$$

with the inner product

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f(x) g(x) \, dx.$$

[5]The integral is a Lebesgue integral, but is equivalent to the Riemann integral for "well-behaved" functions.

Definition 25 *The Hilbert space $l^2(\mathbb{Z})$ is defined as*

$$l^2(\mathbb{Z}) = \left\{ \{a_i\} \;\middle|\; \sum_{i=-\infty}^{\infty} a_i^2 < \infty \right\}$$

with the inner product

$$\langle \{a_i\}, \{b_i\} \rangle = \sum_{i=-\infty}^{\infty} a_i b_i.$$

These definitions correspond to the sets of functions and sequences with "finite energy" respectively.

B.4 Orthogonal Bases

In a finite (n-)dimensional Hilbert space H a set $\{\mathbf{e}_0, \mathbf{e}_1, \ldots, \mathbf{e}_{n-1}\}$ forms an orthonormal basis of H if $\langle \mathbf{e}_i, \mathbf{e}_j \rangle = \delta_{i,j}$ and $\text{span}\{\mathbf{e}_0, \mathbf{e}_1, \ldots, \mathbf{e}_{n-1}\} = H$. Any vector $\mathbf{u} \in H$ may be expressed as [197] $\mathbf{u} = \sum_i c_i \mathbf{e}_i$ where the *Fourier coefficients* of $\mathbf{u}$ are $c_i = \langle \mathbf{u}, \mathbf{e}_i \rangle$.

The situation is more complicated in an infinite dimensional vector space, since by definition, no finite set can form a basis. In general an orthonormal set $\{\mathbf{e}_i\}_{i \in \mathbb{N}} \subset H$ forms an orthonormal basis of H iff any of the following equivalent conditions hold [141, ch. 5]

1.	$\langle \mathbf{u}, \mathbf{e}_i \rangle = 0 \quad \forall i \in \mathbb{N} \Rightarrow \mathbf{u} = \mathbf{0}$	$\{\mathbf{e}_i\}$ is a maximal orthonormal set
2.	$\mathbf{u} = \sum_{i \in \mathbb{N}} \langle \mathbf{u}, \mathbf{e}_i \rangle \mathbf{e}_i \quad \forall \mathbf{u} \in H$	Fourier series expansion
3.	$\|\mathbf{u}\|^2 = \sum_{i \in \mathbb{N}} \langle \mathbf{u}, \mathbf{e}_i \rangle^2 \quad \forall \mathbf{u} \in H$	Parseval equality
4.	$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i \in \mathbb{N}} \langle \mathbf{u}, \mathbf{e}_i \rangle \langle \mathbf{v}, \mathbf{e}_i \rangle \quad \forall \mathbf{u}, \mathbf{v} \in H$	Generalised Parseval equality
5.	$\overline{\text{span}\{\mathbf{e}_i\}_{i \in \mathbb{N}}} = H$	$\text{span}\{\mathbf{e}_i\}_{i \in \mathbb{N}}$ is dense in H

B.5 Biorthogonal Bases

More general representations than those provided by orthogonal bases find application where the the conditions necessary for a set to form an orthogonal basis are excessively restrictive. A frame [39, pp. 130-131] [45, ch. 3] [197] [209, ch. 4] is a set of vectors providing a stable representation in a Hilbert space, but which are not necessarily linearly independent, and therefore do not necessarily constitute a basis. Stability is guaranteed by the following definition:

Definition 26 *A frame in a Hilbert space H is a set of vectors $\{\mathbf{e}_i\}_{i\in\mathbb{N}} \subset H$ where $\exists A, B \in \mathbb{R}$, $A, B > 0$ such that $\forall \mathbf{u} \in H$*

$$A\|\mathbf{u}\|^2 \leq \sum_i \langle \mathbf{u}, \mathbf{e}_i\rangle^2 \leq B\|\mathbf{u}\|^2.$$

A more general basis than an orthogonal basis may be defined by the addition of a linear independence requirement to the definition of a frame [39, pp. 130-131] [209, ch. 1]:

Definition 27 *A Riesz basis of a Hilbert space H is a frame such that the constituent vectors $\{\mathbf{e}_i\}_{i\in\mathbb{N}} \subset H$ are linearly independent.*

Since a Riesz basis $\{\mathbf{e}_i\}_{i\in\mathbb{N}}$ is generally not orthogonal, the appropriate Fourier coefficients in such a basis may not be calculated as inner products with the basis vectors themselves. It may be shown, however, that the Fourier coefficients in this basis may be computed with respect to a *dual* Riesz basis $\{\tilde{\mathbf{e}}_i\}_{i\in\mathbb{N}}$, selected such that the basis and its dual are biorthogonal

$$\langle \mathbf{e}_i, \tilde{\mathbf{e}}_j\rangle = \delta_{i,j} \quad \forall i, j \in \mathbb{N}.$$

Definition 28 *A set of vectors $\{\mathbf{e}_i\}_{i\in\mathbb{N}}$ and its dual $\{\tilde{\mathbf{e}}_i\}_{i\in\mathbb{N}}$ represent biorthogonal bases [39] [197] of a Hilbert space H iff each set is a frame, and $\langle \mathbf{e}_i, \tilde{\mathbf{e}}_j\rangle = \delta_{i,j}$ $\forall i, j \in \mathbb{N}$.*

The primary basis and its biorthogonal dual basis are automatically Riesz bases, since the biorthogonality condition implies linear independence of both sets [39, pg. 131]. Any $\mathbf{u} \in H$ may be expressed in such biorthogonal bases as

$$\mathbf{u} = \sum_{i\in\mathbb{N}} \langle \mathbf{u}, \mathbf{e}_i\rangle \tilde{\mathbf{e}}_i = \sum_{i\in\mathbb{N}} \langle \mathbf{u}, \tilde{\mathbf{e}}_i\rangle \mathbf{e}_i.$$

Note that an orthogonal basis corresponds to the special case of $\{\mathbf{e}_i\} = \{\tilde{\mathbf{e}}_i\}$.

The Parseval equality and the generalised Parseval equality for biorthogonal bases are [197, pg. 26]

$$\|\mathbf{u}\|^2 = \sum_{i\in\mathbb{N}} \langle \mathbf{u}, \mathbf{e}_i\rangle\langle \mathbf{u}, \tilde{\mathbf{e}}_i\rangle$$

and

$$\langle \mathbf{u}, \mathbf{v}\rangle = \sum_{i\in\mathbb{N}} \langle \mathbf{u}, \mathbf{e}_i\rangle\langle \mathbf{v}, \tilde{\mathbf{e}}_i\rangle = \sum_{i\in\mathbb{N}} \langle \mathbf{u}, \tilde{\mathbf{e}}_i\rangle\langle \mathbf{v}, \mathbf{e}_i\rangle$$

respectively.

It is, in some circumstances, convenient to index the set of basis vectors $\{\mathbf{e}_i\}_{i\in\mathbb{Z}}$ by the integers, requiring trivial modifications to the conditions above.

Appendix C

Multiresolution Analysis and Wavelet Bases

The basis functions of transforms such as the DFT and DCT are constructed by dilation of a single periodic function (the complex exponential and the cosine function for the DFT and DCT respectively). The resulting decompositions discard all spatial localisation, since the periodic basis functions are not spatially concentrated within a particular region. Some spatial information may be retained by modulating the basis functions by translates of a suitable "window" function (blocked transforms may be considered to result from modulation by appropriate translates of the box function), but the fixed width of the window function represents a compromise between spatial resolution (requiring a narrow window) and accuracy of representation on the selected basis (requiring a wide window).

The wavelet transform achieves variable spatial resolution by decomposing functions onto a basis constructed from translations and dilations of a single non-periodic function (the mother wavelet). Dyadic wavelet bases, in which the basis functions are constructed from dilations of the mother wavelet by powers of two, are particularly useful in signal processing and compression applications.

C.1 Multiresolution Analysis

Although there are alternative avenues [45, pg. 136], the simplest construction of dyadic wavelets is via Multiresolution Analysis (MRA), which provides a formal framework for analysing functions at different resolutions. Low resolution approximations to a function may be considered to be lowpass filtered versions thereof, while progressively higher resolution approximations correspond progressively more closely to the original function as more function detail is included in the approximation. The formal definition of an MRA ensures that these approximations are structured so that wavelets generate the bases appropriate for representing the detail required to move from one resolution to the next.

Definition 29 *A* multiresolution analysis *of* $L^2(\mathbb{R})$ *is a sequence* $\{V_i\}_{i\in\mathbb{Z}}$ *of closed subspaces of* $L^2(\mathbb{R})$ *such that [98] [197, ch. 4]*

1.	$V_i \subset V_{i+1}$	*Containment*
2.	$\overline{\bigcup_{i\in\mathbb{Z}} V_i} = L^2(\mathbb{R})$	*Upward completeness*
3.	$\bigcap_{i\in\mathbb{Z}} V_i = \{0\}$	*Downward completeness*
4.	$f(x) \in V_i \Leftrightarrow f(2x) \in V_{i+1}$	*Scale invariance*
5.	$f(x) \in V_0 \Leftrightarrow f(x+1) \in V_0$	*Translation invariance*
6.	$\exists\phi \in V_0$ *such that* $\int_{-\infty}^{\infty} \phi(x)\,dx \neq 0$ *and* $\{\phi(x-j)\}_{j\in\mathbb{Z}}$ *is a Riesz basis of* V_0.	

Each subspace V_i contains the approximation at resolution i of all functions $f \in L^2(\mathbb{R})$, where V_{i+1} contains functions at a higher[1] resolution than V_i. Each detail space W_i is defined as the complement of V_i in V_{i+1}

$$V_{i+1} = V_i \oplus W_i,$$

representing the "detail information" removed by the approximation of a function $v_{i+1} \in V_{i+1}$ by a function $v_i \in V_i$. Any approximation space may therefore be expressed as the direct sum of a lower resolution approximation space and the intervening detail spaces

$$V_N = V_M \oplus \bigoplus_{i=M}^{N-1} W_i \quad M < N.$$

Defining the projection operators $P_i : L^2(\mathbb{R}) \to V_i$ and $Q_i : L^2(\mathbb{R}) \to W_i$ into the approximation and detail spaces respectively, this implies that, for any $f \in L^2(\mathbb{R})$

$$P_N f = P_M f + \sum_{i=M}^{N-1} Q_i f \quad M < N,$$

resulting in a representation of $P_N f \in V_N$ as the sum of a low resolution approximation and detail signals, each containing localised spectral information in a different frequency subband [36, pg. 120]. Any $f \in L^2(\mathbb{R})$ may, as a result of the upward completeness property, be expressed as $f = P_M f + \sum_{i=M}^{\infty} Q_i f$ or $f = \sum_{i\in\mathbb{Z}} Q_i f$.

It may be shown that since the *scaling function* ϕ generates a Riesz basis of V_0, it also generates a Riesz basis $\{\phi_{i,j}\}_{j\in\mathbb{Z}}$ of V_i, where

$$\phi_{i,j} = \sqrt{2^i}\phi(2^i x - j).$$

[1] Daubechies [45] and a number of other authors index in the opposite direction, i.e. $V_{i+1} \subset V_i$.

The Fourier coefficients of function f in this basis are the *approximation* coefficients $c_{i,j} = \langle f, \phi_{i,j} \rangle$. The *dilation equation*

$$\phi(x) = \sqrt{2} \sum_k h_k \phi(2x - k),$$

is a result of the observation that, since $\phi \in V_0 \subset V_1$, it must be possible to express ϕ as a linear combination of the basis functions $\{\phi_{1,j}\}_{j \in \mathbb{Z}}$ of V_1 . The sequence $\{h_k\}$, together with the normalisation [98]

$$\int_{-\infty}^{\infty} \phi(x)\, dx = 1$$

uniquely characterises the scaling function.

