ATTRACTOR IMAGE CODING WITH LOW BLOCKING EFFECTS

By

HO, Hau Lai

A Thesis

submitted in partial fulfillment of the requirements

for the Degree of Master of Philosophy

Division of Electronic Engineering

The Chinese University of Hong Kong

July 1997

To God

for his everlasting grace and glory.

To my love, Lo Tak Ching,

for her love and care.

Acknowledgement

"What has been will be again, what has been done will be done again; there is nothing new under the sun." Ecclesiastes 1:9

Thoughout my research studies in these two years, I have encountered the state-of-the-art techniques and theories in the scope of signal processing and applied mathematics that inspire me to do research in image coding using fractal methods. It is my pleasure to acknowledge and thank all people who have influenced me through my research studies. Firstly, I express by gratitude to my research supervisor Prof. Cham Wai Kuen for his support and encouragement. I thank Prof. Tsui Hung Tat and Prof. Lee Tong for being on the thesis committee.

I would like to thank all friends and colleagues in computer vision and image processing laboratory. Tse Fu Wing told me much about computers and networks. Liu Jian-zhuang taught me how to play GO in our leisure time. Au Yeung Man Ching, Liu Jian-zhuang, Wong Po Ming and Yeung Shu Yuk played tennis with me. Fan Guo-liang has inspired me in many aspects of image coding through our discussions. I would like to thank Alan Kong, A.C.O. of our department, and Chan Ding Wah, our laboratory technical staff, for the maintenance of SGI workstations I used to write my programs. I would also like to thank Dr. Geir E. Øien in Norwegian Institute of Technology for mailing me a hard copy of his PhD thesis on which my ideas build.

I am grateful to my parents for all encouragement and support they have provided me in all my years spent on educational pursuits. I would like to thank my brothers

and sisters in Joshua Fellowship for their prayers. At last but not least, I am specially thankful to my love, Tak Ching, for her love and patience through my M.Phil. programme in CUHK.

Abstract

In the present work, a relatively novel method called attractor image coding (commonly known as fractal image coding) for lossy image compression is investigated. Roughly speaking, the original image is partitioned into several disjoint range blocks. For each range block, an affine transformation is constructed to approximate the range block by another part of the same image. Compression is obtained by compactly storing only the descriptions of these transformations. The decoded image is obtained by iteratively applying these transformations on any initial image. However, the attractor coding, being a block-based algorithm, suffers from the usual blocking artifacts which is highly disturbing to human visual system (HVS). The blockiness is mainly due to the independent processing of each block in encoding. Discontinuities may occur across the block boundaries in the decoded image that are smooth in the original image. The problem is more prominent when the bit rate is reduced.

In this thesis, two variants of the existing attractor coding are proposed and examined in details. The main goal is to produce a decoded image which suffers less from blocking effects and they are summarized as follows:

First, a novel attractor coding scheme with **adjacent block parameter estimations** is proposed that exploits the redundancies of block parameters in smooth regions of an image. A criterion called δ-minimum edge difference (δ-MED) is proposed to select those blocks which can be estimated well from the adjacent blocks. The goal is to

produce a decoded image in which the blocking artifacts is not easily visible to the human vision systems (HVS). Experiments show that the proposed method can produce a decoded image with most blockiness not easily spotted out and, at the same time, achieve similar compression ability as those of variable range block size.

Since the previous proposed method still employs disjoint range blocks in the formulation, the blockiness exists in the decoded images. In this work, a novel attractor coding techniques using **the partitioned iterated function systems with lapped range blocks (LPIFS)** is proposed. Each range block is formulated to lap with its adjacent blocks through a weighting operator which is diminishing in magnitudes towards its boundaries. Range block preprocessing is proposed to compute the transformation parameters in order to reduce the error. No large system of equations as used in previous work is needed. Moreover, the local domain block matching is proposed. We show that this block matching technique is natural for general images and thus reduces the number of bits for specifying domain block addresses. Experimental results show that the image details are preserved and nearly all undesirable blocking artifacts are eliminated.

List of Special Symbols

$\mathbb{R}$	the set of real numbers
$d(x, y)$	the distance between x and y
$\|\|x\|\|$	the norm of x
$< x, y >$	the inner product of x and y
$\mathcal{I}$	a digital image of $2^n \times 2^n$ pixels
$\{\mathcal{I}_i\}$	the set of all digital image of $2^n \times 2^n$ pixels
$i_{u,v}$	the pixel of $\mathcal{I}$ at the $(u+1)$th row and the $(v+1)$th column
R, $\tilde{\mathsf{R}}$	the range pools
D	the domain pool
$\mathcal{R}_{i,j}$, $\tilde{\mathcal{R}}_{i,j}$	range blocks with i, j being the coordinates of its upper-left pixel
$\mathcal{D}_{i,j}$	a domain block with i, j being the coordinates of its upper-left pixel
$[x_{i,j}]_{0 \le i,j < N}$	a $N \times N$ matrix with $x_{i-1,j-1}$ is its element at the ith row and the jth column
$\vec{e}_{l,i,j}$, $\vec{e}_{r,i,j}$	the two difference vectors
$\mathbf{B}^B_{k,l}$	the operator to extract a $2^B \times 2^B$ block with the upper-left corner at (k, l) from an image
$\mathbf{B'}^B_{k,l}$	the operator to insert a $2^B \times 2^B$ sub-block into an all-zero image
$\mathbf{U}^B$	a $2^B \times 2^B$ square matrix with all entries being 1
$\mathbf{O}$	the orthogonalization operator
$\mathbf{E}$	the averaging operator / expectation
$\mathbf{A}$	the range block preprocessing operator
$\mathbf{W}$	the weighting operator

List of Figures

List of Tables

Contents

Chapter 1

Introduction

One of the most distinctive features of modern society is the massive amount of information being processed, transmitted, and stored by electronic means. The rapid development of internet, B-ISDN and mobile computing leads the prevalence of use of visual information. HDTV, video conferencing and digital camera are typical examples of applications with heavy demands of images. However digital images being represented in raw data format without any compression use a huge amount of storage space and consume much bandwidths in data transmission [11]. Therefore an effective representation of images plays a role of increasing importance in the present information age.

Image coding is one kind of source coding to represent an image by fewer bits by exploiting the signal redundancy. Mathematically, a good coder should represent an image by the number of bits close to its information entropy [12]. An image is represented in its most compact form and can be retrieved without any error or loss. This kind of coding scheme in which no loss is introduced is termed as lossless coding. In some applications, lossy representation is unavoidable in order to meet the bandwidth or storage requirements. Moreover, owing to the nature of human perception, some kind of image information can be removed such that the distortion induced is not

perceived by human observer. Many coding algorithms have been devised to represent an image with some information loss which cannot be perceived by human eyes. Thus, lossy coding gains many research interests in the field of image compression and can be classified into three main categories, i.e., transform coding [10], wavelet and subband coding [49], and attractor coding [27]. The first two has been well developed and satisfactory results have been reported. Among them, attractor coding is still under studies and leaves rooms for further development.

1.1 Overview of Attractor Image Coding

In the present work, a relatively novel method called attractor image coding for lossy image compression is investigated. In most publications on related methods, the name fractal coding has been used. However, several critics [38] have pointed out that fractal coding is not appropriate to describe such method. Rather than using this misleading name, some researchers have started to use attractor coding which, in the author's opinion, can well describe the nature of this method.

Attractor image compression is based on the concepts and mathematical results of iterated function systems (IFS) [2, 21, 35]. Signal redundancy is exploited by self similarity within an image [34]. Given an image, the attractor coding algorithm finds a contractive transformation on the space of images. Compression is obtained by compactly storing only the descriptions of this transformation. The initial work on such technique was originally proposed by Barnsley *et al*, who discussed the possibility of employing mathematical concepts from fractal geometry in image compression [3]. They demonstrated the power of the probabilistic IFS for compressing color images at compression ratios of over 10,000 to 1. While decoding can proceed automatically, the encoding procedure requires the human interaction in the segmentation of an image.

It was not until the publication of Jacquin's PhD thesis in 1989 that a fully automatic algorithm was invented [26, 27]. His work which based on partitioned iterated function systems (PIFS) provides the starting point for further research and extensions in many possible directions. Some of the main subjects have been addressed in great success: Fisher *et al* introduced adaptive partition in encoding to improve the image fidelity [16]. This work arose the interests to find a better partition scheme. On the other hand, Baharav *et al* proposed a fast decoding algorithm based on a hierarchical interpretation of PIFS [1]. Lundheim presented a systematic analysis of fractal coding in which blocks of an image are seen as points in a finite-dimensional inner product space [32]. Using this mathematical framework, Lepsøy and Øien generalized Jacquin's algorithm by letting the translation term be spanned by several blocks and introducing orthogonalization in translation blocks [38, 39]. By introducing orthogonalization for all blocks, the optimization in encoding was shown to be computationally less expensive. Vines, on the other hand, developed an new variant of fractal coding by an orthonormal basis approach which is a hybrid method combining principles of transform encoding with a fractal decoding [50]. Many hybrid coding methods combining attractor coding with transform coding were invented [5, 37, 46]. On the theoretical aspects, attractor coding is found to be closely related to the multiresolutional analysis and wavelets that gives an indepth understanding of the coding principles [8, 13, 35].

1.2 Scope of Thesis

Though several improvements have been made in different aspects of attractor coding, the attractor coding, being a block-based algorithm, suffers from the usual blocking artifacts which is highly disturbing to human visual system (HVS). The blockiness is mainly due to the independent processing of each block in encoding. Discontinuities may occur across the block boundaries in the decoded image that are smooth in the original image. The problem is more prominent when the bit rate is reduced.

In this thesis, several modifications on the existing fractal-based scheme are proposed and examined in details. The main goal is to produce a decoded image which suffers less from blocking effects. The theme of each subsequent chapter is summarized as follows:

Chapter 2 provides all definitions and underlying principles on which the subsequent chapters base. The fundamental theorems and mathematical background are given. The coding algorithm based on the partitioned iterated function systems (PIFS) is described. The quadtree partitioning introduced by Fisher and the orthogonalization made by Øien are introduced. Coding examples are given to illustrate the pros and cons of these coding methods.

Chapter 3 presents a novel attractor coding scheme with adjacent block parameter estimations to exploit the redundancies of block parameters in smooth regions of an image. A criterion called δ-minimum edge difference (δ-MED) is proposed to select blocks which can be estimated well from the adjacent blocks. The goal we want to achieve is to produce a decoded image in which the blocking artifacts is not easily visible to the human vision systems. Experimental results and discussions are provided.

Chapter 4 presents a novel attractor coding techniques using the partitioned iterated function systems with lapped range blocks (LPIFS). Each range block is formulated to lap with its adjacent blocks through a weighting operator which is diminishing in magnitudes towards its boundaries. Range block preprocessing is proposed to compute the transformation parameters in order to reduce the error. Unlike previous works, no large system of equations is needed. Moreover, the local domain block matching is proposed. We show that this block matching technique is natural for general images and thus reduces the number of bits for specifying domain block addresses. Experimental results show that the image details are preserved and the undesirable blocking effect is eliminated.

Chapter 5 concludes the original contributions in this thesis and raise some potential research directions based on the proposed methods.

Chapter 2

Fundamentals of Attractor Coding

This chapter presents all definitions and underlying principles on which the subsequent chapters base. The fundamental theorems and mathematical background are given. The attractor coding algorithm based on the partitioned iterated function systems (PIFS) is described. The quadtree partitioning introduced by Fisher and the orthogonalization made by Øien are introduced. Their works are the foundations on which the author lays the ideas, which will be presented from the next chapter onwards. Some coding examples are included in the last part of this chapter to illustrate the pros and cons of these existing schemes and signify the importance of the author's contributions.

2.1 Notations

In this thesis, $\mathbb{R}$ denotes the set of real numbers. $\mathcal{I}$ denotes a digital image of $2^n \times 2^n$ pixels. Whenever two or more images are involved, subscripts are added to distinguish them, e.g., $\mathcal{I}_1$ and $\mathcal{I}_2$ denotes two distinct images. The set of all digital images is written as $\{\mathcal{I}_i\}$. The pixel value at (u, v) within an image $\mathcal{I}_k$ is denoted by $i^k_{u,v}$. With the abuse of notations, an image $\mathcal{I}$ or an image sub-block can be seen as a set with

its pixels $i_{u,v}$ as its elements. Therefore the usual set notations like $i_{u,v} \in \mathcal{I}$ are well defined and understood.

All operators on an image or an image sub-block are represented by bold letters, for example the expectation operator is denoted by $\mathbf{E}$. Subscripts are appended to denote two or more distinct operators of the same functionality. A $N \times N$ matrix or an image sub-block of size $N \times N$ is sometimes written as $[x_{i,j}]_{0 \le i,j < N}$ in which $x_{i,j}$ is its element at the $(i+1)$th row and the $(j+1)$th column.