The central result of multiresolution analysis is that, given an MRA as defined above, a *mother wavelet* ψ may be derived from the scaling function ϕ such that $\{\psi(x - j)\}_{j \in \mathbb{Z}}$ is a Riesz basis of W_0 [45, ch. 5]. Since the detail spaces inherit the scaling and translation properties of the approximation spaces, this implies that $\{\psi_{i,j}\}_{j \in \mathbb{Z}}$ is a Riesz basis of W_i, where

$$\psi_{i,j} = \sqrt{2^i} \psi(2^i x - j).$$

The Fourier coefficients of function f in this basis are the *detail* coefficients $d_{i,j} = \langle f, \psi_{i,j} \rangle$. A second dilation equation

$$\psi(x) = \sqrt{2} \sum_k g_k \phi(2x - k)$$

is obtained by observing that $\psi \in W_0 \subset V_1$, where the condition

$$\int_{-\infty}^{\infty} \psi(x)\, dx = 0$$

ensures that the sequence[2] $\{g_k\}$ uniquely characterises the wavelet.

The dilation equations imply $\phi_{i,j} = \sum_k h_{k-2j} \phi_{i+1,k}$ and $\psi_{i,j} = \sum_k g_{k-2j} \phi_{i+1,j}$, and therefore

$$c_{i,j} = \sum_k h_{k-2j} c_{i+1,k} \quad \text{and} \quad d_{i,j} = \sum_k g_{k-2j} c_{i+1,k}.$$

These equations describe the "pyramid algorithm" for wavelet analysis, providing an efficient means of obtaining the approximation coefficients and detail coefficients at resolution i, given the approximation coefficients at resolution $i + 1$, as depicted in Figure C.1, where $\mathbf{c}_i = (\ldots, c_{i,-1}, c_{i,0}, c_{i,1}, \ldots)^T$ and $\mathbf{d}_i = (\ldots, d_{i,-1}, d_{i,0}, d_{i,1}, \ldots)^T$. Since each stage is equivalent to convolution by the filters with coefficients $\{h_{-k}\}$ and $\{g_{-k}\}$ followed by downsampling by a factor of two, each stage of the decomposition may be implemented as the analysis part of a two-channel filter bank.

[2] A common alternative notation employed by Chui [36] and others absorbs the factors of $\sqrt{2}$ into the sequences and denotes them as $p_k \equiv \sqrt{2}\, h_k$ and $q_k \equiv \sqrt{2}\, g_k$ instead.

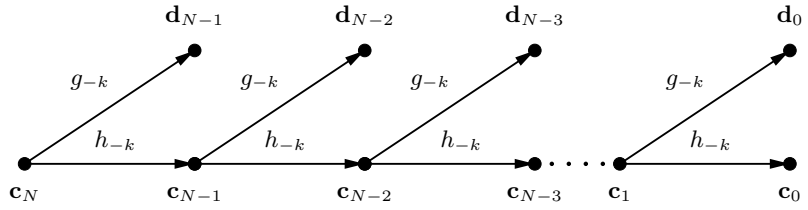
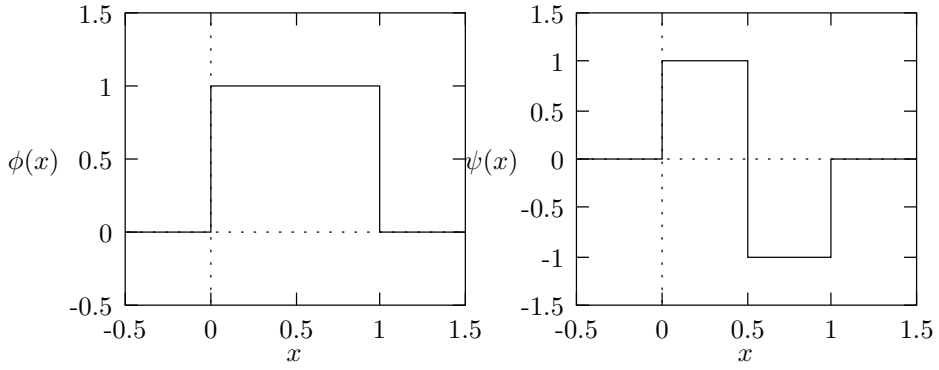


Figure C.1: The pyramid algorithm for wavelet analysis.

Given a sampled signal, the inner products required for the highest resolution $c_{i,j}$ may be calculated explicitly for a particular resolution i, with the pyramidal algorithm producing the $d_{i,j}$ and $c_{i,j}$ for lower resolutions. Alternatively, if the signal is sufficiently smooth at the resolution at which it is sampled, these samples may be used as the initial $c_{i,j}$ [197, pp. 270-271].

It may be shown that finite $\{h_k\}$ and $\{g_k\}$ filter sequences correspond to a scaling function and wavelet with compact support[3] [45, ch. 6] [189, pp. 185-186]. In this case it is convenient to construct the scaling function and wavelet by designing appropriate filter sequences (the dilation equations represent necessary, but not sufficient conditions on the scaling function and wavelet; additional restrictions are required to ensure that specific sequences correspond to a suitable scaling function and wavelet). The simplest example of a scaling function and wavelet pair, forming the Haar basis, is displayed in Figure C.2, while the corresponding filter sequences are displayed in Table C.1.



(a) Scaling Function (b) Wavelet

Figure C.2: Haar scaling function and wavelet.

[3]The support of a function $f : \mathbb{R} \to \mathbb{R}$ is the closure of the set $\{x \in \mathbb{R} \,|\, f(x) \neq 0\}$ [166, pg. 246]. The function has compact support if its support is some closed interval $[a, b]$ in $\mathbb{R}$.

Table C.1: Haar MRA dilation equation coefficients.

k	h_k	g_k
0	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$
1	$\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{2}}{2}$

C.2 Orthogonal Wavelets

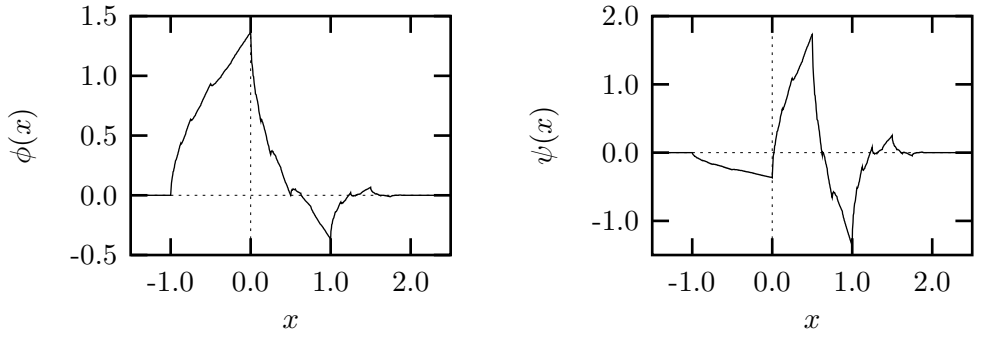
The imposition of the orthonormality requirement

$$\langle \phi_{0,0}, \phi_{0,k} \rangle = \delta_{0,k}$$

on the scaling function of an MRA results in an *orthogonal* MRA, in which $\{\phi(x-j)\}_{j\in\mathbb{Z}}$ is an orthonormal basis of V_0, and $\{\phi_{i,j}\}_{j\in\mathbb{Z}}$ is therefore an orthonormal basis of V_i. Each detail space W_i is taken as the *orthogonal* complement of V_i in V_{i+1}, so that $V_i \perp W_i \ \forall i \in \mathbb{Z}$ and therefore $W_i \perp W_j \ \forall i, j \in \mathbb{Z}, i \neq j$ and ψ (called an orthogonal wavelet) forms an orthonormal basis $\{\psi(x-j)\}_{j\in\mathbb{Z}}$ of W_0. Since distinct detail spaces are mutually orthogonal, $\{\psi_{i,j}\}_{i,j\in\mathbb{Z}}$ is an orthonormal basis of $L^2(\mathbb{R})$. It may be shown [45, ch. 5] [189] that an orthogonal wavelet may be derived from an orthogonal scaling function by choosing

$$g_k = (-1)^k h_{1-k}.$$

An example of an orthogonal scaling function and wavelet is displayed in Figure C.3, the corresponding filter coefficients being tabulated in Table C.2.



(a) Scaling Function (b) Wavelet

Figure C.3: Daubechies 4-coefficient scaling function and wavelet.

Since $\{\phi_{i,j}\}_{j\in\mathbb{Z}}$ and $\{\psi_{i,j}\}_{j\in\mathbb{Z}}$ are orthonormal bases of V_i and W_i respectively, the projec-

Table C.2: Filter coefficients [45, pg. 195] for scaling function in Figure C.3.

k	0	1	2	3
$\frac{h_k}{\sqrt{2}}$	$\frac{1+\sqrt{3}}{8}$	$\frac{3+\sqrt{3}}{8}$	$\frac{3-\sqrt{3}}{8}$	$\frac{1-\sqrt{3}}{8}$

tion operators into the approximation spaces and detail spaces are

$$P_i f = \sum_j \langle f, \phi_{i,j} \rangle \phi_{i,j} = \sum_j c_{i,j} \phi_{i,j}$$

and

$$Q_i f = \sum_j \langle f, \psi_{i,j} \rangle \psi_{i,j} = \sum_j d_{i,j} \psi_{i,j}$$

respectively.

It may be shown, using the dilation equations and the orthonormality of the bases of V_i and W_i, that $\langle \phi_{i+1,j}, \phi_{i,k} \rangle = h_{j-2k}$ and $\langle \phi_{i+1,j}, \psi_{i,k} \rangle = g_{j-2k}$. Observing that $P_{i+1}\phi_{i+1,j} = \phi_{i+1,j}$ since $\phi_{i+1,j} \in V_{i+1}$, and noting that $P_{i+1}f = P_i f + Q_i f$, it follows that $\phi_{i+1,j} = \sum_k h_{j-2k}\phi_{i,k} + \sum_k g_{j-2k}\psi_{i,k}$ and therefore

$$c_{i+1,j} = \sum_k h_{j-2k} c_{i,k} + \sum_k g_{j-2k} d_{i,k}.$$

This equation describes the pyramid algorithm for orthogonal wavelet synthesis, depicted in Figure C.4. Each stage may be implemented as the synthesis part of a two-channel filter bank with filters $\{h_k\}$ and $\{g_k\}$.

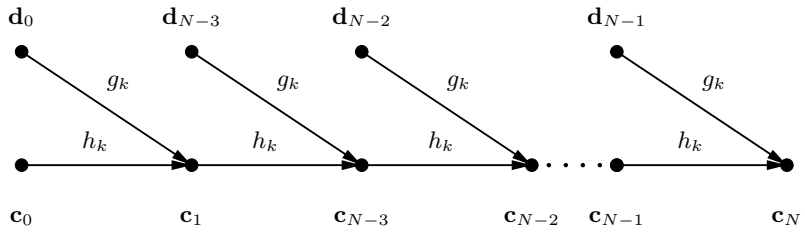


Figure C.4: The pyramid algorithm for orthogonal wavelet synthesis.

C.3 Biorthogonal Wavelets

Relaxation of the orthogonality requirement allows additional flexibility in the design of wavelet bases. The primary wavelet basis is constructed using an MRA $\{V_i\}$ with scaling function ϕ as before, while the dual Riesz basis required to form a biorthogonal pair is con-

structed by introducing a *dual* MRA $\{\tilde{V}_i\}$ with dual scaling function $\tilde{\phi}$. Biorthogonality of the bases $\{\phi_{0,j}\}_{j\in\mathbb{Z}}$ of V_0 and $\{\tilde{\phi}_{0,j}\}_{j\in\mathbb{Z}}$ of $\tilde{V}_0$, which implies biorthogonality of the bases $\{\phi_{i,j}\}_{j\in\mathbb{Z}}$ of V_i and $\{\tilde{\phi}_{0,j}\}_{j\in\mathbb{Z}}$ of $\tilde{V}_i$, is ensured by the condition

$$\langle \phi_{0,0}, \tilde{\phi}_{0,k} \rangle = \delta_{0,k}.$$

The detail spaces and dual detail spaces are taken as the complements of V_i in V_{i+1} and $\tilde{V}_i$ in $\tilde{V}_{i+1}$ respectively, where the biorthogonality condition implies that $\tilde{V}_i \perp W_i$ and $V_i \perp \tilde{W}_i$ for all $i \in \mathbb{Z}$, and $\tilde{W}_i \perp W_j \ \forall i, j \in \mathbb{Z}, i \neq j$. The bases $\{\psi_{i,j}\}_{j\in\mathbb{Z}}$ of W_i and $\{\tilde{\psi}_{i,j}\}_{j\in\mathbb{Z}}$ of $\tilde{W}_i$ are biorthogonal, and $\{\psi_{i,j}\}_{i,j\in\mathbb{Z}}$ and $\{\tilde{\psi}_{i,j}\}_{i,j\in\mathbb{Z}}$ are biorthogonal bases of $L^2(\mathbb{R})$, where

$$\langle \psi_{i,j}, \tilde{\psi}_{k,l} \rangle = \delta_{i,k}\delta_{j,l}.$$

The dual scaling function and wavelet satisfy dilation equations involving sequences[4] $\{\tilde{h}_k\}$ and $\{\tilde{g}_k\}$ respectively, and the wavelet dilation equation coefficients may be chosen as [7]

$$g_k = (-1)^k \tilde{h}_{1-k} \quad \text{and} \quad \tilde{g}_k = (-1)^k h_{1-k}.$$

An example of a biorthogonal pair of scaling functions and wavelets is displayed in Figure C.5, the corresponding filter coefficients being tabulated in Table C.3.