2.2 Mathematical Preliminaries

Elementary analysis is the basic tool in this thesis. In nearly all literatures on attractor coding, fundamental theorems like Banach's fixed point theorem are originally expressed in general metric space. Most of the propositions and analyses in this thesis can be stated and proved in a concise and neat way by considering an image as an element in a space. Thus the knowledge of some abstract spaces and general properties are necessary to understand the materials in this thesis. All prerequisite materials are introduced without proof in this section. Interested readers can consult books like [31, 44] on elementary analysis for the details.

An image or an image sub-block can be seen as an element in a metric space (inner product space) by properly assigning a distance function (an inner product respectively) between any two images. For an image collection $\{\mathcal{I}_i\}$, the inner product between two images $\mathcal{I}_1, \mathcal{I}_2$ is defined in the usual sense, i.e.,

$$< \mathcal{I}_1, \mathcal{I}_2 >= \frac{1}{2^n \times 2^n} \sum_{u,v} i^1_{u,v} i^2_{u,v}, \quad \forall \mathcal{I}_1, \mathcal{I}_2 \in \{\mathcal{I}_i\}. \tag{2.1}$$

$\mathcal{I}_1, \mathcal{I}_2$ is said to be orthogonal when their inner product vanishes. Metric between two elements $\mathcal{I}_1, \mathcal{I}_2 \in \{\mathcal{I}_i\}$ can be defined in terms of the norm induced by the inner product:

$$d(\mathcal{I}_1, \mathcal{I}_2) \equiv \|\mathcal{I}_1 - \mathcal{I}_2\| \equiv \sqrt{< \mathcal{I}_1 - \mathcal{I}_2, \mathcal{I}_1 - \mathcal{I}_2 >} \tag{2.2}$$

and this definition gives the root-mean-square metric in $\{\mathcal{I}_i\}$:

$$d(\mathcal{I}_1, \mathcal{I}_2) \equiv \sqrt{\frac{1}{2^n \times 2^n} \sum_{u,v} (i^1_{u,v} - i^2_{u,v})^2}, \quad \forall \mathcal{I}_1, \mathcal{I}_2 \in \{\mathcal{I}_i\}. \tag{2.3}$$

Therefore $\{\mathcal{I}_i\}$ is a metric space by equipping a metric d. Sometimes, the supremum metric is used instead whenever its use can simplify our work:

$$d(\mathcal{I}_1, \mathcal{I}_2) \equiv \max_{u,v} |i^1_{u,v} - i^2_{u,v}|, \quad \forall \mathcal{I}_1, \mathcal{I}_2 \in \{\mathcal{I}_i\}. \tag{2.4}$$

Definition 2.1 *(Contractive mapping) A mapping $\mathbf{T} : \mathcal{M} \to \mathcal{M}$ on $(\mathcal{M}, d)$ is said to be a contraction if there exists $s \in (0, 1)$ and a positive integer k s.t. $\forall \mathcal{I}_1, \mathcal{I}_2 \in \mathcal{M}$,*

$$d(\mathbf{T}^k \mathcal{I}_1, \mathbf{T}^k \mathcal{I}_2) \leq s d(\mathcal{I}_1, \mathcal{I}_2) \tag{2.5}$$

where $\mathbf{T}^k$ denotes the k-th iteration of $\mathbf{T}$. s is called the contractivity of $\mathbf{T}$. If k is equal to 1, then $\mathbf{T}$ is termed a strictly contractive transformation. If k is larger than 1, $\mathbf{T}$ is called an eventually contractive transformation.

Theorem 2.1 *(Banach's fixed point theorem) Let $(\mathcal{M}, d)$ be a complete metric space* [1] *and $\mathbf{T} : \mathcal{M} \to \mathcal{M}$ is a contraction, there exists an unique $\mathcal{I} \in \mathcal{M}$ such that $\mathbf{T}\mathcal{I} = \mathcal{I}$. Moreover, $\mathcal{I}$ can be found by*

$$\mathcal{I} = \lim_{n \to \infty} \mathbf{T}^n \mathcal{I}_0 \tag{2.6}$$

for any $\mathcal{I}_0 \in \mathcal{M}$ and $\mathcal{I}$ is called the fixed point of $\mathbf{T}$.

Banach's fixed point theorem implies that if an element is a fixed point for some contraction on a complete metric space, it can always be generated in a simple iterative manner. Thus, it is sometimes called an attractive fixed point of a contraction.

Definition 2.2 *(Collage) For any $\mathcal{I} \in \mathcal{M}$ and any transformation $\mathbf{T} : \mathcal{M} \to \mathcal{M}$, $\mathbf{T}\mathcal{I}$ is termed the collage of $\mathcal{I}$ with respect to $\mathbf{T}$.*

[1] *In mathematical analysis, a metric space is defined as complete if every Cauchy sequence is a convergent sequence. The collection of all images of $2^n \times 2^n$ pixels being a finite-dimensional vector space is trivial to satisfy this condition.*

Theorem 2.2 *(Collage approximation) Let* $\mathbf{T}$ *be a contraction on a complete metric space* $(\mathcal{M}, d)$ *with a fixed point* $\mathcal{I}$. *Let* $s_1 \in \mathbb{R}$ *such that* $\forall \mathcal{I}_1, \mathcal{I}_2 \in \mathcal{M}$, $d(\mathbf{T}\mathcal{I}_1, \mathbf{T}\mathcal{I}_2) \leq s_1 d(\mathcal{I}_1, \mathcal{I}_2)$. *Then, for any* $\mathcal{I}_0 \in \mathcal{M}$ *and any arbitrary small* $\epsilon > 0$,

$$d(\mathcal{I}_0, \mathbf{T}\mathcal{I}_0) \leq \epsilon \quad \Longrightarrow \quad d(\mathcal{I}, \mathcal{I}_0) \leq \frac{1}{1-s} \frac{1-s_1^k}{1-s_1} \epsilon. \tag{2.7}$$

In particular, if $\mathbf{T}$ is a strictly contractive transformation, the following corollary is obtained by putting $k = 1$ and $s_1 = s$.

Corollary 2.1 *Let* $\mathbf{T}$ *be a strictly contractive transformation on a complete metric space* $(\mathcal{M}, d)$ *with a fixed point* $\mathcal{I}$. *Then, for any* $\mathcal{I}_0 \in \mathcal{M}$ *and any arbitrary small* $\epsilon > 0$,

$$d(\mathcal{I}_0, \mathbf{T}\mathcal{I}_0) \leq \epsilon \quad \Longrightarrow \quad d(\mathcal{I}, \mathcal{I}_0) \leq \frac{1}{1-s} \epsilon. \tag{2.8}$$

Theorem 2.2 implies that if an element $\mathcal{I}_0$ is close to its collage $\mathbf{T}\mathcal{I}_0$ with respect to a contraction $\mathbf{T}$, then the fixed point of $\mathbf{T}$ is also close to $\mathcal{I}_0$. Hence, if an image and a contraction with a set of parameters are given, this theorem tells us that minimizing the distance between the image and its collage by choosing an appropriate set of parameters can produce an attractive fixed point close to the original image. This idea lays the theoretical foundation of attractor coding: the task of the encoder is to minimize the distance between the original image and its collage. The decoded image is the fixed point of the contraction that is very close to the original image. The question is how to define a parametric contraction $\mathbf{T}$ such that the minimization is computationally tractable and simple. In the next section, the partitioned iterated function systems (PIFS) is introduced that is the basis on which the author's contributions build. The following theorem provides the theoretical foundation of block matching in attractor coding.

Theorem 2.3 *(Subspace approximation) Let* $\mathcal{H}$ *be a complete inner product space* [2] *and* $\mathcal{H}_0$ *be its subspace. For any element* $\mathcal{I} \in \mathcal{H} \backslash \mathcal{H}_0$, *there exists an unique* $\mathcal{I}_0 \in \mathcal{H}_0$

[2] *An inner product space is complete if its induced metric space is complete. Usually, a complete inner product space is called a Hibert space.*

such that

$$d(\mathcal{I}_0, \mathcal{I}) < d(\mathcal{I}_1, \mathcal{I})$$

for all $\mathcal{I}_1 \in \mathcal{H}_0$ *and* $\mathcal{I}_1 \neq \mathcal{I}_0$. *Moreover,* $\mathcal{I}_0 - \mathcal{I}$ *is orthogonal to* $\mathcal{I}_0$.

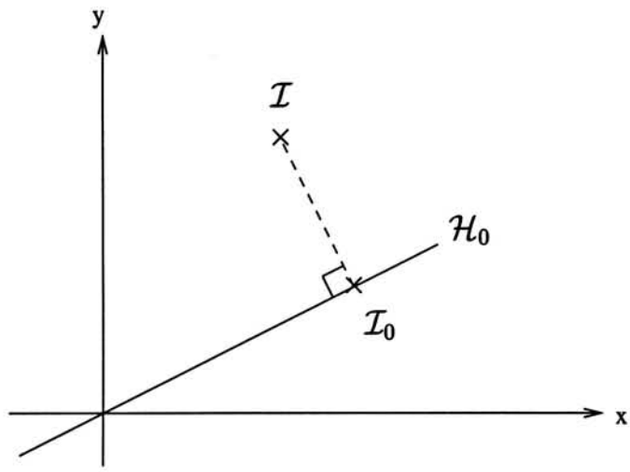


Figure 2.1: The subspace approximation of $\mathcal{I}_0$ by $\mathcal{I}$ in $\mathbb{R}^2$.

This theorem states that approximation of a given vector by the linear span of one or more vectors is minimized if the error is orthogonal to the approximation. (see Fig. 2.1) Moreover, this approximation is unique. This theorem ensures the uniqueness of the approximation and characterize the approximation error during encoding in attractor coding.

2.3 Partitioned Iterated Function Systems

The idea of attractor image coding is to formulate a contraction for an image such that its fixed point is close to the original image. (see Fig. 2.2) Theorem 2.2 in the last section provides a convenient way to find such contraction: minimizing the distance between the original image and its collage by selecting the appropriate parameters of the contraction. The decoded image can be found by applying the contraction on any initial image recursively. Eventually, the decoded image emerges as its fixed point.

However, what the structure of a contraction should be such that the minimization is computationally simple and tractable. Iterated function systems (IFS) is one kind of techniques to provide the basic structure of a contraction in fractal theory. Many variants of IFS have been devised on different metric spaces, e.g., probabilistic IFS, recurrent IFS, iterated fuzzy set systems and partitioned IFS [21, 23, 50]. In particular, partitioned iterated function systems (PIFS) gains most attention in the literatures on attractor coding techniques.

Given:

Original image $\mathcal{I}$

and

A parametric transformation **T**

Find: the appropriate parameters of T with a fixed point s.t.

Fixed point of **T**

→ **T** with appropriate parameters →

Fixed point of **T**

Figure 2.2: Fundamental idea of attractor coding.

Jacquin in his PhD thesis provided a simple and practical technique, partitioned iterated function systems (PIFS), which consists a family of affine transformations on each part of an image. Each affine transformation is constrained to apply on only one specific image block. Roughly speaking, the contraction being constructed approximates each part of an image by another part of the same image up to an affine transformation. Thus, instead of approximating the whole image at once, PIFS decouples the approximation into a set of minimization problems on parts of an image. The following section gives the mathematical and algorithmic formulation of PIFS.

2.3.1 Mathematical Formulation of the PIFS

Let $\mathcal{I}$ be the original image of $2^N \times 2^N$ pixels. The following two linear operators and an all-one matrix are defined as follows:

- Let $\mathbf{B}_{k,l}^B : \{\mathcal{I}_i\} \rightarrow \mathbb{R}^{2^B \times 2^B}$ be a linear operator that extracts a sub-block of size $2^B \times 2^B$ with the upper-left corner at (k, l) from the image $\mathcal{I}$.
- Its transpose $\mathbf{B}'^B_{k,l} : \mathbb{R}^{2^B \times 2^B} \rightarrow \{\mathcal{I}_i\}$ is an operator that inserts a $2^B \times 2^B$ sub-block into an all-zero image of size $2^N \times 2^N$ in the way that the upper-left corner of inserted sub-block at (k, l).
- Define $\mathbf{U}^B$ be a $2^B \times 2^B$ square matrix with all entries being 1.