Table C.3: Filter coefficients (see Table II in Antonini *et al.* [7]) for scaling functions in Figure C.5.

k	0	±1	±2	±3	±4
$\frac{h_k}{\sqrt{2}}$	0.602949	0.266864	-0.078223	-0.016864	0.026749
$\frac{\tilde{h}_k}{\sqrt{2}}$	0.557543	0.295636	-0.028772	-0.045636	

The projection operators into $\tilde{V}_i$ and $\tilde{W}_i$ are

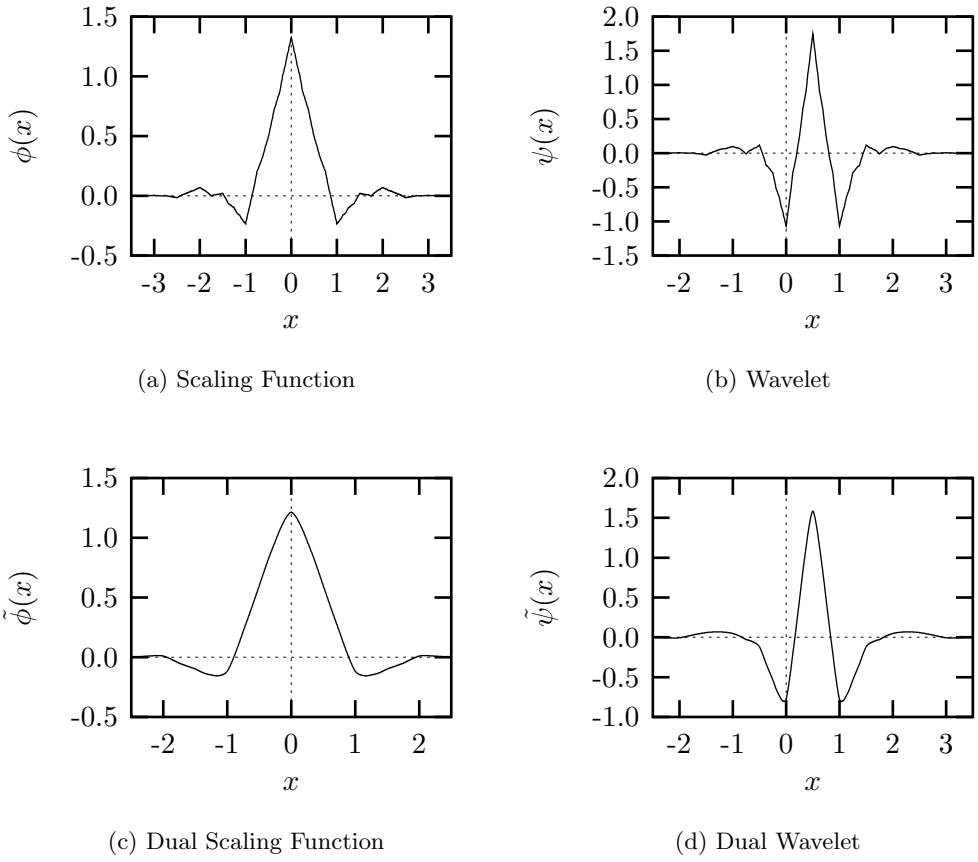
$$\tilde{P}_i f = \sum_j \langle f, \phi_{i,j} \rangle \tilde{\phi}_{i,j} = \sum_j c_{i,j} \tilde{\phi}_{i,j}$$

and

$$\tilde{Q}_i f = \sum_j \langle f, \psi_{i,j} \rangle \tilde{\psi}_{i,j} = \sum_j d_{i,j} \tilde{\psi}_{i,j}$$

respectively. These projections correspond to analysis using the primary scaling function and wavelet and reconstruction using the dual scaling function and wavelet; these roles are reversed when projecting into spaces V_i and W_i. The properties of the reconstruction basis functions are particularly important, since they represent the elementary building blocks from

[4]Chui [36] denotes the duals of h_k and g_k as $a_k \equiv \sqrt{2}\,\tilde{h}_k$ and $b_k \equiv \sqrt{2}\,\tilde{g}_k$ respectively.



(a) Scaling Function

(b) Wavelet

(c) Dual Scaling Function

(d) Dual Wavelet

Figure C.5: Spline Variant with Less Dissimilar Lengths scaling functions and wavelets.

which a function is reconstructed [45, pg. 269].

The dilation equations, together with the biorthogonality of the bases of V_i and $\tilde{V}_i$ imply that $\langle \phi_{i+1,j}, \tilde{\phi}_{i,k} \rangle = \tilde{h}_{j-2k}$ and $\langle \phi_{i+1,j}, \tilde{\psi}_{i,k} \rangle = \tilde{g}_{j-2k}$, and therefore

$$c_{i+1,j} = \sum_k \tilde{h}_{j-2k} c_{i,k} + \sum_k \tilde{g}_{j-2k} d_{i,k}.$$

This equation describes the pyramid algorithm for biorthogonal wavelet synthesis, depicted in Figure C.6. Each stage may be implemented as the synthesis part of a two-channel filter bank with filters $\{\tilde{h}_k\}$ and $\{\tilde{g}_k\}$.

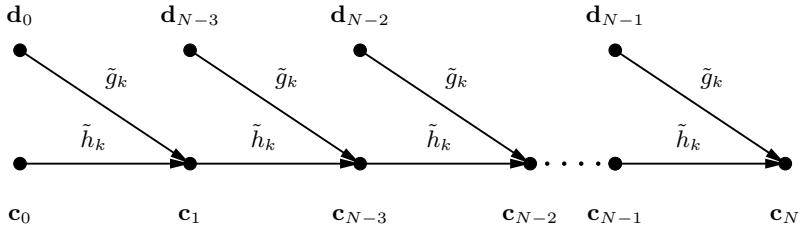


Figure C.6: The pyramid algorithm for biorthogonal wavelet synthesis.

C.4 Wavelets in Signal Processing

The wavelets described in the preceding sections generate bases of $L^2(\mathbb{R})$, and are thus not suitable for the decomposition of functions defined only on a bounded interval in $\mathbb{R}$, as is the case in most signal processing applications [40]. Given a function f defined on the interval $[0, 1)$, the simplest solution is to periodically extend f, resulting in a function $g(x) = f(x - \lfloor x \rfloor)$ which is defined over all of $\mathbb{R}$ and coincides with f where it is defined [45, pp. 304-305, 333-334] [98]. Since g has unit period, there are only 2^i distinct inner products of g with translations of the wavelet at resolution $i > 0$; the corresponding detail coefficients may therefore be arranged in a binary tree structure as illustrated in Figure C.7 (the indexing relationship between parent and child coefficients in this tree structure is depicted in Figure C.8). As a result, each stage of the pyramid algorithm may, in this context, only be applied to a signal containing an even number of samples.

Superior alternatives to periodic extension include symmetric extension [189, ch. 8] [45, pp. 333-337] and the construction of "boundary wavelets" to take care of complications at the interval boundaries [40] [98].

Desirable properties (some of which are mutually exclusive) of wavelet bases for signal processing, and signal compression in particular, include [98] [189, ch. 10,11] [197, ch. 7]:

Compact support Finite filter sequences, corresponding to a scaling function and wavelet with compact support, are necessary for efficient implementation of the pyramid al-

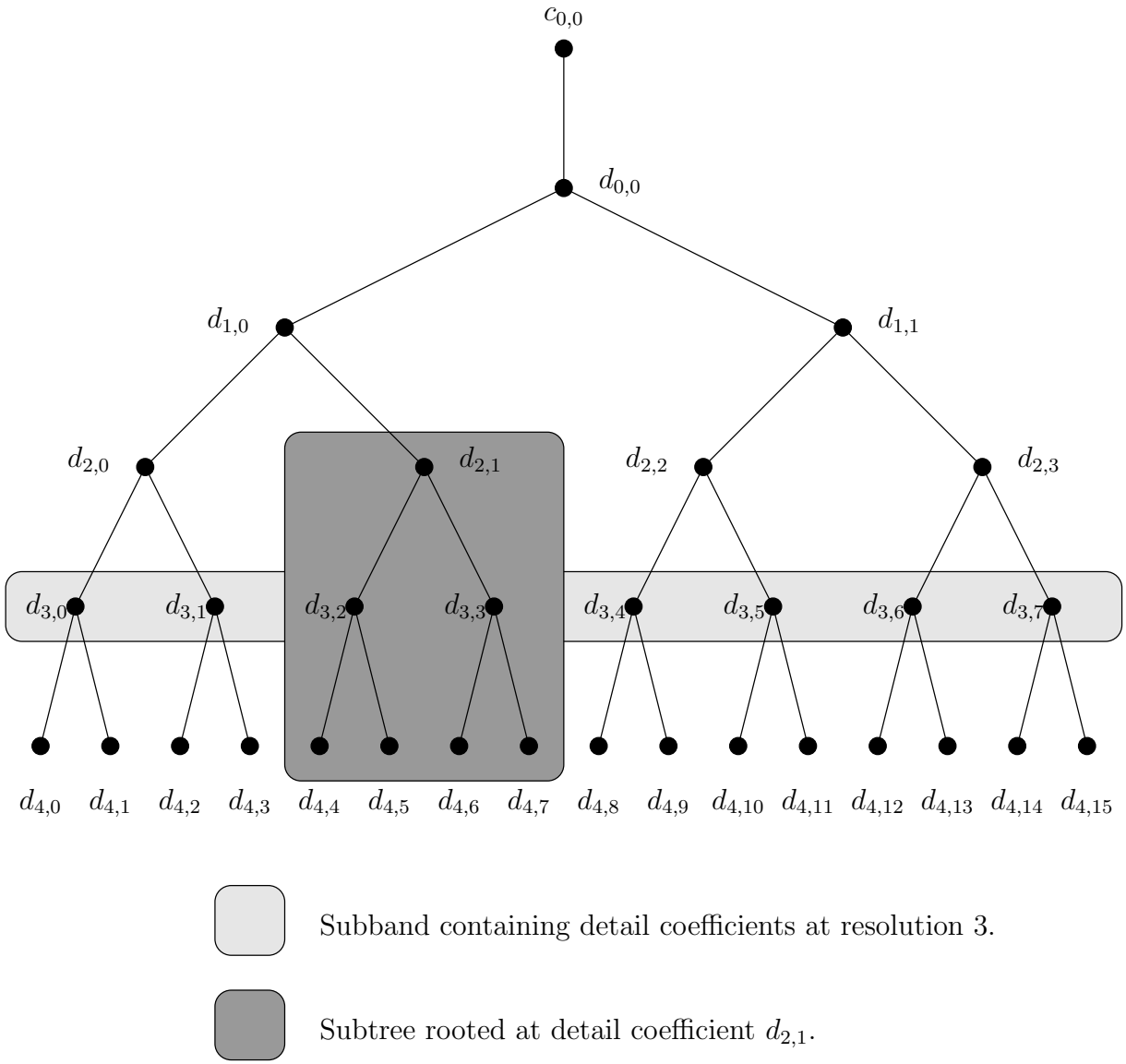


Figure C.7: Wavelet decomposition coefficient tree.