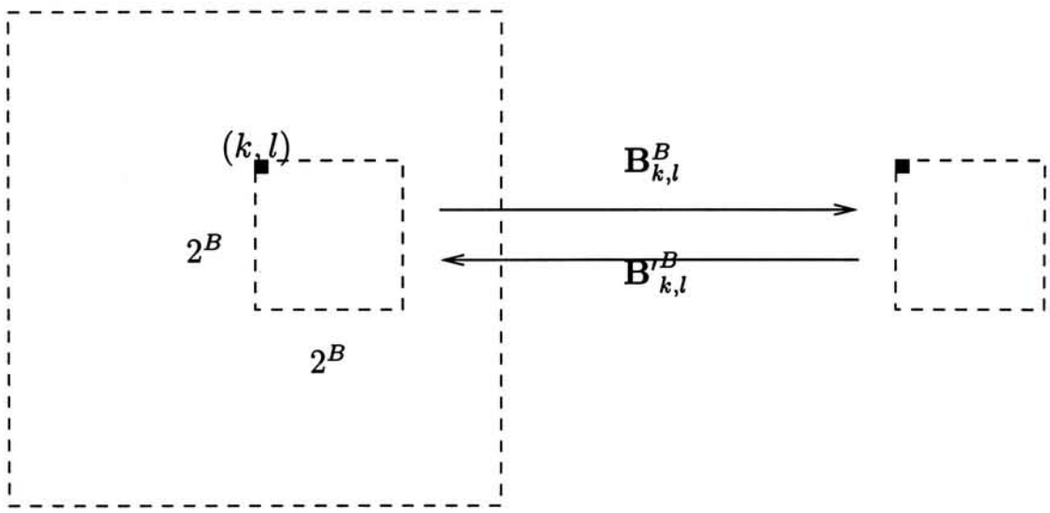


Figure 2.3: The actions of $\mathbf{B}_{k,l}^B$ and $\mathbf{B}'^B_{k,l}$.

Definition 2.3 *(Range pool) The original image $\mathcal{I}$ is partitioned into a set of non-overlapping blocks of size $2^B \times 2^B$ called range blocks* [3] *$\mathcal{R}_{i,j}$ where (i, j) denotes the pixel coordinates of the upper-left corner of the block. The collection of all these range blocks is called the range pool which is denoted by* R*:*

$$\mathsf{R} \equiv \{\mathcal{R}_{i,j} : \mathcal{R}_{i,j} \equiv \mathbf{B}_{i,j}^B \mathcal{I}_0 \text{ with } i = 2^B p, j = 2^B q, 0 \leq p, q < 2^{N-B}\}. \tag{2.9}$$

[3] *The term "target block" is used instead of range block in some literatures.*

Remark 2.1 *Using this definition of* R, *every pixel in the image* $\mathcal{I}$ *is now covered by exactly one range block* $\mathcal{R}_{i,j} \in \mathsf{R}$. *Therefore it can be easily verified that an image* $\mathcal{I}$ *of* $2^N \times 2^N$ *pixels can be written as the summation of all non-overlapping range block of size* $2^B \times 2^B$, *i.e.,*

$$\mathcal{I} = \sum_{k=0}^{2^{N-B}-1} \sum_{l=0}^{2^{N-B}-1} \mathbf{B'}^{B}_{2^B k, 2^B l} \mathbf{B}^{B}_{2^B k, 2^B l} \mathcal{I}.$$

Definition 2.4 *(Domain pool)*[4] *The original image* $\mathcal{I}$ *is partitioned into a number of square blocks (possibly overlapping) of size* $2^D \times 2^D$ *called domain blocks* $\mathcal{D}_{i,j}$ *where* (i, j) *denotes the pixel coordinates of the upper-left corner of the block.* D *should be chosen to be larger than* B. *The collection of all domain blocks is called the domain pool and is denoted as* D.

Remark 2.2 D *may be designed to contain some "artificial" blocks not extracted from the image. These blocks, of course, are predefined and made known to both encoder and decoder. In this case, the vector quantization (VQ) is used in the attractor coding method [14, 18].*

Remark 2.3 *There is no specific rule to govern the choice of the size and the position of a domain pool within an image. The most common choice of* D *is* $D = B + 1$ *and so a domain block size is four times of that of a range block. The common choice of* D *is the one in which all square blocks with half of their support overlapped:*

$$\mathsf{D} \equiv \{\mathcal{D}_{i,j} : \mathcal{D}_{i,j} \equiv \mathbf{B}^{D}_{i,j} \mathcal{I}_0 \text{ with } i = 2^{D-1}p, j = 2^{D-1}q, 0 \leq p, q < 2^{N-D}\}. \tag{2.10}$$

The choice of the domain pool size affects the performance of the coding scheme. In general, a larger domain pool gives a better performance in the cost of using more bits to represent the locations of domain blocks.

[4] *Domain pool is called library in some literatures.*

After both range block collection and the library are defined, an affine transformation is defined on each $\mathcal{R}_{i,j} \in \mathsf{R}$ mapping a domain block to each range block. The action of this affine transformation can be seen to consist of three parts: the first part is to pick up an appropriate domain block from the domain pool, the second part is to transform the selected domain block, and the last part places the resultant block at the range block location. (see Fig. 2.4) Mathematically, for each $\mathcal{R}_{i,j} \in \mathsf{R}$ an affine transformation is constructed in the following way:

1. Define $\mathbf{P}_{i,j} : \{\mathcal{I}_i\} \rightarrow \mathbb{R}^{2^D \times 2^D}$ as a operator which gets a domain block $\mathcal{D}_{k,l} \in \mathsf{D}$ from $\mathcal{I}$, i.e., $\mathbf{P}_{i,j} \equiv \mathbf{B}^D_{k,l}$.

2. Define $\mathbf{G}_{i,j} : \mathbb{R}^{2^D \times 2^D} \rightarrow \mathbb{R}^{2^B \times 2^B}$ be an affine transformation which maps a domain block $\mathcal{D}_{k,l}$ to the range block location $\mathcal{R}_{i,j}$ up to an affine transformation. $\mathbf{G}_{i,j}$ consists of the following parts:

   - A decimation operator $\mathbf{D} : \mathbb{R}^{2^D \times 2^D} \rightarrow \mathbb{R}^{2^B \times 2^B}$ which reduces the domain block size from $2^D \times 2^D$ to $2^B \times 2^B$ by first dividing the $2^D \times 2^D$ block into $\frac{D}{B} \times \frac{D}{B}$ number of non-overlapping $2^{D-B} \times 2^{D-B}$ sub-blocks and then taking the average value of each sub-block.
   - A isometrical operator $\mathbf{I}_{i,j} : \mathbb{R}^{2^B \times 2^B} \rightarrow \mathbb{R}^{2^B \times 2^B}$ which transforms the given block by the combination of 90 degree rotations and reflections. There are altogether eight isometries of a square block. [5]
   - Scalar multiplication of the given block by the scaling coefficient $s_{i,j} \in \mathbb{R}$ and an addition of a block $\mathbf{U}^B$ scaled by the offset coefficient $o_{i,j} \in \mathbb{R}$.

   Thus $\mathbf{G}_{i,j}$ can be written as

$$\mathbf{G}_{i,j}(\mathcal{D}_{k,l}) = s_{i,j} \times \mathbf{DI}_{i,j}(\mathcal{D}_{k,l}) + o_{i,j} \times \mathbf{U}^B. \tag{2.11}$$

[5] An isometrical operator $\mathbf{I}_{i,j} : \mathcal{X} \rightarrow \mathcal{X}$ on some normed space $\mathcal{X}$ is by definition a norm-preserving operator, i.e., $\|\mathbf{I}_{i,j}x\| = \|x\|$ for all $x \in \mathcal{X}$. In this application, only eight isometries are used: the identity operator, three operators by rotations of 90°, 180° and 270°, and four operators by reflections along both diagonals and along the vertical and horizontal center axes respectively. More isometries formulated by conformal mappings of a square block are also investigated by some researchers [41].

3. Define $\mathbf{B'}^{B}_{i,j} : \mathbb{R}^{2^B \times 2^B} \rightarrow \{\mathcal{I}_i\}$ which puts the resultant block at the range block location.

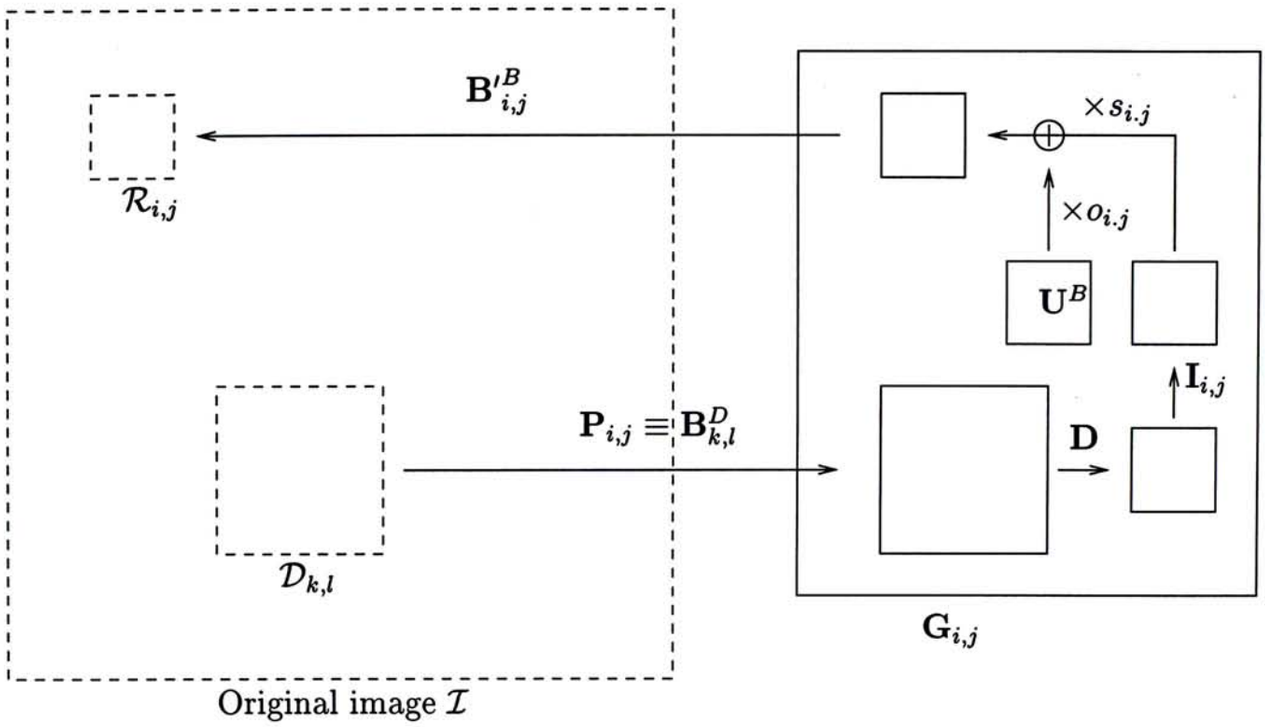


Figure 2.4: Illustrations of an affine transformation on a range block $\mathcal{R}_{i,j}$ in the PIFS.

Having defined the affine transformation for each $\mathcal{R}_{i,j} \in \mathsf{R}$, we define the overall transformation on the image space is defined as follows:

Definition 2.5 *(Overall transformation) The overall transformation* $\mathbf{T} : \{\mathcal{I}_i\} \rightarrow \{\mathcal{I}_i\}$ *is defined by summing up all affine transformations of each* $\mathcal{R}_{i,j} \in \mathsf{R}$*:*

$$\mathbf{T}\mathcal{I} \equiv \sum_{\mathcal{R}_{i,j} \in \mathsf{R}} \mathbf{B'}^{B}_{i,j} \mathbf{G}_{i,j} \mathbf{P}_{i,j}(\mathcal{I}), \quad \forall \mathcal{I} \in \{\mathcal{I}_i\}. \tag{2.12}$$

$\mathbf{T}$ is a transformation based on the PIFS. $\mathbf{T}$ consists of the following parameters of each range block $\mathcal{R}_{i,j} \in \mathsf{R}$:

1. The location of the domain block chosen, $\mathbf{P}_{i,j}$.
2. The isometry used, $\mathbf{I}_{i,j}$.
3. The scaling coefficient $s_{i,j}$.

4. The offset coefficient $o_{i,j}$.

These parameters completely specify the transformation $\mathbf{T}$ and hence the fixed point of this transformation is determined by these parameters. By substituting (2.11) into (2.12), we get

$$\mathbf{T}\mathcal{I} \equiv \underbrace{\left(\sum_{\mathcal{R}_{i,j} \in \mathsf{R}} \mathbf{B'}^{B}_{i,j}(s_{i,j} \times \mathbf{DI}_{i,j}\mathbf{P}_{i,j}) \right) \mathcal{I}}_{\text{linear term}} + \underbrace{\sum_{\mathcal{R}_{i,j} \in \mathsf{R}} \mathbf{B'}^{B}_{i,j}(o_{i,j} \times \mathbf{U}^{B})}_{\text{constant term}} . \tag{2.13}$$

This transformation $\mathbf{T}$ is called an affine transformation as it is composed of a linear term and a constant term. Up to this stage, the contractivity of the transformation $\mathbf{T}$ is not discussed. $\mathbf{T}$ still has the possibility to be a non-contractive transformation. Theorem 2.4 gives the condition for $\mathbf{T}$ to be a contraction.

Theorem 2.4 *(Contractivity of* $\mathbf{T}$*)* $\mathbf{T} : \{\mathcal{I}_i\} \rightarrow \{\mathcal{I}_i\}$ *is a strictly contractive transformation respect to the supremum metric if all scaling coefficients have absolute value less than* 1*, i.e.,* $|s_{i,j}| < 1$ *for all* $\mathcal{R}_{i,j} \in \mathsf{R}$.

Remark 2.4 *However, it is only a sufficient condition to secure contractivity.* $\mathbf{T}$ *may still be a contraction even some* $s_{i,j}$ *have magnitude greater than* 1 *[38].*

2.4 Attractor Coding using the PIFS

In the encoding procedure, the goal is to minimizing the distance between the original image and its collage. This is equivalent to finding the best parameters for every range block $\mathcal{R}_{i,j} \in \mathsf{R}$ such that the chosen domain block after the affine transformation is closest to $\mathcal{R}_{i,j}$, i.e.,

$$\min_{\mathbf{P}_{i,j}, \mathbf{I}_{i,j}, s_{i,j}, o_{i,j}} d(\, \mathcal{R}_{i,j},\ \mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I})\,) \quad \text{for each} \quad \mathcal{R}_{i,j} \in \mathsf{R}. \tag{2.14}$$

This minimization problem is termed as collage minimization or collage optimization. $\mathbf{P}_{i,j}$ and $\mathbf{I}_{i,j}$ of each $\mathcal{R}_{i,j}$ can be exactly specified in binary format: the location is

represented by the horizontal and vertical coordinates in an image and each of the eight isometry is represented by $3(= \log_2 8)$ bits. However, the scaling and offset parameters are in general real numbers and quantization must be involved if they are stored in digital form.

The collage minimization is carried out by exhaustive searching the domain pool for a suitable domain block and then finding the corresponding parameters, $s_{i,j}$ and $o_{i,j}$, and an isometry to minimize the distance. The domain block with its corresponding parameters and isometry that gives the minimum distance is stored. If the root-mean-square metric is used in (2.14), given a domain block with a certain isometry, both parameters, $s_{i,j}$ and $o_{i,j}$, can be calculated explicitly by setting their corresponding partial derivatives of (2.14) to zero, i.e.,

$$\frac{\partial}{\partial s_{i,j}} d(\ \mathcal{R}_{i,j},\ \mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I})\ ) = \frac{\partial}{\partial s_{i,j}} \sqrt{\sum_{u,v} (i^{\mathcal{R}}_{u,v} - s_{i,j} \times i^{\mathcal{D}}_{u,v} - o_{i,j})^2} = 0, \tag{2.15}$$

$$\frac{\partial}{\partial o_{i,j}} d(\ \mathcal{R}_{i,j},\ \mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I})\ ) = \frac{\partial}{\partial o_{i,j}} \sqrt{\sum_{u,v} (i^{\mathcal{R}}_{u,v} - s_{i,j} \times i^{\mathcal{D}}_{u,v} - o_{i,j})^2} = 0 \tag{2.16}$$

where $i^{\mathcal{R}}_{u,v} \in \mathcal{R}_{i,j}$ and $i^{\mathcal{D}}_{u,v} \in \mathbf{DI}_{i,j}\mathbf{P}_{i,j}(\mathcal{I})$ are the pixels in the range blocks $\mathcal{R}_{i,j}$ and the decimated domain block $\mathbf{DI}_{i,j}\mathbf{P}_{i,j}(\mathcal{I})$ respectively and they are both ordered lexicographically in summation. By solving these simultaneous equations in $s_{i,j}$ and $o_{i,j}$, we can obtain the explicit expression for the optimal parameters $s_{i,j}$ and $o_{i,j}$ as follows:

$$s_{i,j} = \frac{2^{2B} \sum_{u,v} i^{\mathcal{R}}_{u,v} i^{\mathcal{D}}_{u,v} - (\sum_{u,v} i^{\mathcal{R}}_{u,v})^2}{(\sum_{u,v} i^{\mathcal{D}}_{u,v})^2 - 2^{2B} \sum_{u,v} (i^{\mathcal{D}}_{u,v})^2}, \tag{2.17}$$

$$o_{i,j} = \frac{\sum_{u,v} i^{\mathcal{R}}_{u,v} - s_{i,j} \sum_{u,v} i^{\mathcal{D}}_{u,v}}{2^B \times 2^B}. \tag{2.18}$$

These parameters ($\mathbf{P}_{i,j}, \mathbf{I}_{i,j}, s_{i,j}$ and $o_{i,j}$) constitute a representation of $\mathcal{I}$. Compression is achieved by storing only the descriptions of the transformation $\mathbf{T}$. In the decoding procedure, the quantized parameters are used to construct the contractive transformation $\mathbf{T}$. This contractive transformation $\mathbf{T}$ is performed recursively on any initial

image $\mathcal{I}_0$. The simplest and usual choice of initial image is the image with all pixel values equal zero. The decoded image emerges as the fixed point of $\mathbf{T}$ after several iterations.

2.4.1 Quadtree Partitioning

In the definition of range pool, all range blocks are of the same size. Since an image is in general a non-stationary signal, there are some regions of an image that are difficult to cover well by just one range block, i.e., the minimized distance between the original image and its collage on that region is still large. On the other hand, there are some low-activity regions that can be covered well even using a larger range block. In order to take into account of the varying local activities, different range block sizes and even other block geometries are allowed to ensure acceptable decoded image quality. Quadtree [16] and HV [17] partitioning are two typical examples using rectangles of different sizes as range blocks. Triangular partitioning, on the other hand, uses triangles of irregular sizes and orientations [14]. Among them, quadtree partitioning is the most common and easy way in implementation.

For a quadtree partition with predefined I levels, the initial range pool contains blocks of the same size, say $2^B \times 2^B$, at level 0. Initially the parameter i which serves as the level counter is set to be zero. The encoder starts to approximate each range block of level i (of size $2^{B-i} \times 2^{B-i}$) in the range pool. On each level i, blocks that cannot be approximated well are splitted into four non-overlapping sub-blocks of level $i+1$ (of size $2^{B-i-1} \times 2^{B-i-1}$). This splitting process is repeated for each block until an acceptable approximation is found or the predefined level I is reached.

2.4.2 Inclusion of an Orthogonalization Operator

As stated before, the transformation $\mathbf{T}$ must be contractive in order to find the decoded image by iterative method. One approach to secure contractivity is to constrain all magnitudes of scaling coefficients strictly less than one. If this constraint is relaxed, there is at least a theoretical possibility that the decoding algorithm may fail, i.e., divergence occurs. Some researchers showed that this constraint on the scaling coefficients often degrades the coding performance, both in error measure and visual quality [38, 16]. Based on these observations, a simple modification on the structure of the transformation $\mathbf{T}$ was proposed by Øien. An orthogonalization operator is included in the linear term of the affine transformation [38, 39]. Øien showed that constraint on scaling coefficients is no longer necessary to secure convergence. The decoding algorithm becomes exact convergence in a finite, image-independent number of iterations. The only modification is the inclusion of an orthogonalization operator in the affine transformation in (2.11):

Definition 2.6 *(Orthogonalization operator)* $\mathbf{O} : \mathbb{R}^{2^B \times 2^B} \to \mathbb{R}^{2^B \times 2^B}$ *is defined to remove the DC component of the given block of size* $2^B \times 2^B$*:*

$$\mathbf{O}(\mathbf{B}_{i,j}^{B}\mathcal{I}) = \mathbf{B}_{i,j}^{B}\mathcal{I} - \mathbf{E}(\mathbf{B}_{i,j}^{B}\mathcal{I}) \times \mathbf{U}^{B} \tag{2.19}$$

where $\mathbf{B}_{i,j}^{B}\mathcal{I}$ *is an arbitrary* $2^B \times 2^B$ *block and* $\mathbf{E} : \mathbb{R}^{2^B \times 2^B} \to \mathbb{R}$ *is the averaging operator of a given block.*

Then the affine transformation on each range block is alternatively defined as (see Fig. 2.5):

$$\mathbf{G}_{i,j}(\mathcal{D}_{k,l}) = s_{i,j} \times \mathbf{ODI}_{i,j}(\mathcal{D}_{k,l}) + o_{i,j} \times \mathbf{U}^{B}. \tag{2.20}$$

Overall transformation $\mathbf{T}$ is still given by (2.12) using $\mathbf{G}_{i,j}$ defined by (2.20), i.e.,

$$\mathbf{T}\mathcal{I} \equiv \left(\sum_{\mathcal{R}_{i,j} \in \mathsf{R}} \mathbf{B'}_{i,j}^{B}(s_{i,j} \times \mathbf{ODI}_{i,j}\mathbf{P}_{i,j}) \right) \mathcal{I} \quad + \quad \sum_{\mathcal{R}_{i,j} \in \mathsf{R}} \mathbf{B'}_{i,j}^{B}(o_{i,j} \times \mathbf{U}^{B}). \tag{2.21}$$

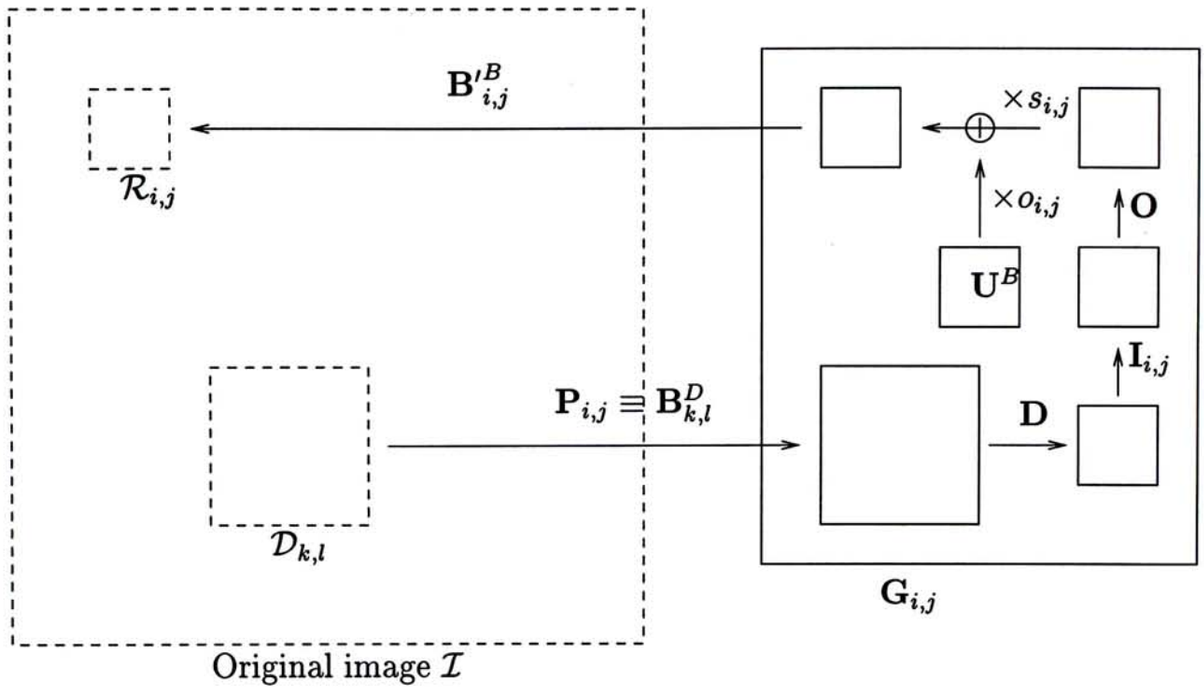


Figure 2.5: Illustrations of the orthogonalized affine transformation on a range block $\mathcal{R}_{i,j}$ in the orthogonalized PIFS.