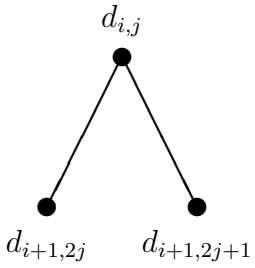


Figure C.8: Relationship between parent and child detail coefficients.

gorithm [36, pg. 21]. Shorter filter sequences decrease the computation time for the pyramid algorithm, and are desirable for good spatial localisation [197, pg. 263].

Symmetry Symmetrical wavelets and scaling functions correspond to symmetrical filter sequences, for which the analysis and synthesis operations are linear phase [36, pp. 159-168]. Symmetry of the synthesis wavelet is also desirable in order to reduce the visibility of quantisation errors in image coding [45, pp. 253-254]. Biorthogonal symmetric wavelets with compact support are often used in signal processing, since the Haar wavelet is the only orthogonal symmetrical wavelet with compact support [45, pp. 251-254].

Regularity The regularity or smoothness of the wavelet (regularity is improved with increased filter lengths) is usually described in terms of the number of continuous derivatives or its Lipschitz or Hölder regularity [45, pg. 216] [197, pp. 86-87]. The more regular the analysis wavelet, the better the frequency localisation of the decomposition. Regularity of the synthesis wavelet is also considered desirable in reducing the visibility of quantisation artifacts [98] (while there is some evidence for this assertion [7], an alternative measure based on the impulse response of the filters appears to indicate more accurately the suitability of the filters for image compression applications [198]).

Vanishing moments A wavelet ψ has M vanishing moments if $\int_{-\infty}^{\infty} \psi(x)x^m\,dx = 0$ for $0 \leq m \leq M$ [98]. The greater the number of vanishing moments of the analysis wavelet, the faster the coefficients decay with increasing resolution in smooth signal regions [45, pp. 242-245] [197, pp. 266-267]. There is also a connection between the number of vanishing moments of the analysis wavelet and the differentiability of the synthesis wavelet [45, pg. 269].

C.5 Wavelets in Image Processing

Although it is possible to design an inherently two-dimensional MRA [45, pp. 315-319], images are often analysed by two-dimensional wavelets constructed from one-dimensional wavelets. In the standard construction [45, pg. 313] [188], the two-dimensional basis consists of the functions

$$\begin{aligned}
\Phi_{i,j;k,l}(x,y) &= \phi_{i,k}(x)\phi_{j,l}(y) \\
\Psi^h_{i,j;k,l}(x,y) &= \phi_{i,k}(x)\psi_{j,l}(y) \\
\Psi^v_{i,j;k,l}(x,y) &= \psi_{i,k}(x)\phi_{j,l}(y) \\
\Psi_{i,j;k,l}(x,y) &= \psi_{i,k}(x)\psi_{j,l}(y)
\end{aligned}$$

where i and j are the horizontal and vertical resolutions respectively, and k and l are the horizontal and vertical positions respectively. The arrangement of the basis functions is depicted in Figure C.9. Such a decomposition of an image may be computed by applying the full one-dimensional decomposition to each row and then to each of the resulting columns [188].

$\Phi_{0,0;}$	$\Psi^v_{0,0;}$	$\Psi^v_{1,0;}$	$\Psi^v_{2,0;}$
$\Psi^h_{0,0;}$	$\Psi_{0,0;}$	$\Psi_{1,0;}$	$\Psi_{2,0;}$
$\Psi^h_{0,1;}$	$\Psi_{0,1;}$	$\Psi_{1,1;}$	$\Psi_{2,1;}$
$\Psi^h_{0,2;}$	$\Psi_{0,2;}$	$\Psi_{1,2;}$	$\Psi_{2,2;}$

Figure C.9: Standard two-dimensional wavelet decomposition. The block labelled as $\Psi_{2,2;}$ contains the coefficients of basis functions $\Psi_{2,2;0,0}$ through $\Psi_{2,2;3,3}$, with a similar convention adopted for the other blocks.

In the non-standard construction the two-dimensional basis consist of the functions [45, pp. 313-316]

$$\begin{aligned}
\Phi_{i;j,k}(x,y) &= \phi_{i,j}(x)\phi_{i,k}(y) \\
\Psi^h_{i;j,k}(x,y) &= \phi_{i,j}(x)\psi_{i,k}(y) \\
\Psi^v_{i;j,k}(x,y) &= \psi_{i,j}(x)\phi_{i,k}(y) \\
\Psi^d_{i;j,k}(x,y) &= \psi_{i,j}(x)\psi_{i,k}(y)
\end{aligned}$$

where i is the resolution, j and k are the horizontal and vertical positions respectively, and h, v and d represent "horizontal", "vertical" and "diagonal" respectively. The arrangement of the basis functions is depicted in Figure C.10. Such a decomposition of an image may be computed by alternating between applying a single stage of a one-dimensional decomposition to all of the rows and to all of the columns [188].

Although the non-standard construction of a two-dimensional wavelet basis is usually used in image compression, the standard construction has been reported to offer improved

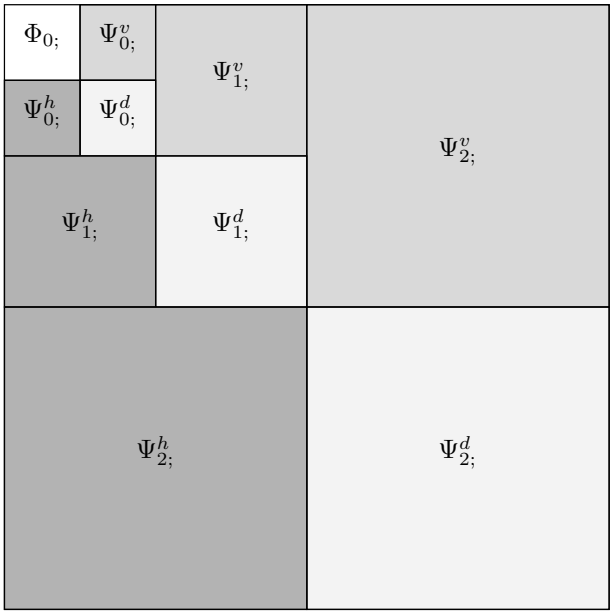
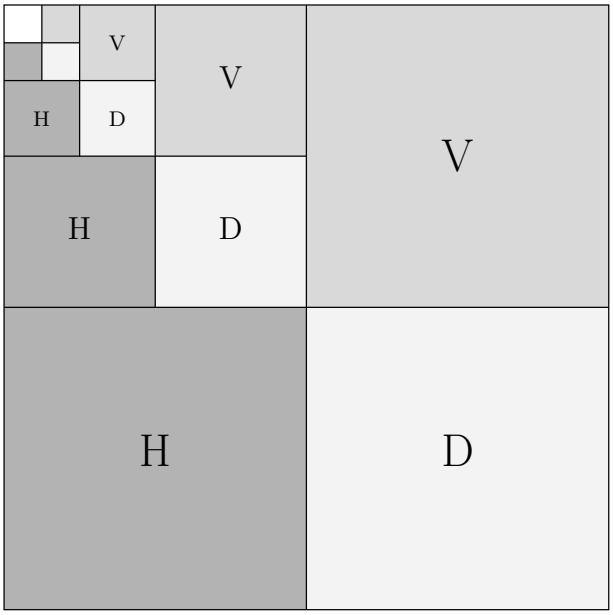


Figure C.10: Non-standard two-dimensional wavelet decomposition. The block labelled $\Psi^d_{2;}$ contains the coefficients of basis functions $\Psi^d_{2;0,0}$ through $\Psi^d_{2;3,3}$, with a similar convention adopted for the other blocks.

performance for some quantisation strategies [165]. An interpretation of the non-standard image decomposition as consisting subbands produced by directional filters is illustrated in Figure C.11, while the corresponding tree structure of the detail coefficients is illustrated in Figure C.12.



H Horizontal low frequencies and vertical high frequencies (horizontally oriented edges)

V Vertical low frequencies and horizontal high frequencies (vertically oriented edges)

D Horizontal high frequencies and vertical high frequencies (diagonally oriented edges)

Figure C.11: Horizontal, vertical and diagonal subbands in the non-standard decomposition.

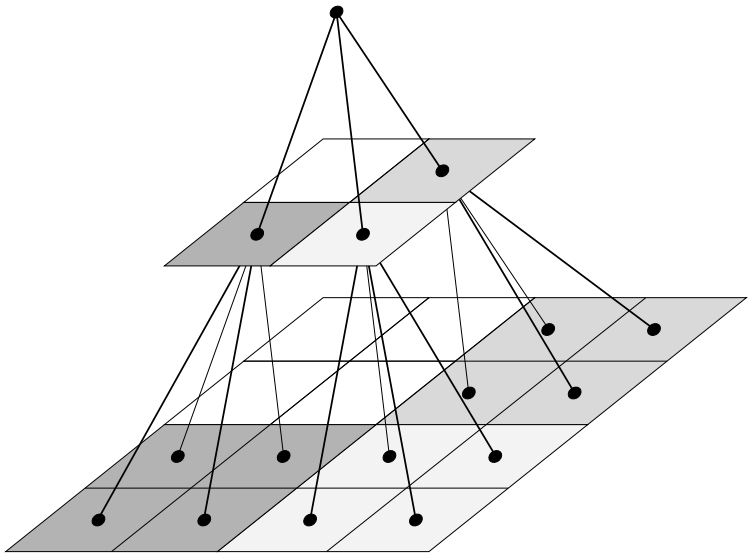


Figure C.12: Detail coefficient tree in the non-standard decomposition.

Appendix D

Mathematical Foundations of Fractal Compression

Fractal image compression is based on a result of metric space theory known as Banach's Fixed Point Theorem, which guarantees that an image may be reconstructed from its representation as a contractive transform of which it is a fixed point. The resulting coding scheme is described as "fractal" since the representation may be constructed in a finite dimensional space, but decoded in an infinite dimensional space in which detail is synthesised at all resolutions not present in the original space. The relevant results of metric space theory are summarised here.

Any point mapped to itself by a function is a fixed point of that function.

Definition 30 *A* fixed point *of a mapping $T : X \to X$ is an $x \in X$ such that $Tx = x$ [108, pg. 299].*

A contraction mapping always takes two points closer together, as illustrated in Figure D.1.

Definition 31 *A mapping $T : X \to X$ on a metric space (X, d) is a* contraction mapping *if $\exists \alpha \in \mathbb{R},\ 0 < \alpha < 1$ such that $\forall x, y \in X$*

$$d(Tx, Ty) \leq \alpha d(x, y)$$

Theorem 1 (Banach Fixed Point Theorem) *A contraction mapping $T : X \to X$ on a complete metric space (X, d) has precisely one fixed point [108, pg. 300].*

Proof: Construct a sequence $\{x_n\} \subset X$ by defining $x_n = T^n x_0$ for an arbitrary $x_0 \in X$. Contractivity of T implies

$$d(x_{n+1}, x_n) = d(T^{n+1}x_0, T^n x_0) \leq \alpha d(T^n x_0, T^{n-1}x_0) \leq \ldots \leq \alpha^n d(Tx_0, x_0)$$

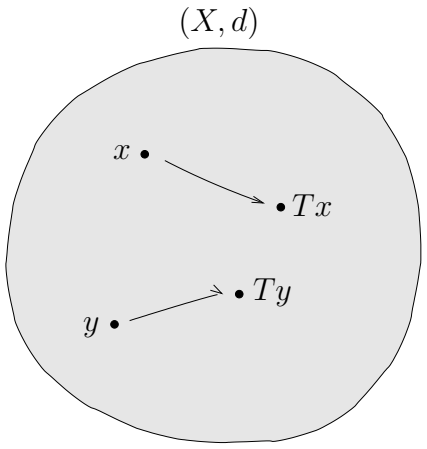


Figure D.1: Action of contraction mapping.