We term such $\mathbf{T}$ as the orthogonalized PIFS. Then, the optimal scaling and offset parameters can be obtained with the following equations:

$$s_{i,j} = \frac{2^{2B}\sum_{u,v} i^{\mathcal{R}}_{u,v} i^{\mathcal{D}}_{u,v} - (\sum_{u,v} i^{\mathcal{R}}_{u,v})^2}{(\sum_{u,v} i^{\mathcal{D}}_{u,v})^2 - 2^{2B}\sum_{u,v}(i^{\mathcal{D}}_{u,v})^2}, \tag{2.22}$$

$$o_{i,j} = \frac{\sum_{u,v} i^{\mathcal{R}}_{u,v}}{2^B \times 2^B}. \tag{2.23}$$

where $i^{\mathcal{R}}_{u,v} \in \mathcal{R}_{i,j}$ and $i^{\mathcal{D}}_{u,v} \in \mathbf{DI}_{i,j}\mathbf{P}_{i,j}(\mathcal{I})$ are the pixels in the range block $\mathcal{R}_{i,j}$ and the decimated domain block $\mathbf{DI}_{i,j}\mathbf{P}_{i,j}(\mathcal{I})$ respectively and they are both ordered lexicographically in summation. The following theorem summaries the important facts about the orthogonalized PIFS:

Theorem 2.5 *(Orthogonalized PIFS) With the use of an orthogonalization operator and the domain pool defined in (2.10), the transformation* $\mathbf{T}$ *is shown to have the following three properties:*

- $\mathbf{T}$ *is an eventually contractive affine transformation.*

- *There is no need to constrain the magnitudes of the scaling coefficients to secure contractivity, i.e.,* $\mathbf{T}$ *is a contraction unconditionally.*
- *The decoded image can be obtained in a finite, image-independent number of iterations in the decoder. Moreover, the number of iterations depends only on the sizes of the range and domain blocks only:*

$$\log_{2^{D-B}} 2^B = \frac{B}{D-B}.$$

Remark 2.5 *Since there is no constraint on the magnitudes of the scaling coefficients, the overall coding performance would not be degraded owing to the magnitude constraint on scaling coefficients.*

Remark 2.6 *If the usual choices of* $B = 3$ *and* $D = 4$ *are used, the number of iterations in the decoding of the orthogonalized PIFS is* 3 *only. Conventional PIFS usually requires more iterations for most images and the number of iterations is image dependent.*

2.5 Coding Examples

In this section, experiments are conducted on digital images to show how the attractor coding introduced in the last section actually works. The basic attractor coding algorithm using PIFS with constant range block size are experimented. Moreover, the effects of using quadtree partitioning and orthogonalization are demonstrated. After going through this section, the readers should fully understand the principles of PIFS and see the pros and cons of these existing schemes.

2.5.1 Evaluation Criterion

Our quantatitive measure of the image quality of image $\mathcal{I}_2$ which is a reconstruction of the original image $\mathcal{I}_1$ is the peak-to-peak signal-to-noise ratio (PSNR) is defined as

$$\text{PSNR} = 10 \log_{10} \left[\frac{255^2}{\frac{1}{2^n \times 2^n} \sum_{u,v} (i^1_{u,v} - i^2_{u,v})^2} \right] \text{ dB.} \tag{2.24}$$

The images are of size $2^n \times 2^n$ and the gray level of each pixel ranges from 0 to 255. Sometimes an image with high PSNR does not give a perceptually desired images. It is due to the frequency selectivity and masking effect in human vision system which cannot be not accurately reflected in PSNR. For example, blocking effect usually cannot be reflected in PSNR but is visibly annoying to human vision system. Thus both image visual quality and PSNR are employed to give fair judgment of the performance.

2.5.2 Experimental Settings

Two images are used here as our original images. Each of these images is partitioned into non-overlapping range blocks. Minimization of each range block is carried by exhaustively searching all domain blocks in the library and optimizing the scaling and offset coefficients with respect to the root-mean-square metric. The following settings are used:

- The range pool containing range block of constant size 8×8 ($B = 3$) is used. Thus, there are altogether 4096 blocks in R for an image of 512×512 pixels.
- The domain pool contains blocks of size 16×16, i.e., $D = B + 1$, that implies each domain block contains four times number of pixels than a range block. Therefore the decimation operator $\mathbf{D}$ averages the four neighboring pixels in the domain block and downsamples by 2 in both vertical and horizontal directions.
- All 8 isometries of a square block are used and 3 bits ($= \log_2 8$) are required to represent them.

- Both scaling and offset parameters are real numbers which require quantization in any compression scheme. The magnitudes of scaling parameters are constrained to be less than 1. The range of offset parameters is from -127 to 127 for an image of 256 grey levels. Both parameters are uniformly quantized in their respective range and represented by 5 bits and 7 bits respectively which, in Fisher's experiments [18], gives the reasonable results in terms of decoded image quality and bit rate required.

The conventional PIFS coding scheme is implemented using the above settings and the affine transformation defined in (2.11) on each range block. The effect of the inclusion of orthogonalization operator is investigated by replacing (2.11) by (2.20) and is termed as the orthogonalized PIFS coding method. Since there is no constraint on the magnitude of the scaling parameters, the range of scaling parameters can be extended to the whole real line. In practice, the range is re-defined to be $(-2, 2)$ that is sufficient to include nearly all scaling parameters without severe truncation on the largest. Moreover, the quadtree partitioning is demonstrated by just allowing different sizes of range blocks and domain blocks. [6]

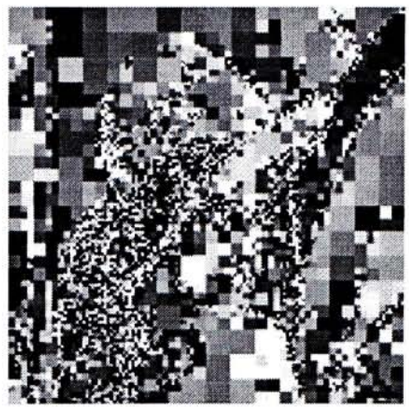
In the decoding process, all parameters on each range block are retrieved to reconstruct the contraction $\mathbf{T}$. This contraction is applied on any initial image recursively. The decoded image is the fixed point of the contraction. Fig. 2.6 shows the typical decoded process of the image Lena.

2.5.3 Results and Discussions

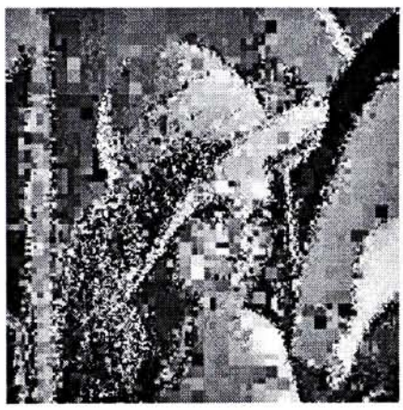
All three methods are tuned to give similar bit rates for the same image. The numerical results from this experiment are given in table 2.1 and the decoded image, Lena, of three methods are plotted in Fig. 2.7. The performance are in many ways encouraging

[6]The coding results using quadtree partition are generated by Fisher's sample code using its default settings. The program source code and user manual can be freely downloaded from `http://inls.ucsd.edu/y/Fractals/frac_comp.tar.Z`.

Initial black image



After one iteration



After two iterations

After ten iterations

Figure 2.6: Illustrations of the iterative decoding process in attractor coding using Lena as the original image and an all-zero image as the initial image.

but several drawbacks are found. Some points are noted in these experiments:

Observation 1: Visible blocking artifacts are observed in the range block boundaries in all three decoded images. They are particularly visible on the smooth regions like shoulder of Lena owing to the poor masking effect of our HVS. In the case of using quadtree partition, the blockiness is more severe as most smooth regions are now covered by larger range blocks. Though the PSNR is smaller for the decoded image obtained by quadtree partition, the discontinuities across the block boundaries are more obvious in these regions. Based on this observation, an adjacent block parameter estimation is proposed in the next chapter to use the constant range block size but result in lower PSNR comparable to adaptive partition.

Table 2.1: Performance of three attractor image coding methods

Image	Conventional PIFS		Orthogonalized PIFS		PIFS with quadtree partition	
	bpp	PSNR	bpp	PSNR	bpp	PSNR
Lena	0.42	30.99	0.42	31.66	0.40	30.95
Fruits	0.42	31.74	0.42	31.75	0.37	31.00

Observation 2: Some image details like the hat texture and eye regions are lost and grainy textures are smoother in the decoded images. These losses can be explained by the fact that the root-mean-square metric in the collage minimization does not specially take the high frequency components into account. Moreover, the quadtree partition preserves more image details as it assigns smaller blocks to those high activity regions.

Observation 3: The orthogonalized PIFS gives better results than the conventional PIFS in general. The performance improvement is mainly due to the unconstrained magnitudes of scaling parameters. Constraint on the magnitudes of scaling coefficients can reduce the chance of finding a good match for the range blocks.

2.6 Summary

In this chapter, all basic definitions and theoretical backgrounds of attractor coding are introduced. The coding method based on the partitioned iterated function systems (PIFS) is clearly defined. The quadtree partition and the inclusion of the orthogonalization operator are presented. These coding methods are experimented on digital images in order to reveal their pros and cons. Blockiness and loss of details are observed in the decoded images. In the following chapters, some novel methods are proposed and analysed based on the observations of the drawbacks of the existing methods.

(a) Original image Lena

(b) PIFS with quadtree partitioning: 0.40 bpp, PSNR=30.95dB

(c) Conventional PIFS: 0.42 bpp, PSNR=30.99dB

(d) Orthogonalized PIFS: 0.42 bpp, PSNR=31.66dB

Figure 2.7: Experimental results of image Lena.

Chapter 3

Attractor Coding with Adjacent Block Parameter Estimations

In the last chapter we have seen that attractor coding suffers from the visible blocking artifacts as the usual block-based coding schemes do. Blocking artifacts appear as the results of the independent optimization of each of the non-overlapping range blocks and so undesirable discontinuities across block boundaries may occur in the decoded image. This problem is quite difficult to deal with as it is not totally reflected by just observing the root-mean-square error. The root-mean-square minimization may produce a decoded image which is very close to the original but full of blockiness. Root-mean-square metric is still commonly employed as it is quite difficult to find another "objective" measure which can reflect the degree of blockiness very well. Therefore instead of eliminating the blocking artifacts in the encoder, some researchers proposed some post-processing techniques to alleviate the blocking artifacts [9, 36, 40]. Their methods, relying heavily on the smoothness properties on most parts of images, average the pixel values on the block boundaries by some low-pass filters. These methods can produce a decoded image suffering less from blockiness but their approach is a remedy rather than a precaution: the blockiness is minimized in decoding after it is introduced

in the encoding procedure. In this chapter we propose a scheme which hides the blocking artifacts in encoding in the hope that the blockiness though still exists in the decoded image is not easily visible to the human vision systems.

For most images, large regions are of low activity or smooth in nature. The statistical properties may be stationary on the low activity region covering quite a number of pixels. If the constant range block size is used, the best matching domain blocks of the range blocks in smooth regions are similar. Thus the parameters of the adjacent blocks in a smooth region contain redundancies. Adaptive partitions (i.e., variable range block size) like quad-tree partition and HV partition were proposed to exploit these redundancies. Higher compression ratio is achieved by assigning large block sizes to smooth portions and small block sizes to high activity regions of an image. However blocking effects appear at the boundaries of large blocks and they are particularly visibly annoying owing to the poor masking effect of HVS on smooth regions. This problem is quite serious in many adaptive partitioning scheme. However, it is not very serious when the constant range block size is used in the cost of representing more blocks in the smooth regions.

In this work, we propose an attractor coding scheme with constant range block size that exploits the redundancies of block parameters in smooth regions by adjacent block parameter estimation [24]. The criterion to select the blocks that can be estimated well from the adjacent blocks is the δ-minimum edge difference (δ-MED) which is a variant of the minimum edge difference (MED) model [7, 48]. The MED model was experimented in the framework of transform coding to estimate the DC coefficients. It is found that MED can successfully estimate about 80% of DC coefficients for most images in the DCT-based JPEG scheme. We found that the δ-MED criterion is an effective tool to determine if a block and its adjacent blocks are similar.

Roughly speaking, the proposed estimation scheme processes each block of the image from left to right and from top to bottom in the conventional way. The major

modification is the inclusion of an adjacent block estimation before any domain block matching procedure. The current range block is first examined by the δ-MED criterion to see whether it can share some information with its adjacent block(s). In other words, the redundancies in coding are exploited by sharing the common information in these blocks.

In the following sections, the δ-MED criterion is introduced first followed by the details of the proposed estimation scheme. In the last section, results and discussions of the proposed methods are presented.

3.1 δ-Minimum Edge Difference

Generally, most parts of an image are locally stationary and so adjacent blocks usually have similar characteristics like smoothness, textures and orientations of discontinuity. Thus it is very likely for these blocks to share the same domain blocks. Redundancies can be removed by using just one domain block for all these range blocks instead of using two or more domain blocks. In order to find out whether the parameters of the current block can share with its adjacent blocks, the δ-MED condition is used to identify these blocks out of the range pool.