which implies, by repeated application of the triangle inequality (for $m, n > 0$ and $n > m$)

$$\begin{aligned} d(x_m, x_n) &\leq d(x_m, x_{m+1}) + d(x_{m+1}, x_{m+2}) + \ \ldots \ + d(x_{n-1}, x_n) \\ &= \alpha^m d(x_1, x_0) + \alpha^{m+1} d(x_1, x_0) + \ \ldots \ + \alpha^{n-1} d(x_1, x_0) \\ &= (\alpha^m + \alpha^{m+1} + \ \ldots \ + \alpha^{n-1}) d(x_1, x_0) \\ &= \alpha^m \frac{1 - \alpha^{n-m}}{1 - \alpha} d(x_0, x_1) \\ &< \frac{\alpha^m}{1 - \alpha} d(x_0, x_1) \qquad\qquad (\text{since } 1 - \alpha^{n-m} < 1). \end{aligned}$$

The sequence $\{x_n\}$ is therefore Cauchy since, taking $n > m$ and $N = \log_\alpha \frac{(1-\alpha)\epsilon}{d(x_0, x_1)}$

$$\forall \epsilon > 0 \quad m, n > N \ \Rightarrow \ d(x_m, x_n) < \epsilon.$$

Since X is complete, $\{x_n\}$ converges to some $x_T \in X$. In order to show that x_T is a fixed point, consider

$$\begin{aligned} d(x_T, Tx_T) &\leq d(x_T, x_m) + d(x_m, Tx_T) \quad \forall m \\ &\leq d(x_T, x_m) + d(x_{m-1}, x_T). \end{aligned}$$

Since $\{x_n\}$ converges to x_T it is clear that $d(x_T, Tx_T) = 0$ by taking the limit as $m \to \infty$ above. Uniqueness follows, since if $Tx_T = x_T$ and $T\tilde{x}_T = \tilde{x}_T$ then $d(x_T, \tilde{x}_T) = d(Tx_T, T\tilde{x}_T) \leq \alpha d(x_T, \tilde{x}_T)$, which implies that $d(x_T, \tilde{x}_T) = 0$. □

It is clear from the proof of Banach's theorem that the fixed point of any contraction mapping in a complete metric space may be approximated to arbitrary accuracy by iterated application of the contraction mapping to an arbitrary initial element of the metric space. The inverse problem of finding a contraction mapping having a given point as its fixed point

is considerably more difficult. Barnsley's collage theorem suggests a possible approach to this problem.

Lemma 1 *The metric $d(a,b)$ of any metric space (X,d) is continuous in $b \in X$ for fixed $a \in X$.*

Proof: The triangle inequality implies

$$\begin{aligned} d(x,z) \leq d(x,y) + d(z,y) \quad &\Rightarrow \quad d(x,z) - d(y,z) \leq d(x,y) \\ d(y,z) \leq d(y,x) + d(x,z) \quad &\Rightarrow \quad d(y,z) - d(x,z) \leq d(x,y) \end{aligned}$$

and consequently

$$| \, d(x,z) - d(y,z) \, | \leq d(x,y).$$

Defining the mapping $T_a b = d(a,b)$, and considering any convergent sequence $\{x_n\} \to x$ in X, it is clear that $T_a x_n$ converges to $T_a x$ since

$$| \, T_a x_n - T_a x \, | = | \, d(a,x_n) - d(a,x) \, | \leq d(x,x_n).$$

The mapping $T_a b = d(a,b)$ is therefore continuous, giving the required continuity of the metric. □

Theorem 2 (Collage Theorem) *If (X,d) is a complete metric space, $T : X \to X$ is a contraction mapping with contractivity $0 \leq \alpha < 1$ and fixed point x_T, then [15, pp. 94-95, 102-103]*

$$d(x,x_T) \leq (1-\alpha)^{-1} d(x,Tx) \;\; \forall x \in X.$$

Proof: The metric $d(a,b)$ is continuous in $b \in X$ for fixed $a \in X$.

$$\begin{aligned} d(x,x_T) &= d(x, \lim_{n\to\infty} T^n x) \\ &= \lim_{n\to\infty} d(x, T^n x) \qquad \text{(by Lemma 1)} \\ &\leq \sum_{m=1}^{n} d(T^{m-1}x, T^m x) \\ &\leq d(x,Tx)(1 + \alpha + \ldots + \alpha^{n-1}) \\ &\leq (1-\alpha)^{-1} d(x,Tx). \end{aligned}$$

□

Appendix E

Standard Images

Figure E.1: Airplane.

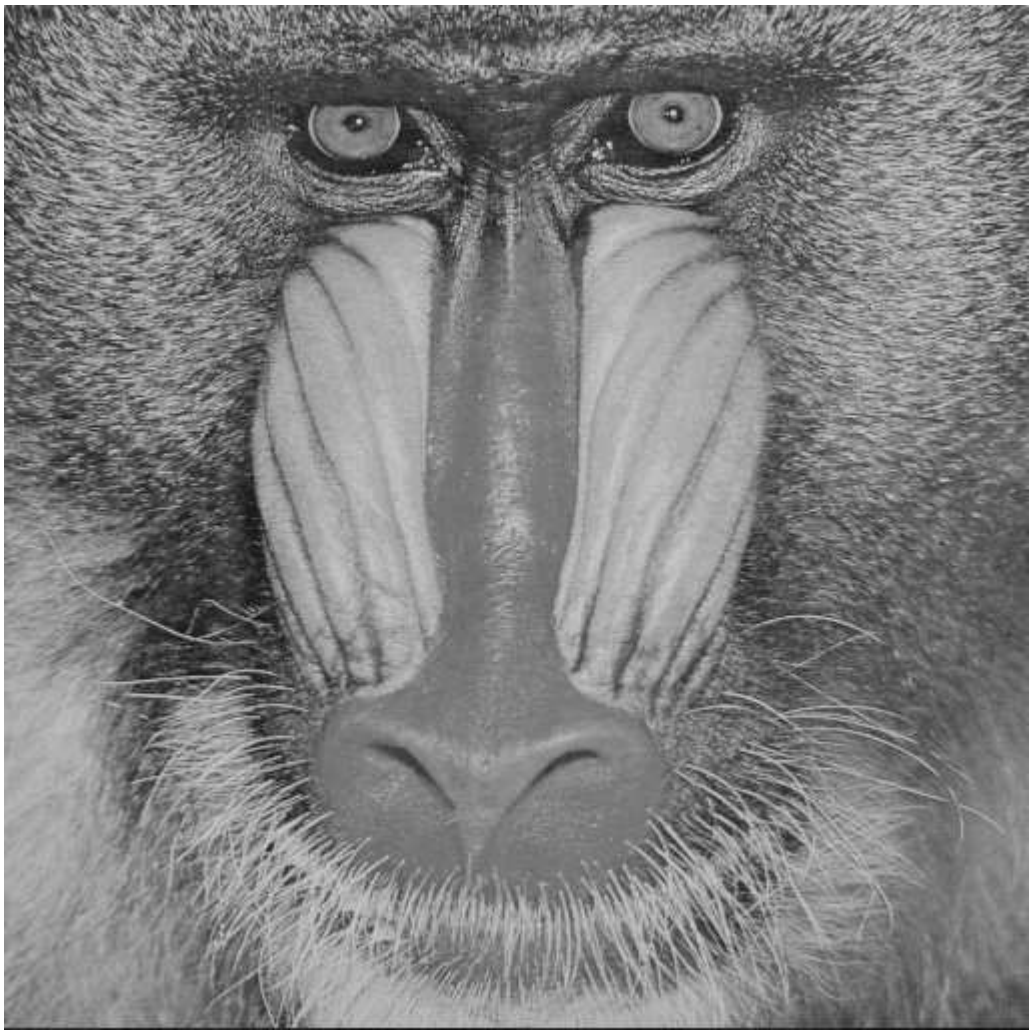


Figure E.2: Baboon.

Figure E.3: Bridge.

Figure E.4: Lena.

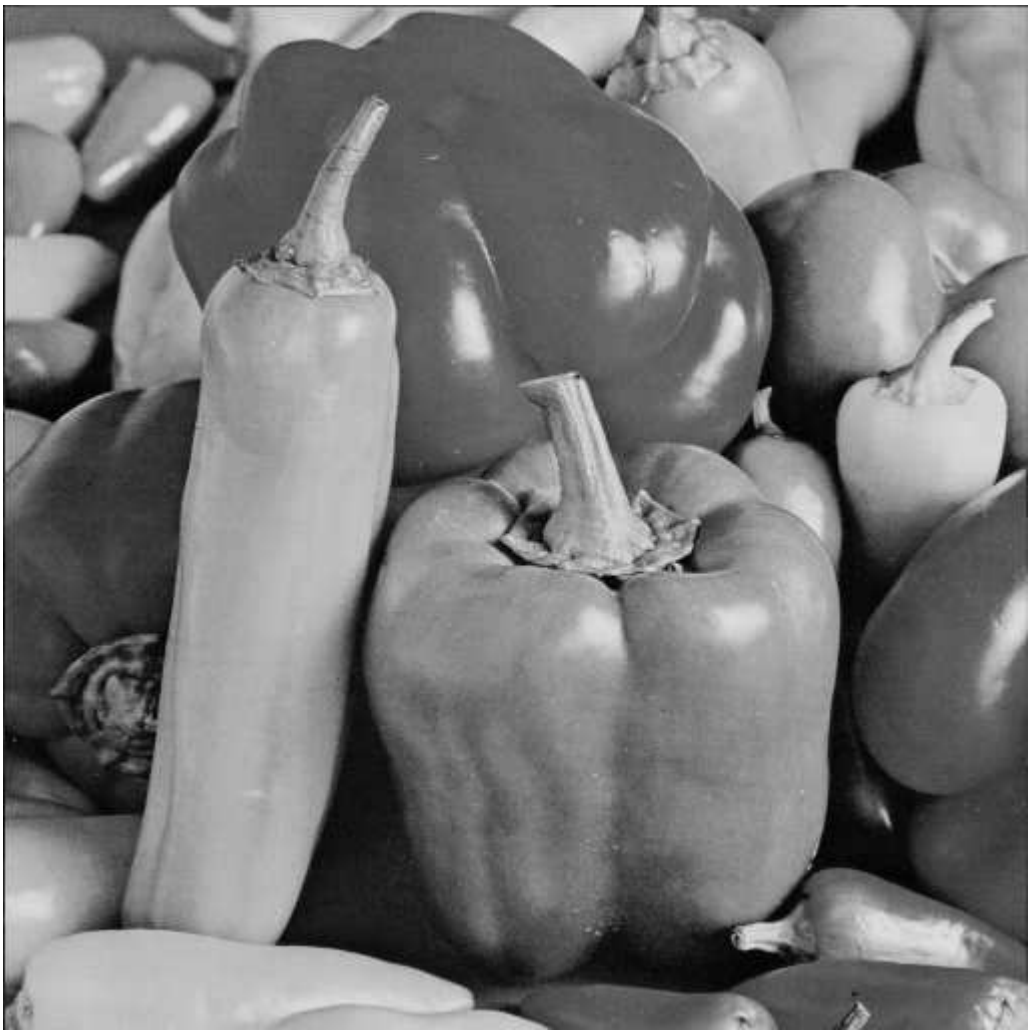


Figure E.5: Peppers.

References

[1] N. Abramson. *Information Theory and Coding*. McGraw-Hill, New York, NY, USA, 1963.

[2] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, January 1974.

[3] A. N. Akansu and R. A. Haddad. *Multiresolution Signal Decomposition*. Academic Press, San Diego, CA, USA, 1992.

[4] A. N. Akansu and M. J. T. Smith, editors. *Subband and Wavelet Transforms: Design and Applications*. Kluwer Academic Publishers, Boston, MA, USA, 1996.

[5] H. C. Andrews and C. L. Patterson. Singular value decomposition (SVD) image coding. *IEEE Transactions on Communications*, COM-24(4):425–432, April 1976.

[6] L. F. Anson. Fractal image compression. *Byte*, pages 195–202, October 1993.

[7] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 2(1):205–220, April 1992.

[8] S. Arya and D. M. Mount. Algorithms for fast vector quantization. In J. Storer and M. Cohn, editors, *Proceedings DCC'93 (IEEE Data Compression Conference)*, pages 381–390, Snowbird, UT, USA, March 1993.

[9] S. Arya and D. M. Mount. Approximate nearest neighbor queries in fixed dimensions. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 271–280, Austin, TX, USA, January 1993.

[10] G. Bachman and L. Narici. *Functional Analysis*. Academic Press, New York, NY, USA, 1966.

[11] B. Bani-Eqbal. Enhancing the speed of fractal image compression. *Optical Engineering*, 34(6):1705–1710, June 1995.

[12] B. Bani-Eqbal. Speeding up fractal image compression. In M. Rabbani, E. J. Delp, and S. A. Rajala, editors, *Still-Image Compression*, volume 2418 of *SPIE Proceedings*, pages 67–74, San Jose, CA, USA, February 1995.

[13] Z. Barahav, D. Malah, and E. Karnin. Hierarchical interpretation of fractal image coding and its application to fast decoding. In *Proceedings of the IEEE International Conference on Digital Signal Processing*, pages 190–195, Nicosia, Cyprus, July 1993.

[14] Z. Barahav, D. Malah, and E. Karnin. Hierarchical interpretation of fractal image coding and its applications. In Y. Fisher, editor, *Fractal Image Compression: Theory and Application*, chapter 5, pages 91–117. Springer-Verlag, New York, NY, USA, 1995.

[15] M. F. Barnsley. *Fractals Everywhere*. Academic Press, Cambridge, MA, USA, 2nd edition, 1993.

[16] M. F. Barnsley and L. P. Hurd. *Fractal Image Compression*. AK Peters, Wellesley, MA, USA, 1993.

[17] M. F. Barnsley and A. D. Sloan. A better way to compress images. *Byte*, pages 215–223, January 1988.

[18] K. U. Barthel. Entropy constrained fractal image coding. In *NATO ASI on Fractal Image Encoding and Analysis*, Trondheim, Norway, July 1995.

[19] K. U. Barthel, J. Schüttemeyer, T. Voyé, and P. Noll. A new image coding technique unifying fractal and transform coding. In *Proceedings ICIP-94 (IEEE International Conference on Image Processing)*, volume III, pages 112–116, Austin, TX, USA, November 1994.

[20] K. U. Barthel and T. Voyé. Adaptive fractal image coding in the frequency domain. In *Proceedings of the International Workshop on Image Processing*, volume XLV of *Theory, Methodology, Systems and Applications*, pages 33–38, Budapest, Hungary, June 1994.

[21] K. U. Barthel, T. Voyé, and P. Noll. Improved fractal image coding. In *Proceedings PCS'93 (International Picture Coding Symposium)*, page 1.5, Lausanne, Switzerland, March 1993.

[22] M. Basseville, A. Benveniste, K. C. Chou, S. A. Golden, R. Nikoukhah, and A. S. Willsky. Modelling and estimation of multiresolution stochastic processes. *IEEE Transactions on Information Theory*, 38(2 (Part II)):766–784, March 1992.

[23] J. M. Beaumont. Advances in block based fractal coding of still pictures. In *Proceedings IEE Colloquium: The Application of Fractal Techniques in Image Processing*, pages 3.1–3.6, December 1990.

[24] T. Bedford, F. M. Dekking, and M. S. Keane. Fractal image coding techniques and contraction operators. *Nieuw Archief Voor Wiskunde*, 10(3):185–217, November 1992.

[25] T. C. Bell, J. G. Cleary, and I. H. Witten. *Text Compression*. Prentice Hall, Englewood Cliffs, NJ, USA, 1990.

[26] T. Berger. *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Information and System Sciences Series. Prentice-Hall, Englewood Cliffs, NJ, USA, 1971.

[27] A. Bogdan. Multiscale fractal image coding and the two-scale difference equation. Technical Report CU/CTR/TR 358-94-05, Columbia University, New York, NY, USA, March 1994.

[28] A. Bogdan. Multiscale (inter/intra-frame) fractal video coding. In *Proceedings ICIP-94 (IEEE International Conference on Image Processing)*, volume I, pages 760–764, Austin, TX, USA, November 1994.

[29] A. Bogdan. The fractal pyramid with application to image coding. In *Proceedings ICASSP-95 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, volume 4, pages 2595–2598, Detroit, MI, USA, May 1995.

[30] A. Bogdan. *Image Coding using Iterative Transformations with Applications to Image Communications*. PhD thesis, Columbia University, New York, NY, USA, 1995.

[31] R. D. Boss and E. W. Jacobs. Archetype classification in an iterated transformation image compression algorithm. In Y. Fisher, editor, *Fractal Image Compression: Theory and Application*, chapter 4, pages 79–90. Springer-Verlag, New York, NY, USA, 1995.

[32] A. Buzo, A. H. Gray, Jr., R. M. Gray, and J. D. Markel. Speech coding based upon vector quantization. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-28(5):562–574, October 1980.

[33] S.-H. G. Chang. Image interpolation using wavelet-based edge enhancement and texture analysis. Master's thesis, University of California, Berkeley, Berkeley, CA, USA, 1995.

[34] S.-H. G. Chang, Z. Cvetković, and M. Vetterli. Resolution enhancement of images using wavelet transform extrema extrapolation. In *Proceedings ICASSP-95 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, volume 4, pages 2379–2382, Detroit, MI, USA, May 1995.

[35] B. CHENG AND X. ZHU. Multiresolution approximation of fractal transform. Submitted to *Signal Processing*, 1997. Preprint.

[36] C. K. CHUI. *An Introduction to Wavelets*, volume 1 of *Wavelet Analysis and its Applications*. Academic Press, San Diego, CA, USA, 1992.

[37] R. J. CLARKE. *Transform Coding of Images*. Number 4 in Microelectronics and Signal Processing. Academic Press, London, England, 1985.

[38] R. J. CLARKE AND L. M. LINNETT. Fractals and image representation. *Electronics & Communication Engineering Journal*, 5(4):233–239, August 1993.

[39] A. COHEN. Biorthogonal wavelets. In C. K. CHUI, editor, *Wavelets: A Tutorial in Theory and Applications*, volume 2 of *Wavelet Analysis and its Applications*, pages 123–152. Academic Press, San Diego, CA, USA, 1992.

[40] A. COHEN, I. DAUBECHIES, AND P. VIAL. Wavelets on the interval and fast wavelet transforms. *Applied and Computational Harmonic Analysis*, 1(1):54–81, December 1993.

[41] J. H. CONWAY AND N. J. A. SLOANE. *Sphere Packings, Lattices and Groups*. Springer-Verlag, New York, NY, USA, 2nd edition, 1993.

[42] P. C. COSMAN, R. M. GRAY, AND R. A. OLSHEN. Evaluating quality of compressed medical images: SNR, subjective rating, and diagnostic accuracy. *Proceedings of the IEEE*, 82(6):919–932, June 1994.

[43] P. C. COSMAN, R. M. GRAY, AND M. VETTERLI. Vector quantization of image subbands: A survey. *IEEE Transactions on Image Processing*, 5(2):202–225, February 1996.

[44] T. M. COVER AND J. A. THOMAS. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley & Sons, New York, NY, USA, 1991.

[45] I. DAUBECHIES. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

[46] G. DAVIS. Adaptive self-quantization of wavelet subtrees: A wavelet-based theory of fractal image compression. In A. F. LAINE, M. A. UNSER, AND M. V. WICKERHAUSER, editors, *Wavelet Applications in Signal and Image Processing III*, volume 2569 of *SPIE Proceedings*, pages 294–307, San Diego, CA, USA, July 1995.

[47] G. DAVIS. Self-quantized wavelet subtrees: A wavelet-based theory of fractal image compression. In H. H. SZU, editor, *Wavelet Applications II*, volume 2491 of *SPIE Proceedings*, pages 141–152, Orlanda, FL, USA, April 1995.

[48] F. Davoine, M. Antonini, J.-M. Chassery, and M. Barlaud. Fractal image compression based on Delaunay triangulation and vector quantization. *IEEE Transactions on Image Processing*, 5(2):338–346, February 1996.

[49] F. Davoine, E. Bertin, and J.-M. Chassery. From rigidity to adaptive tessellations for fractal image compression: Comparative studies. In *Proceedings of the 8th IEEE Workshop on Image and Multidimensional Signal Processing*, pages 56–57, Cannes, France, September 1993.

[50] F. Davoine and J.-M. Chassery. Adaptive Delaunay triangulation for attractor image coding. In *Proceedings of the 12th International Conference on Pattern Recognition*, pages 801–803, Jerusalem, Israel, October 1994.

[51] F. Davoine, J. Svensson, and J.-M. Chassery. A mixed triangular and quadrilateral partition for fractal image coding. In *Proceedings ICIP-95 (IEEE International Conference on Image Processing)*, volume III, pages 284–287, Washington, D.C., USA, October 1995.

[52] M. H. DeGroot. *Probability and Statistics*. Addison-Wesley Series in Statistics. Addison-Wesley, Reading, MA, USA, 2nd edition, 1986.

[53] A. S. Deif. *Advanced Matrix Theory for Scientists and Engineers*. John Wiley & Sons, New York, NY, USA, 1982.

[54] R. Dettmer. Form and function: Fractal-based image compression. *IEE Review*, 38(9):323–327, 1992.

[55] R. W. Dijkerman and R. R. Muzumdar. Wavelet representations of stochastic processes and multiresolution stochastic models. *IEEE Transactions on Signal Processing*, 42(7):1640–1652, July 1994.

[56] J. Domaszewicz and V. A. Vaishampayan. Iterative collage coding for fractal compression. In *Proceedings ICIP-94 (IEEE International Conference on Image Processing)*, volume III, pages 127–131, Austin, TX, USA, November 1994.

[57] A. W. Drake. *Fundamentals of Applied Probability Theory*. McGraw-Hill Series in Probability and Statistics. McGraw-Hill, New York, NY, USA, 1967.

[58] H. B. Dwight. *Tables of Integrals and other Mathematical Data*. The McMillan Company, New York, NY, USA, 4th edition, 1961.

[59] W. H. Equitz. A new vector quantization clustering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(10):1568–1575, October 1989.

[60] K. FALCONER. *Fractal Geometry: Mathematical Foundations and Applications*. John Wiley & Sons, Chichester, England, 1990.

[61] T. R. FISCHER. A pyramid vector quantizer. *IEEE Transactions on Information Theory*, IT-32(4):568–583, July 1986.

[62] Y. FISHER, E. W. JACOBS, AND R. D. BOSS. Fractal image compression using iterated transforms. In J. A. STORER, editor, *Image and Text Compression*, chapter 2, pages 35–61. Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[63] Y. FISHER, T. P. SHEN, AND D. ROGOVIN. Comparison of fractal methods with discrete cosine transform (DCT) and wavelets. In S.-S. CHEN, editor, *Neural and Stochastic Methods in Image and Signal Processing III*, volume 2304 of *SPIE Proceedings*, pages 132–143, San Diego, CA, USA, July 1994.

[64] Y. FISHER. A discussion of fractal image compression. In H.-O. PEITGEN, H. JÜRGENS, AND D. SAUPE, editors, *Chaos and Fractals: New Frontiers of Science*, appendix A, pages 903–919. Springer-Verlag, New York, NY, USA, 1992.

[65] Y. FISHER, editor. *Fractal Image Compression: Theory and Application*. Springer-Verlag, New York, NY, USA, 1995.

[66] P. FLANDRIN. On the spectrum of fractional Brownian motions. *IEEE Transactions on Information Theory*, 35(1):197–199, January 1989.

[67] P. FLANDRIN. Wavelet analysis and synthesis of fractional Brownian motion. *IEEE Transactions on Information Theory*, 38(2 (Part II)):910–917, March 1992.

[68] J. H. FRIEDMAN, J. L. BENTLEY, AND R. A. FINKEL. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, September 1977.

[69] C. FRIGAARD, J. GADE, T. HEMMINGSEN, AND T. SAND. Image compression based on a fractal theory. Internal Report S701, Institute for Electronic Systems, Aalborg University, Aalborg, Denmark, 1994.

[70] R. G. GALLAGER. *Information Theory and Reliable Communication*. John Wiley & Sons, New York, NY, USA, 1968.

[71] N. GARGUIR. Comparative performance of SVD and adaptive cosine transform in coding images. *IEEE Transactions on Communications*, COM-27(8):1230–1234, August 1979.

[72] A. GERSHO. Asymptotically optimal block quantization. *IEEE Transactions on Information Theory*, IT-25(4):373–380, July 1979.

[73] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Communications and Information Theory. Kluwer Academic Publishers, Norwell, MA, USA, 1992.