3.1.1 Definition

Assuming that the attractor coding scheme with constant range block size is used. All range blocks are of size $2^B \times 2^B$. Let $\mathcal{R}_{i,j} = \mathbf{B}^B_{i,j}\mathcal{I}$ be an arbitrary range block taken from the range pool. This range block has $\mathcal{R}_{i+2^B,j} = \mathbf{B}^B_{i+2^B,j}\mathcal{I}$ and $\mathcal{R}_{i,j+2^B} = \mathbf{B}^B_{i,j+2^B}\mathcal{I}$ as its lower and right adjacent range blocks respectively. $i^{i,j}_{u,v} \in \mathcal{R}_{i,j}$ is defined as the (u,v)th pixel of the block $\mathcal{R}_{i,j}$. Similarly, $i^{i+2^B,j}_{u,v} \in \mathcal{R}_{i+2^B,j}$ and $i^{i,j+2^B}_{u,v} \in \mathcal{R}_{i,j+2^B}$ are the (u,v)th pixels of the block $\mathcal{R}_{i+2^B,j}$ and $\mathcal{R}_{i,j+2^B}$ respectively. Define two difference

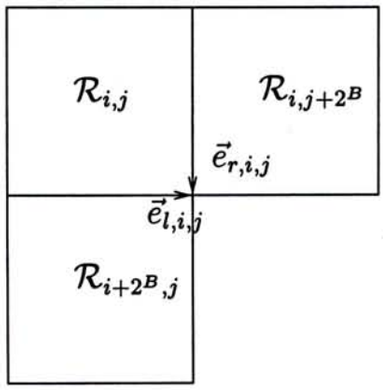


Figure 3.1: δ-MED Criterion

column vectors $\vec{e}_{l,i,j}, \vec{e}_{r,i,j} \in \mathbb{R}^{2^B}$ as follows: (see Fig. 3.1)

$$\vec{e}_{l,i,j} = \begin{bmatrix} i^{i,j}_{2^B-1,0} - i^{i,j+2^B}_{0,0} \\ i^{i,j}_{2^B-1,1} - i^{i,j+2^B}_{0,1} \\ \vdots \\ i^{i,j}_{2^B-1,2^B-1} - i^{i,j+2^B}_{0,2^B-1} \end{bmatrix}$$

and

$$\vec{e}_{r,i,j} = \begin{bmatrix} i^{i,j}_{0,2^B-1} - i^{i+2^B,j}_{0,0} \\ i^{i,j}_{1,2^B-1} - i^{i+2^B,j}_{1,0} \\ \vdots \\ i^{i,j}_{2^B-1,2^B-1} - i^{i+2^B,j}_{2^B-1,0} \end{bmatrix}.$$

Definition 3.1 *(δ-MED criterion)* [1] *The δ-MED criterion is said to be satisfied if the following inequality holds:*

$$\min\{\|\vec{e}_{l,i,j}\|, \|\vec{e}_{r,i,j}\|\} < \delta \tag{3.1}$$

where $\delta \in \mathbb{R}$ is a predefined threshold.

We found that if the δ-MED condition is satisfied, then the two adjacent blocks are most likely of low activity and having similar transformation parameters. This criterion requires minimal computation as the calculation is restricted on the boundary pixels of the blocks only.

[1] *δ-MED criterion is termed as MED criterion in [24].*

3.1.2 Theoretical Analysis

In order to understand how and why the proposed scheme works, we analyze the δ-MED criterion using the first-order Gauss-Markov image model introduced here. The usefulness of δ-MED is fully demonstrated from this analysis which provides the theoretical principals on which the adjacent block parameter estimation bases.

An image model is important in developing new algorithms in image processing as it is impossible to test all images by the proposed algorithms. In our work, the first-order Gauss-Markov model with correlation coefficient ρ is used for this purpose [28]. For a vector $\vec{X}$ of length N, the autocorrelation matrix $\mathbf{E}(\vec{X}\vec{X}^T)$ of a Gauss-Markov process is given by

$$\mathbf{E}(\vec{X}\vec{X}^T) = \begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{N-1} \\ \rho & 1 & \rho & \dots & \rho^{N-2} \\ \vdots & & \ddots & \ddots & \vdots \\ \rho^{N-2} & \dots & & 1 & \rho \\ \rho^{N-1} & \dots & & \rho & 1 \end{bmatrix} \tag{3.2}$$

where $\mathbf{E}$ is the expectation operator. The correlation coefficient ρ is usually very close to one. It should be noted that this is a generally accepted model for local behavior only. It is not global as image is in general a non-stationary signal.

Based on this model, we start to establish the theoretical foundation of the δ-MED criterion. The fundamental question we concern is whether this criterion can effectively find out those blocks of low activity or similar characteristic. If the criterion can indeed serve this purpose, we want to know whether these blocks can be covered well by the same domain block. These two questions are in general hard to answer as we do not know every characteristic of all images. However if we assume that most parts of an image can be well modeled by the first-order Gauss-Markov model defined above, these two questions turn out to be simpler. The following two propositions provide the answers and thus the theoretical foundation of δ-MED criterion:

Proposition 3.1 *If a vector $\vec{X}$ satisfies the first-order Gauss-Markov process, then the difference of any two adjacent elements in $\vec{X}$ tends to zero if $\rho \rightarrow 1$.*

Proof: *Let $x_{i-1}, x_i \in \vec{X}$ be two arbitrary adjacent elements. The variance of the difference of any two adjacent elements is given by*

$$\begin{aligned}
\mathbf{E}((x_{i-1} - x_i)(x_{i-1} - x_i)) &= \mathbf{E}(x_{i-1}x_{i-1} + x_i x_i - x_{i-1}x_i - x_i x_{i-1}) \\
&= \mathbf{E}(x_{i-1}x_{i-1}) + \mathbf{E}(x_i x_i) - \mathbf{E}(x_{i-1}x_i) - \mathbf{E}(x_i x_{i-1}) \\
&= 2 - 2\rho.
\end{aligned}$$

As $\rho \rightarrow 1$, the variance of the difference tends to zero. □

Proposition 3.1 tells us that if a part of an image satisfying the first-order Gauss-Markov model, it turns out that δ-MED criterion is automatically satisfied on that part. In the other words, the δ-MED criterion is an effective tool to identify the parts of an image satisfying the first-order Gauss-Markov model. Once those blocks satisfying this model are found out, our estimation scheme tries to fit just one domain block to these blocks with the same transformation.

Proposition 3.2 *Consider two adjacent vectors $\vec{X}_{i-1}$ and $\vec{X}_i$, both of length 2^B, satisfying the first-order Gauss-Markov process. If a vector $\vec{Y}$ is a good approximation to $\vec{X}_{i-1}$ up to a transformation $\mathbf{T}$, i.e., $d(\vec{X}_{i-1}, \mathbf{T}(\vec{Y})) = d_{i-1}$ where $d_{i-1} \approx 0$, then*

$$d(\vec{X}_i, \mathbf{T}(\vec{Y})) \leq d_{i-1} + \sqrt{2(1 - \rho^{2^B})} \tag{3.3}$$

Furthermore if $\rho \rightarrow 1$, then $d(\vec{X}_i, \mathbf{T}(\vec{Y})) \leq d_{i-1}$.

Proof: *Consider the collection $\mathcal{H}$ of all vectors of size $2^B \times 1$ with each element in a vector is a random variable. An inner product can be assigned to this collection: (see appendix A)*

$$< \vec{X}_1, \vec{X}_2 > = \mathbf{E}(\vec{X}_1^T \vec{X}_1) \quad , \quad \forall \vec{X}_1, \vec{X}_2 \in \mathcal{H}.$$

Therefore the following triangular inequality holds:

$$\begin{aligned} d(\vec{X}_i, \mathbf{T}(\vec{Y})) &\equiv \sqrt{< \vec{X}_i - \mathbf{T}(\vec{Y}), \vec{X}_i - \mathbf{T}(\vec{Y}) >} \\ &= \sqrt{\mathbf{E}((\vec{X}_i - \mathbf{T}(\vec{Y}))^T(\vec{X}_i - \mathbf{T}(\vec{Y})))} \\ &\leq \sqrt{\mathbf{E}((\vec{X}_{i-1} - \mathbf{T}(\vec{Y}))^T(\vec{X}_{i-1} - \mathbf{T}(\vec{Y})))} + \sqrt{\mathbf{E}((\vec{X}_i - \vec{X}_{i-1})^T(\vec{X}_i - \vec{X}_{i-1}))} \\ &= d(\vec{X}_{i-1}, \mathbf{T}(\vec{Y})) + \sqrt{\mathbf{E}\left(\sum_{i=1}^{2^B}(x_i - x_{i-1})^2\right)} \\ &= d_{i-1} + \sqrt{2(1-\rho^{2^B})} \end{aligned}$$

and $\sqrt{2(1-\rho^{2^B})} \to 0$ *as* $\rho \to 1$. □

The estimation scheme tries to fit both blocks with the same transformation. Proposition 3.2 tells us that it is at least theoretically possible to do so. A domain block being a good fit to the current block's neighbor up to a transformation may turn out to cover well to the current block with the same transformation. The error of this approximation is bounded by the error of the adjacent block fit up to an addition of a constant term $\sqrt{2(1-\rho^{2^B})}$. This constant term vanishes if the correlation coefficient tends to one.

One must note that there is a possibility that δ-MED may identify some parts with other statistical properties which satisfy this criterion. These blocks may not be able to be covered well by the same domain block. Some modifications are necessary to tackle these blocks that will be discussed in the next section.

3.2 Adjacent Block Parameter Estimation Scheme

After the δ-MED criterion has been defined in the last section, the proposed adjacent block parameter estimation scheme is introduced here that is based on the δ-MED criterion as the estimation predictor. Attractor coding schemes, both with and without the inclusion of an orthogonalization operator, are used here. The range pool contains

range block of size $2^B \times 2^B$. Redundancies in specifying the block parameters on smooth regions are exploited by the proposed estimation.

The proposed algorithm processes each block of the image from lower to right and from top to bottom just like the conventional approach. The major modification is in the domain block matching procedure. In the conventional attractor coding the domain matching is carried by exhaustive searching or some fast searching algorithm like the one proposed by Jacquin. The domain block with minimum distance to the range block up to a transformation is chosen. In the present scheme for each range block $\mathcal{R}_{i,j}$ the δ-MED criterion is performed before the domain block searching. If the current range block $\mathcal{R}_{i,j}$ fails to fulfill the δ-MED criterion, its transformation parameters will be found just like the conventional case. Otherwise, either its right block $\mathcal{R}_{i,j+2^B}$ or its lower block $\mathcal{R}_{i+2^B,j}$ or even both has a block boundary close to that of the current block. We term such an adjacent block which has a boundary close to that of the current block as the similar adjacent block respect to the current block. The current block may be covered well by a domain block and has similar transformation parameters as its similar adjacent block(s). In order to remove this kind of redundancies, a joint optimization is performed on these blocks.

3.2.1 Joint Optimization

If the current block satisfies the δ-MED criterion, it implies the current block and the adjacent block(s) are similar. Our goal is to exploit redundancies among these two or three blocks as much as possible. All these blocks are now approximated by exactly the same domain block with the same set of transformation parameters, i.e., $\mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I})$. (see Fig. 3.2) A joint optimization of these blocks is performed to find a common set of transformation parameters $(s_{i,j}, o_{i,j}, \mathbf{I}_{i,j}, \mathcal{D}_{k,l})$:

$$\min_{s_{i,j},o_{i,j},\mathbf{I}_{i,j},\mathcal{D}_{k,l}} \Big(\quad d(\mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I}), \mathcal{R}_{i,j}) + b_l * d(\mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I}), \mathcal{R}_{i,j+2^B}) + \quad b_r * d(\mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I}), \mathcal{R}_{i+2^B,j})\Big) \quad (3.4)$$

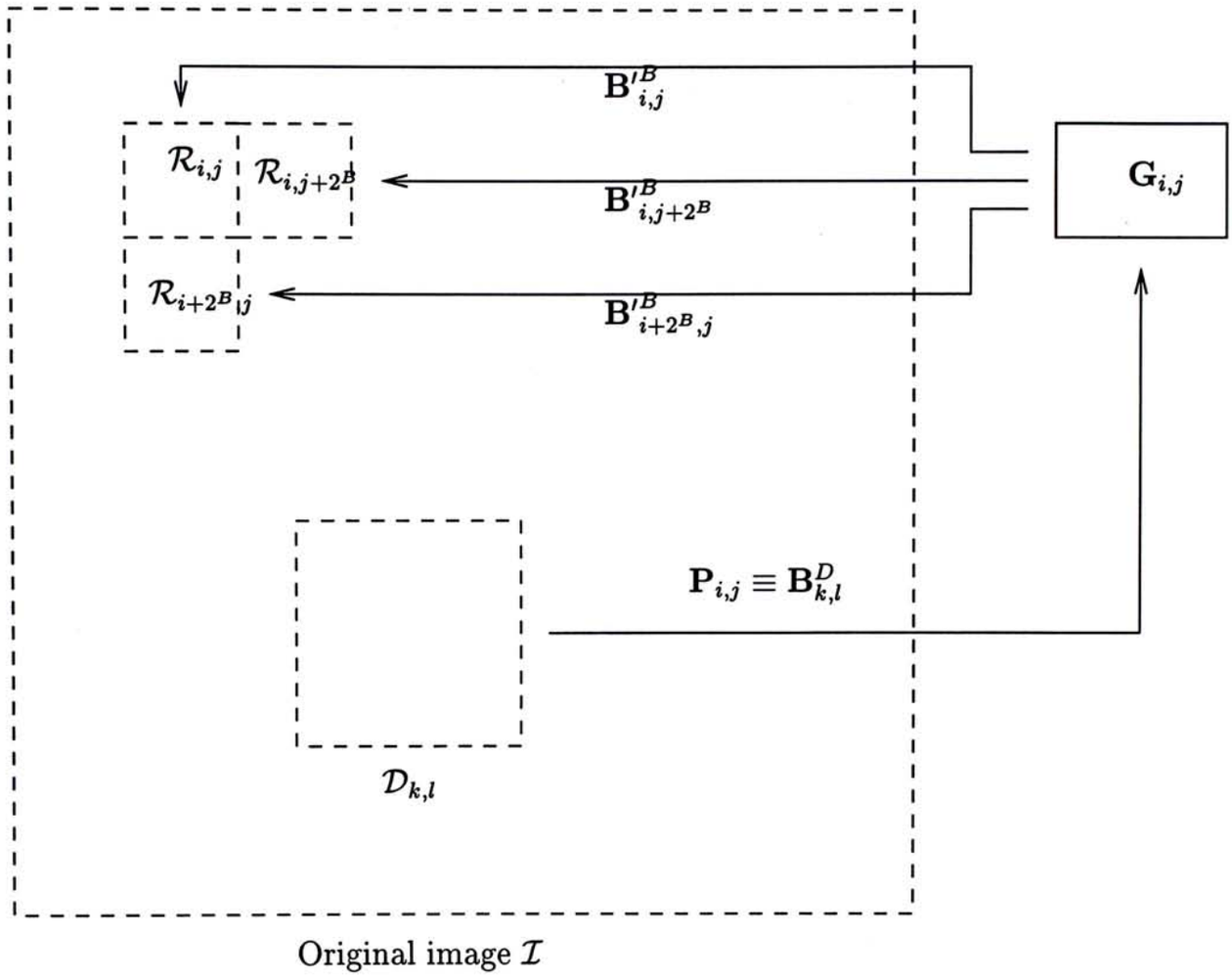


Figure 3.2: Joint optimization of three adjacent blocks.

where $b_l, b_r \in \{0, 1\}$ that turn to be one if their respective blocks are involved in the optimization. However, we find that it is difficult to solve this joint optimization problem. Given a domain block $\mathcal{D}_{k,l}$ and an isometry $\mathbf{I}_{i,j}$, the optimal $s_{i,j}$ and $o_{i,j}$ are found by setting their respective partial derivative of (3.4) to zero. The optimal parameters are obtained by solving two simultaneous equations. However, the four combinations of b_l and b_r, together with $s_{i,j}$ and $o_{i,j}$, make the optimization problem more complex. Solving this problem requires much more computations than the original collage minimization. It is thus not feasible to compute the optimal solution. Therefore, instead of solving it directly, a sub-optimal but faster method is used. The parameters for the current block are found first. In other words, (3.4) is solved by setting $b_l =$

$b_r = 0$:

$$\min_{s_{i,j},o_{i,j},\mathbf{I}_{i,j},\mathcal{D}_{k,l}} d(\mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I}), \mathcal{R}_{i,j}). \tag{3.5}$$

Then the values of b_l and b_r are determined by observing the collage distance on the lower and right blocks using the block parameters of the current block respectively:

$$d(\mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I}), \mathcal{R}_{i,j+2^B}) < d_T \implies b_r = 1, \tag{3.6}$$

$$d(\mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I}), \mathcal{R}_{i+2^B,j}) < d_T \implies b_l = 1 \tag{3.7}$$

where $d_T \in \mathbb{R}$ is a predefined error tolerance. This set of parameters is used for all these blocks in the decoder. Compression is obtained by storing only one set of parameters instead of two or more sets. Using this method, the number of computations in the joint optimization is reduced to the same order of the conventional PIFS.

3.2.2 Predictive Coding

Adjacent blocks in some cases fulfill the δ-MED criterion but fail to give satisfactory results on joint optimization. In other words, the collage distance in (3.6) or (3.6) in joint optimization exceeds a predefined error tolerance $d_T \in \mathbb{R}$. The current block shares the same domain block and a few parameters with its adjacent block(s) (either right or lower or both) while other parameters are found explicitly for the adjacent block(s). These new parameters must be known to the decoder and so consume some bits to represent them. Predictive coding is found to be appropriate for these parameters. Both the conventional and orthogonalized PIFS are discussed here.

Conventional PIFS: In the case of using conventional affine transformation, these blocks share the same domain block with its adjacent block(s). However, the isometry, offset and scaling coefficients for the adjacent block(s) are recomputed. Without loss of generality, the collage optimization of the lower block $\mathcal{R}_{i+2^B,j}$ is (see Fig. 3.3)

$$\min_{s_{i+2^B,j},o_{i+2^B,j},\mathbf{I}_{i+2^B,j}} d(s_{i+2^B,j} \times \mathbf{DI}_{i+2^B,j}\mathbf{P}_{i,j}(\mathcal{I}) + o_{i+2^B,j} \times \mathbf{U}^B, \mathcal{R}_{i+2^B,j}). \tag{3.8}$$

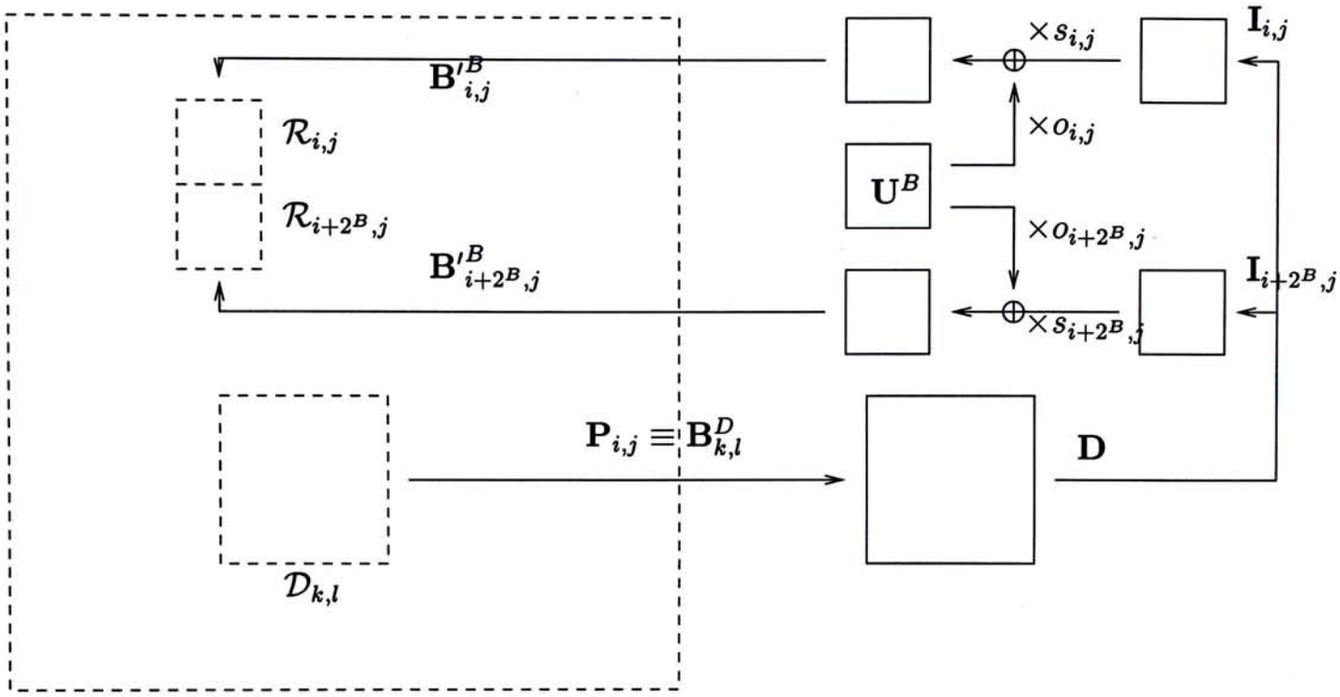


Figure 3.3: Predictive coding of two adjacent blocks in the conventional PIFS.

With the sharing of a single domain block for two adjacent blocks, just three parameters remain for storage. The isometry $\mathbf{I}_{i+2^B,j}$ is represented by 3 bits explicitly. For the offset parameter $o_{i+2^B,j}$ and scaling parameter $s_{i+2^B,j}$, they can be efficiently represented by the difference of those of the adjacent blocks and the reason is detailed as follows:

Assuming the right adjacent block $\mathcal{R}_{i,j}$ fulfills the δ-MED criterion, we have

$$\mathbf{E}\{s_{i+2^B,j} \times \mathbf{DI}_{i+2^B,j}\mathbf{P}_{i+2^B,j}(\mathcal{I}) + o_{i+2^B,j} \times \mathbf{U}^B\} \approx \mathbf{E}\{s_{i,j} \times \mathbf{DI}_{i,j}\mathbf{P}_{i+2^B,j}(\mathcal{I}) + o_{i,j} \times \mathbf{U}^B\}$$

or

$$K \times (s_{i+2^B,j} - s_{i-1,j}) \approx -o_{i+2^B,j} + o_{i-1,j}$$

where $K \equiv \mathbf{E}\{\mathbf{DI}_{i+2^B,j}\mathbf{P}_{i+2^B,j}(\mathcal{I})\} \equiv \mathbf{E}\{\mathbf{DI}_{i,j}\mathbf{P}_{i+2^B,j}(\mathcal{I})\}$. By defining

$$\begin{aligned} \Delta s_{i+2^B,j} &\equiv s_{i+2^B,j} - s_{i,j}, \\ \Delta o_{i+2^B,j} &\equiv o_{i+2^B,j} - o_{i,j}. \end{aligned}$$

We obtain

$$K\Delta s_{i+2^B,j} \approx -\Delta o_{i+2^B,j}. \tag{3.9}$$

Thus, $\Delta s_{i+2^B,j}$ is approximately directly proportional to $\Delta o_{i+2^B,j}$ with a negative proportional constant $-K \in \mathbb{R}$. By using this fact, both scaling and offset coefficients of two blocks can be stored compactly by first storing their respective differences followed by any entropy coder.

Orthogonalized PIFS: In the case of using orthogonalized affine transformation, these blocks share the same domain block and offset coefficient. However, the adjacent block uses different isometry and scaling coefficient in order to give a better fit. Without loss of generality, if the lower adjacent block has a close boundary with the current block, the collage optimization of the current block is (see Fig. 3.4)

$$\min_{s_{i+2^B,j},\mathbf{I}_{i+2^B,j}} d(s_{i+2^B,j} \times \mathbf{ODI}_{i+2^B,j}\mathbf{P}_{i,j}(\mathcal{I}) + o_{i,j} \times \mathbf{U}^B, \mathcal{R}_{i+2^B,j}). \tag{3.10}$$

Only two additional parameters are involved instead of four in the conventional case.