[74] M. Gharavi-Alkhansari and T. S. Huang. Fractal-based techniques for a generalized image coding method. In *Proceedings ICIP-94 (IEEE International Conference on Image Processing)*, volume III, pages 122–126, Austin, TX, USA, November 1994.

[75] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, Reading, MA, USA, 3rd edition, 1992.

[76] D. Götting, A. Ibenthal, and R.-R. Grigat. Fractal image coding and magnification using invariant features. In *NATO ASI on Fractal Image Encoding and Analysis*, Trondheim, Norway, July 1995.

[77] N. C. Griswold. Perceptual coding in the cosine transform domain. *Optical Engineering*, 19(3):306–311, May/June 1980.

[78] R. W. Hamming. *Coding and Information Theory*. Prentice-Hall, London, England, 1980.

[79] R. Hamzaoui. Codebook clustering by self-organizing maps for fractal image compression. In *NATO ASI on Fractal Image Encoding and Analysis*, Trondheim, Norway, July 1995.

[80] R. Hamzaoui, M. Müller, and D. Saupe. VQ-enhanced fractal image compression. In *Proceedings ICIP-96 (IEEE International Conference on Image Processing)*, volume I, pages 153–156, Lausanne, Switzerland, September 1996.

[81] C. W. Helstrom. *Probability and Stochastic Processes for Engineers*. Macmillan Publishing Company, New York, NY, USA, 1984.

[82] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, New York, NY, USA, 1994.

[83] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, September 1952.

[84] B. Hürtgen. Contractivity of fractal transforms for image coding. *Electronics Letters*, 29(20):1749–1750, September 1993.

[85] B. Hürtgen. Performance bounds for fractal coding. In *Proceedings ICASSP-95 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, volume 4, pages 2563–2566, Detroit, MI, USA, May 1995.

[86] B. HÜRTGEN. Statistical evaluation of fractal coding schemes. In *Proceedings ICIP-95 (IEEE International Conference on Image Processing)*, volume III, pages 280–283, Washington, D.C., USA, October 1995.

[87] B. HÜRTGEN AND T. HAIN. On the convergence of fractal transforms. In *Proceedings ICASSP-94 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, volume 5, pages 561–564, Adelaide, Australia, April 1994.

[88] B. HÜRTGEN, P. MOLS, AND S. F. SIMON. Fractal transform coding of color images. In A. K. KATSAGGELOS, editor, *Visual Communications and Image Processing '94*, volume 2308 of *SPIE Proceedings*, pages 1683–1691, Chicago, IL, USA, July 1994.

[89] B. HÜRTGEN AND F. MÜLLER. Modelling of fractal coding schemes. In *Proceedings EUSIPCO'94 (European Signal Processing Conference)*, volume 1, pages 600–603, Edinburgh, Scotland, September 1994.

[90] B. HÜRTGEN AND S. F. SIMON. On the problem of convergence in fractal coding schemes. In *Proceedings ICIP-94 (IEEE International Conference on Image Processing)*, volume III, pages 103–106, Austin, TX, USA, November 1994.

[91] B. HÜRTGEN AND C. STILLER. Fast hierarchical codebook search for fractal coding of still images. In R. A. MATTHEUS, A. J. DUERINCKX, AND P. J. VAN OTTERLOO, editors, *Video Communications and PACS for Medical Applications*, volume 1977 of *SPIE Proceedings*, pages 397–408, Berlin, Germany, April 1993.

[92] A. E. JACQUIN. Fractal image coding based on a theory of iterated contractive image transformations. In M. KUNT, editor, *Visual Communications and Image Processing '90*, volume 1360 of *SPIE Proceedings*, pages 227–239, Lausanne, Switzerland, October 1990.

[93] A. E. JACQUIN. A novel fractal block-coding technique for digital images. In *Proceedings ICASSP-90 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, volume 4, pages 2225–2228, Alberquerque, NM, USA, April 1990.

[94] A. E. JACQUIN. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transactions on Image Processing*, 1(1):18–30, January 1992.

[95] A. E. JACQUIN. Fractal image coding: A review. *Proceedings of the IEEE*, 81(10):1451–1465, October 1993.

[96] A. K. JAIN. Advances in mathematical models for image processing. *Proceedings of the IEEE*, 69(5):502–28, May 1981.

[97] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, Englewood Cliffs, NJ, USA, 1989.

[98] B. Jawerth and W. Sweldens. An overview of wavelet based multiresolution analysis. *SIAM Review*, 36(3):377–412, 1994.

[99] N. S. Jayant and P. Noll. *Digital Coding of Waveforms*. Prentice-Hall Signal Processing Series. Prentice-Hall, Englewood Cliffs, NJ, USA, 1984.

[100] H. Kardestuncer and D. H. Norrie, editors. *Finite Element Handbook*. McGraw-Hill, New York, NY, USA, 1987.

[101] M. Kawamata, M. Nagahisa, and T. Higuchi. Multi-resolution tree search for iterated transformation theory-based coding. In *Proceedings ICIP-94 (IEEE International Conference on Image Processing)*, volume III, pages 137–141, Austin, TX, USA, November 1994.

[102] D. Kersten. Predictability and redundancy of natural images. *Journal of the Optical Society of America A*, 4(12):2395–2400, December 1987.

[103] C.-S. Kim, R.-C. Kim, and S.-U. Lee. Novel fractal image compression method with non-iterative decoder. In *Proceedings ICIP-95 (IEEE International Conference on Image Processing)*, volume III, pages 268–271, Washington, D.C., USA, October 1995.

[104] K. Kim and R.-H. Park. Image coding based on fractal approximation and vector quantization. In *Proceedings ICIP-94 (IEEE International Conference on Image Processing)*, volume III, pages 132–136, Austin, TX, USA, November 1994.

[105] T. Kim. Side match and overlap match vector quantizers for images. *IEEE Transactions on Image Processing*, 1(2):170–185, April 1992.

[106] J. Kominek. Algorithm for fast fractal image compression. In A. A. Rodriguez, R. J. Safranek, and E. J. Delp, editors, *Digital Video Compression: Algorithms and Technologies 1995*, volume 2419 of *SPIE Proceedings*, pages 296–305, San Jose, CA, USA, February 1995.

[107] J. Kominek. Convergence of fractal encoded images. In J. A. Storer and M. Cohn, editors, *Proceedings DCC'95 (IEEE Data Compression Conference)*, pages 242–251, Snowbird, UT, USA, March 1995.

[108] E. Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley & Sons, New York, NY, USA, 1978.

[109] H. KRUPNIK, D. MALAH, AND E. KARNIN. Fractal representation of images via the discrete wavelet transform. In *Proceedings of the 18th IEEE Convention of Electrical and Electronics Engineers in Israel*, pages 2.2.2/1–5, Tel-Aviv, Israel, March 1995.

[110] H. J. LARSON. *Introduction to Probability Theory and Statistical Inference*. John Wiley & Sons, New York, NY, USA, 2nd edition, 1974.

[111] A. LEMPEL AND J. ZIV. On the complexity of finite sequences. *IEEE Transactions on Information Theory*, IT-23(1):75–81, January 1976.

[112] S. LEPSØY. *Attractor Image Compression - Fast Algorithms and Comparisons to Related Techniques*. PhD thesis, The Norwegian Institute of Technology, Trondheim, Norway, June 1993.

[113] S. LEPSØY AND G. E. ØIEN. Fast attractor image encoding by adaptive codebook clustering. In Y. FISHER, editor, *Fractal Image Compression: Theory and Application*, chapter 9, pages 177–197. Springer-Verlag, New York, NY, USA, 1995.

[114] A. S. LEWIS AND G. KNOWLES. Image compression using the 2-D wavelet transform. *IEEE Transactions on Image Processing*, 2(1):244–250, April 1992.

[115] J. S. LIM. *Two-dimensional Signal and Image Processing*. Prentice Hall Signal Processing Series. Prentice-Hall, Englewood Cliffs, NJ, USA, 1990.

[116] J. O. LIMB. Entropy of quantised television signals. *Proceedings of the Institution of Electrical Engineers*, 115(1):16–20, January 1968.

[117] J. O. LIMB. Distortion criteria of the human viewer. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-9(12):778–93, December 1979.

[118] D. W. LIN. Fractal image coding as generalized predictive coding. In *Proceedings ICIP-94 (IEEE International Conference on Image Processing)*, volume III, pages 117–121, Austin, TX, USA, November 1994.

[119] H. LIN AND A. VENETSANOPOULOS. A pyramid algorithm for fast fractal image compression. In *Proceedings ICIP-95 (IEEE International Conference on Image Processing)*, volume III, pages 596–599, Washington, D.C., USA, October 1995.

[120] S. P. LLOYD. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, IT-28(2):129–137, March 1982.

[121] G. LU AND T.-L. YEW. Image compression using partitioned iterated function systems. In M. RABBANI AND R. J. SAFRANEK, editors, *Image and Video Compression*, volume 2186 of *SPIE Proceedings*, pages 122–133, San Jose, CA, USA, February 1994.

[122] G. Lu and T.-L. Yew. Image compression using quadtree partitioned iterated function systems. *Electronics Letters*, 30(1):23–24, January 1994.

[123] L. M. Lundheim. A discrete framework for fractal signal modeling. In Y. Fisher, editor, *Fractal Image Compression: Theory and Application*, chapter 7, pages 137–151. Springer-Verlag, New York, NY, USA, 1995.

[124] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[125] S. G. Mallat and W. L. Hwang. Singularity detection and processing with wavelets. *IEEE Transactions on Information Theory*, 38(2 (Part II)):617–643, March 1992.

[126] S. G. Mallat and S. Zhong. Characterization of signals from multiscale edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(7):710–732, July 1992.

[127] J. L. Mannos and D. J. Sakrison. The effects of a visual criterion on the encoding of images. *IEEE Transactions on Information Theory*, IT-20(4):525–536, July 1974.

[128] E. Masry. The wavelet transform of stochastic processes with stationary increments and its applications to fractional Brownian motion. *IEEE Transactions on Information Theory*, 39(1):260–264, January 1993.

[129] W. Mauersberger. Generalised correlation model for designing 2-dimensional image coders. *Electronics Letters*, 15(20):664–665, September 1979.

[130] W. Mauersberger. Experimental results on the performance of mismatched quantizers. *IEEE Transactions on Information Theory*, IT-25(4):381–386, July 1981.

[131] J. Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, IT-6(1):7–12, March 1960.

[132] J. M. Mendel. Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications. *Proceedings of the IEEE*, 79(3):278–305, March 1991.

[133] D. M. Monro. Class of fractal transforms. *Electronics Letters*, 29(4):362–363, February 1993.

[134] D. M. Monro and F. Dudbridge. Fractal approximation of image blocks. In *Proceedings ICASSP-92 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, volume 3, pages 485–488, San Francisco, CA, USA, March 1992.

[135] D. M. MONRO AND F. DUDBRIDGE. Fractal block coding of images. *Electronics Letters*, 28(11):1053–1055, May 1992.

[136] D. M. MONRO AND F. DUDBRIDGE. Rendering algorithms for deterministic fractals. *IEEE Computer Graphics and Applications*, 15(1):32–41, January 1995.

[137] D. M. MONRO AND S. J. WOOLLEY. Fractal image compression without searching. In *Proceedings ICASSP-94 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, volume 5, pages 557–560, Adelaide, Australia, April 1994.

[138] D. M. MONRO AND S. J. WOOLLEY. Rate/distortion in fractal compression: Order of transform and block symmetries. In *Proceedings ISSIPNN'94 (International Symposium on Speech, Image Processing and Neural Networks)*, volume 1, pages 168–171, Hong Kong, April 1994.

[139] T. MURAKAMI, K. ASAI, AND E. YAMAZAKI. Vector quantiser of video signals. *Electronics Letters*, 18(23):1005–1006, November 1982.

[140] T. R. NATARAJAN AND N. AHMED. Performance evaluation for transform coding using a nonseparable covariance model. *IEEE Transactions on Communications*, COM-26(2):310–312, February 1978.

[141] A. W. NAYLOR AND G. R. SELL. *Linear Operator Theory in Engineering and Science*. Holt, Rinehart and Winston, New York, NY, USA, 1971.

[142] A. N. NETRAVELI AND B. G. HASKELL. *Digital Pictures: Representation, Compression and Standards*. Applications of Communications Theory. Plenum Press, New York, NY, USA, 2nd edition, 1995.

[143] N. B. NILL. A visual model weighted cosine transform for image compression and quality assessment. *IEEE Transactions on Communications*, COM-33(6):551–557, July 1985.

[144] M. NOVAK. Attractor coding of images. In *Proceedings PCS'93 (International Picture Coding Symposium)*, page 15.6, Lausanne, Switzerland, March 1993.

[145] M. NOVAK. Attractor coding of images. Licentiate dissertation, Linköping University, Linköping, Sweden, May 1993.

[146] G. E. ØIEN. Parameter quantization in fractal image coding. In *Proceedings ICIP-94 (IEEE International Conference on Image Processing)*, volume III, pages 142–146, Austin, TX, USA, November 1994.

[147] G. E. Øien, Z. Baharav, S. Lepsøy, E. Karnin, and D. Malah. A new improved collage theorem with applications to multiresolution fractal image coding. In *Proceedings ICASSP-94 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, volume 5, pages 565–568, Adelaide, Australia, April 1994.

[148] G. E. Øien and S. Lepsøy. A class of fractal image coders with fast decoder convergence. In Y. Fisher, editor, *Fractal Image Compression: Theory and Application*, chapter 8, pages 153–175. Springer-Verlag, New York, NY, USA, 1995.

[149] G. E. Øien, S. Lepsøy, and T. A. Ramstad. An inner product space approach to image coding by contractive transformations. In *Proceedings ICASSP-91 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, volume 4, pages 2773–2776, Toronto, Canada, May 1991.

[150] M. D. Paez and T. H. Glisson. Minimum mean-squared-error quantization in speech PCM and DPCM systems. *IEEE Transactions on Communications*, COM-20(2):225–230, April 1972.

[151] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, NY, USA, 2nd edition, 1984.

[152] H.-O. Peitgen, H. Jürgens, and D. Saupe. *Chaos and Fractals: New Frontiers of Science*. Springer-Verlag, New York, NY, USA, 1992.

[153] W. B. Pennebaker and J. L. Mitchell. *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, New York, NY, USA, 1993.

[154] A. P. Pentland. Fractal-based description of natural scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):661–674, November 1984.

[155] C. G. C. Pitts. *Introduction to Metric Spaces*. Oliver & Boyd, Edinburgh, Scotland, 1972.

[156] W. K. Pratt. Vector space formulation of two-dimensional signal processing operations. *Computer Graphics and Image Processing*, 4(3):1–24, September 1975.

[157] F. P. Preparata and M. I. Shamos. *Computational Geometry*. Texts and Monographs in Computer Science. Springer-Verlag, New York, NY, USA, 1985.

[158] M. Rabbani and P. W. Jones. *Digital Image Compression Techniques*, volume TT7 of *Tutorial texts in optical engineering*. SPIE Optical Engineering Press, Bellingham, WA, USA, 1991.

[159] B. Ramamurthi and A. Gersho. Classified vector quantization of images. *IEEE Transaction on Communications*, COM-34(11):1105–1115, 1986.

[160] E. Reusens. Overlapped adaptive partitioning for image coding based on the theory of iterated functions systems. In *Proceedings ICASSP-94 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, volume 5, pages 569–572, Adelaide, Australia, April 1994.

[161] E. Reusens. Partitioning complexity issue for iterated functions systems based image coding. In *Proceedings EUSIPCO'94 (European Signal Processing Conference)*, volume 1, pages 171–174, Edinburgh, Scotland, September 1994.

[162] R. Rinaldo and G. Calvagno. An image coding scheme using block prediction of the pyramid subband decomposition. In *Proceedings ICIP-94 (IEEE International Conference on Image Processing)*, volume II, pages 878–882, Austin, TX, USA, November 1994.

[163] R. Rinaldo and G. Calvagno. Image coding by block prediction of multiresolution subimages. *IEEE Transactions on Image Processing*, 4(7):909–920, July 1995.

[164] J. Rissanen and G. G. Langdon, Jr. Universal modeling and coding. *IEEE Transactions on Information Theory*, IT-27(1):12–23, January 1981.

[165] J. Rosiene and I. Greenshields. Standard wavelet basis compression of images. *Optical Engineering*, 33(8):2572–2578, August 1994.

[166] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, New York, NY, USA, 3rd edition, 1976.

[167] M. J. Sabin and R. M. Gray. Product code vector quantizers for waveform and voice coding. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-32(3):474–488, June 1984.

[168] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley Series in Computer Science. Addison-Wesley, Reading, MA, USA, 1989.

[169] D. Saupe. Breaking the time complexity of fractal image compression. Technical Report 53, Institut für Informatik, University of Freiburg, Freiburg, Germany, 1994.

[170] D. Saupe. Accelerating fractal image compression by multi-dimensional nearest neighbor search. In J. A. Storer and M. Cohn, editors, *Proceedings DCC'95 (IEEE Data Compression Conference)*, pages 222–231, Snowbird, UT, USA, March 1995.

[171] D. Saupe. Fractal image compression via nearest neighbor search. In *NATO ASI on Fractal Image Encoding and Analysis*, Trondheim, Norway, July 1995.

[172] D. Saupe. The futility of square isometries in fractal image compression. In *Proceedings ICIP-96 (IEEE International Conference on Image Processing)*, volume I, pages 161–164, Lausanne, Switzerland, September 1996.

[173] D. Saupe. Lean domain pools for fractal image compression. In R. L. Stevenson, A. I. Drukarev, and T. R. Gardos, editors, *Still-Image Compression II*, volume 2669 of *SPIE Proceedings*, pages 150–157, San Jose, CA, USA, January 1996.

[174] 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, September 1994.

[175] D. Saupe and H. Hartenstein. Lossless acceleration of fractal image compression by fast convolution. In *Proceedings ICIP-96 (IEEE International Conference on Image Processing)*, volume I, pages 185–188, Lausanne, Switzerland, September 1996.

[176] D. Saupe and M. Ruhl. Evolutionary fractal image compression. In *Proceedings ICIP-96 (IEEE International Conference on Image Processing)*, volume I, pages 129–132, Lausanne, Switzerland, September 1996.

[177] W. F. Schreiber. The measurement of third order probability distributions of television signals. *IRE Transactions on Information Theory*, IT-2(3):94–105, September 1956.

[178] W. F. Schreiber. *Fundamentals of Electronic Imaging Systems*. Springer series in Information Sciences. Springer-Verlag, New York, NY, USA, 1986.

[179] C. E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27(3):379–423, July 1948. Continued 27(4):623-656, October 1948.

[180] C. E. Shannon. Coding theorems for a discrete source with a fidelity criterion. In *IRE National Convention Record*, volume 7, part 4, pages 142–163, New York, NY, USA, March 1959.

[181] J. M. Shapiro. An embedded hierarchical image coder using zerotrees of wavelet coefficients. In J. A. Storer and M. Cohn, editors, *Proceedings DCC'93 (IEEE Data Compression Conference)*, pages 214–233, Snowbird, UT, USA, April 1993.

[182] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, December 1993.

[183] Y. Shoham and A. Gersho. Efficient bit allocation for an arbitrary set of quantizers. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(9):1445–1453, September 1988.

[184] J. SIGNES. Geometrical interpretation of IFS based image coding. In *NATO ASI on Fractal Image Encoding and Analysis*, Trondheim, Norway, July 1995.

[185] B. SIMON. Explicit link between local fractal transform and multiresolution transform. In *Proceedings ICIP-95 (IEEE International Conference on Image Processing)*, volume I, pages 278–281, Washington, D.C., USA, October 1995.

[186] M. R. SPEIGEL. *Mathematical Handbook of Formulas and Tables*. Schaum's Outline Series. McGraw-Hill, New York, NY, USA, 1968.

[187] M. D. SPRINGER. *The Algebra of Random Variables*. Wiley series in Probability and Mathematical Statistics. John Wiley & Sons, New York, NY, USA, 1979.

[188] E. J. STOLLNITZ, T. D. DEROSE, AND D. H. SALESIN. Wavelets for computer graphics: A primer, part 1. *IEEE Computer Graphics and Applications*, 15(3):76–84, May 1995.

[189] G. STRANG AND T. NGUYEN. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, Wellesley, MA, USA, 1996.

[190] W. SZEPANSKI. Δ-entropy and rate distortion bounds for generalised Gaussian information sources and their applications to image signals. *Electronics Letters*, 16(3):109–111, January 1980.

[191] A. H. TEWFIK AND M. KIM. Correlation structure of the discrete wavelet coefficients of fractional Brownian motion. *IEEE Transactions on Information Theory*, 38(2 (Part II)):904–909, March 1992.

[192] L. THOMAS AND F. DERAVI. Region-based fractal image compression using heuristic search. *IEEE Transactions on Image Processing*, 4(6):832–838, June 1995.

[193] A. VAN DE WALLE. Merging fractal image compression and wavelet transform methods. In *NATO ASI on Fractal Image Encoding and Analysis*, Trondheim, Norway, July 1995.

[194] A. VAN DE WALLE. Relating fractal image compression to transform methods. Master's thesis, University of Waterloo, Waterloo, Canada, 1995.

[195] R. S. VARGA. *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1962.

[196] J. VENBRUX, P.-S. YEH, G. C. ZWEIGLE, AND J. F. VESEL. A VLSI chip solution for lossless medical imagery compression. In Y. KIM, editor, *Medical Imaging 1994: Image Capture, Formatting, and Display*, volume 2164 of *SPIE Proceedings*, pages 561–572, Newport Beach, CA, USA, February 1994.

[197] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*. Prentice Hall Signal Processing Series. Prentice-Hall, Englewood Cliffs, NJ, USA, 1995.

[198] J. D. Villasenor, B. Belzer, and J. Liao. Wavelet filter evaluation for image compression. *IEEE Transactions on Image Processing*, 4(8):1053–1060, August 1995.

[199] G. Vines. Orthogonal basis IFS. In Y. Fisher, editor, *Fractal Image Compression: Theory and Application*, chapter 10, pages 199–214. Springer-Verlag, New York, NY, USA, 1995.

[200] P. H. Westerink, D. E. Boekee, J. Biemond, and J. W. Woods. Subband coding of images using vector quantization. *IEEE Transactions on Communications*, 36(6):713–719, June 1988.

[201] B. E. Wohlberg and G. de Jager. Fast image domain fractal compression by DCT domain block matching. *Electronics Letters*, 31(11):869–870, May 1995.

[202] B. E. Wohlberg and G. de Jager. Fractal coding performance for first-order Gauss-Markov models. *Electronics Letters*, 32(5):441–442, February 1996.

[203] S. J. Woolley and D. M. Monro. Rate-distortion performance of fractal transforms for image compression. *Fractals*, 2(3):395–398, 1994.

[204] S. J. Woolley and D. M. Monro. Optimum parameters for hybrid fractal image coding. In *Proceedings ICASSP-95 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, volume 4, pages 2571–2574, Detroit, MI, USA, May 1995.

[205] G. W. Wornell. A Karhunen-Loève-like expansion for $1/f$ processes via wavelets. *IEEE Transactions on Information Theory*, 36(4):859–861, July 1990.

[206] X. Wu. Globally optimal bit allocation. In J. A. Storer and M. Cohn, editors, *Proceedings DCC'93 (IEEE Data Compression Conference)*, pages 22–31, Snowbird, UT, USA, April 1993.

[207] Z. Xiong, O. Guleryuz, and M. T. Orchard. A DCT-based embedded image coder. *IEEE Signal Processing Letters*, 3(11):289–290, November 1996.

[208] Z. Xiong, K. Ramchandran, and M. T. Orchard. Space-frequency quantization for wavelet image coding. *IEEE Transactions on Image Processing*, 6(5):677–693, May 1997.

[209] R. M. Young. *An Introduction to Nonharmonic Fourier Series*. Academic Press, New York, NY, USA, 1980.

[210] P. I. ZADOR. Asymptotic quantization error of continuous signals and the quantization dimension. *IEEE Transactions on Information Theory*, IT-28(2):139–149, March 1982.

[211] G. ZORPETTE. Fractals: Not just another pretty picture. *IEEE Spectrum*, 25(10):29–31, October 1988.