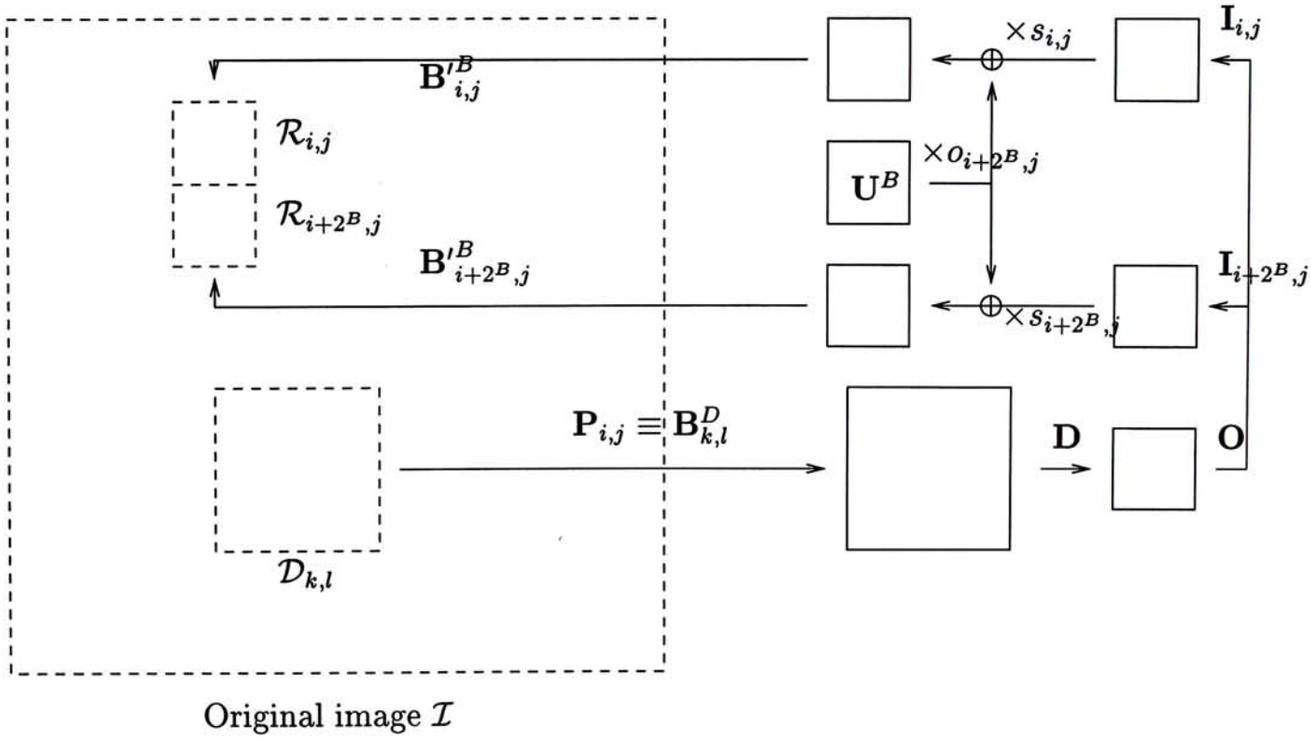


Figure 3.4: Predictive coding of two adjacent blocks in the orthogonalized PIFS.

The reason why two blocks share the same offset is detailed as follows: Assuming that two distinct offsets $o_{i,j}$ and $o_{i+2^B,j}$ are used, two adjacent blocks are similar and so

$$\mathbf{E}(s_{i+2^B,j} \times \mathbf{ODI}_{i+2^B,j}\mathbf{P}_{i+2^B,j}(\mathcal{I}) + o_{i+2^B,j} \times \mathbf{U}^B) \approx \mathbf{E}(s_{i,j} \times \mathbf{ODI}_{i,j}\mathbf{P}_{i+2^B,j}(\mathcal{I}) + o_{i,j} \times \mathbf{U}^B)$$

or

$$s_{i+2^B,j}\mathbf{E}(\mathbf{ODI}_{i+2^B,j}\mathbf{P}_{i+2^B,j}(\mathcal{I})) + o_{i+2^B,j}\mathbf{E}(\mathbf{U}^B) \approx s_{i,j}\mathbf{E}(\mathbf{ODI}_{i,j}\mathbf{P}_{i+2^B,j}(\mathcal{I})) + o_{i,j}\mathbf{E}(\mathbf{U}^B)$$

Since the composite operator **EO** maps all elements in the vector space to the element zero, $o_{i,j} \approx o_{i+2^B,j}$. Therefore sharing the same offset for these two blocks should not suffer from great loss in PSNR but reduces the number of parameters. In this case, only two parameters $s_{i+2^B,j}, \mathbf{I}_{i+2^B,j}$ are involved: $\mathbf{I}_{i+2^B,j}$ are stored explicitly by 3 bits but $s_{i+2^B,j}$ by predictive coding described as follows:

Since two similar adjacent blocks fulfill the δ-MED criterion, we have

$$s_{i+2^B,j} \times \mathbf{ODI}_{i+2^B,j}\mathbf{P}_{i+2^B,j}(\mathcal{I}) + o_{i+2^B,j} \times \mathbf{U}^B \approx s_{i,j} \times \mathbf{ODI}_{i,j}\mathbf{P}_{i+2^B,j}(\mathcal{I}) + o_{i+2^B,j} \times \mathbf{U}^B$$

$$\mathbf{ODI}_{i+2^B,j}\mathbf{P}_{i+2^B,j}(\mathcal{I}) \times (s_{i+2^B,j} - s_{i,j}) \approx 0$$

at the boundaries of two adjacent blocks. Hence,

$$\begin{aligned} \Delta s_{i+2^B,j} &\equiv (s_{i+2^B,j} - s_{i,j}) \\ &\approx 0. \end{aligned}$$

Thus, $\Delta s_{i+2^B,j}$ is close to zero for most of the cases. The isometry $\mathcal{I}_{i+2^B,j}$ is stored explicitly but the scaling coefficients of two similar adjacent blocks can be stored efficiently by storing their difference followed by any existing entropy coder.

3.3 Algorithmic Descriptions of the Proposed Scheme

The proposed estimation scheme processes each block of the image from lower to right and from top to bottom. An adjacent block estimation is performed before the domain block matching procedure. The current range block is first examined by the δ-MED criterion. If it satisfies the δ-MED criterion, a joint optimization is performed on these adjacent blocks. If it does not give a satisfactory results, predictive coding is applied

instead. Conventional domain block searching is used for those failing to satisfy the criterion.

In the decoder, the locations where the block parameters are predictively coded or the same as their adjacent blocks are required. This information can be effectively coded by five symbols defined in Table 3.1. Besides the block parameters, each range block has an extra parameter $c_{i+2^B,j}$. One of the five symbols is assigned to $c_{i+2^B,j}$ of each range block to represent the block nature. These symbols are coded by Huff-

Table 3.1: Symbols for indicating the range block nature

Symbol	Description
NO	new parameters from the image code
JOL	the same parameters of its left block
JOU	the same parameters of its upper block
PCL	parameters obtained from predictive coding of its left adjacent block
PCU	parameters obtained from predictive coding of its upper adjacent block

man and run-length coding in order to take into account the non-uniform probability distribution. The proposed estimation scheme is summarized as an algorithm 1.

The algorithm for the orthogonalized PIFS is obtained by replacing the affine transformation to the orthogonalized one and those lines "store $s_{i,j}, o_{i,j}, \mathbf{I}_{i,j}, \mathbf{P}_{i,j}$" by "store $s_{i,j}, \mathbf{I}_{i,j}, \mathbf{P}_{i,j}$" in algorithm 1. Moreover, the PIFS encoding method can be seen as a particular case of the proposed method. By putting $\delta < 0$, no range block can fulfill the δ-MED criterion as no vector norm is smaller than zero. Only the exhaustive domain block searching is performed.

3.4 Experimental Results

Simulations have been carried out on several images. Each of the images is partitioned into 8×8 disjoint square range blocks. 16×16 domain blocks are taken from the image

Algorithm 1 Proposed Estimation Scheme for the conventional PIFS

Require: $\delta > 0$, $d_T > 0$ and $\forall c_{i,j} = -1$
  for each $\mathcal{R}_{i,j} \in \mathsf{R}$ **do**
    if $c_{i,j} = -1$ **then**
      $c_{i,j} = \mathrm{NO}$
      $\min_{s_{i,j}, o_{i,j}, \mathbf{I}_{i,j}, \mathcal{D}_{k,l}} d(\mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I}), \mathcal{R}_{i,j})$
      if $i < 2^{N-B}$ **then**
        Compute $\|\vec{e}_{l,i,j}\|$
      else
        $\|\vec{e}_{l,i,j}\| = \infty$
      end if
      if $j < 2^{N-B}$ **then**
        Compute $\|\vec{e}_{r,i,j}\|$
      else
        $\|\vec{e}_{r,i,j}\| = \infty$
      end if
      if $\min\{\|\vec{e}_{l,i,j}\|, \|\vec{e}_{r,i,j}\|\} < \delta$ **then**
        if $d(\mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I}), \mathcal{R}_{i,j+2^B}) < d_T$ and $c_{i,j} \neq -1$ **then**
          $c_{i,j} = \mathrm{JOL}$
        else
          $\min_{s_{i,j+2^B}, o_{i,j+2^B}, \mathbf{I}_{i,j+2^B}} d(s_{i,j+2^B} \times \mathbf{DI}_{i,j+2^B}\mathbf{P}_{i,j}(\mathcal{I}) + o_{i,j+2^B} \times \mathbf{U}^B, \mathcal{R}_{i,j+2^B})$
          if $d(s_{i,j+2^B} \times \mathbf{DI}_{i,j+2^B}\mathbf{P}_{i,j}(\mathcal{I}) + o_{i,j+2^B} \times \mathbf{U}^B, \mathcal{R}_{i,j+2^B}) < d_T$ **then**
            $c_{i,j} = \mathrm{PCL}$
            store $s_{i,j+2^B}, o_{i,j+2^B}, \mathbf{I}_{i,j+2^B}$
          end if
        end if
        if $d(\mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I}), \mathcal{R}_{i+2^B,j}) < d_T$ and $c_{i,j} \neq -1$ **then**
          $c_{i,j} = \mathrm{JOU}$
        else
          $\min_{s_{i+2^B,j}, o_{i+2^B,j}, \mathbf{I}_{i+2^B,j}} d(s_{i+2^B,j} \times \mathbf{DI}_{i+2^B,j}\mathbf{P}_{i,j}(\mathcal{I}) + o_{i+2^B,j} \times \mathbf{U}^B, \mathcal{R}_{i+2^B,j})$
          if $d(s_{i+2^B,j} \times \mathbf{DI}_{i+2^B,j}\mathbf{P}_{i,j}(\mathcal{I}) + o_{i+2^B,j} \times \mathbf{U}^B, \mathcal{R}_{i+2^B,j}) < d_T$ **then**
            $c_{i,j} = \mathrm{PCU}$
            store $s_{i+2^B,j}, o_{i+2^B,j}, \mathbf{I}_{i+2^B,j}$
          end if
        end if
      else
        store $s_{i,j}, o_{i,j}, \mathbf{I}_{i,j}, \mathbf{P}_{i,j}$
      end if
    end if
  end for

to form the library in which each of the domain blocks overlaps half of its support with its adjacent domain blocks, i.e., D is defined by (2.10). Without any estimation scheme, there are altogether 4096 blocks to be encoded for a 512×512 image. That means 4096 sets of block parameters are required for the PIFS. The goal of the proposed method is to reduce the number of parameters without significant loss in the decoded image.

In the experiments, the block parameters are quantized and coded in the same way as the coding examples in section 2.5.2. Two extra parameters, δ and d_T, must be chosen carefully. δ is chosen to be 8 which means δ-MED criterion is satisfied if the boundary pixels of the adjacent blocks are on average one pixel apart. d_T, on the other hand, controls the tradeoff between the joint optimization and the predictive coding. It tells the encoder how large the error in joint optimization deserves the use of predictive coding instead. In order to reduce the extra loss owing to the inclusion of the estimation scheme, d_T is chosen to be the RMSE of the decoded image produced by the PIFS without any estimation.

Both the conventional and the orthogonalized PIFS are tested in our experiments. In the case of using the conventional PIFS, table 3.2 shows the number of blocks that can be estimated by their adjacent blocks of four different images. It is found that about $10 - 14\%$ blocks can be jointly optimized with its adjacent blocks and about $13 - 26\%$ blocks can be coded by predictive coding. Table 3.3 compares the PSNR of the decoded images with and without estimation. It can be shown that the PSNR decreases by a very small amount while the bit rates can be reduced by about 20%. Table 3.4 and table 3.5 show the corresponding numerical results for the orthogonalized PIFS. It is found that about $9 - 13\%$ blocks use joint optimization and about $14 - 26\%$ blocks are coded in the predictive way. Fig. 3.7 and 3.8 show two decoded images, Lena and Fruits, produced by an adaptive quadtree partition and non-adaptive partition both with and without estimation.

The following observations are found from the simulation results:

Observation 1: The percentages of these blocks are picture dependent in both cases. In general an image with more low activity regions can benefit more by coding most blocks by joint optimization and predictive coding. This shows that the proposed method is effective for removing redundancies in coding low activity regions. Fig. 3.5 and 3.6 show the locations of blocks to be estimated in the images, Lena and Fruits. Most of these estimated blocks are located in the smooth regions of an image that can verify the proposed scheme removes redundancies in smooth regions.

Observation 2: It can be seen that even the decoded images by adaptive scheme have similar PSNRs with those by our proposed estimation scheme, the undesirable blockiness is much visible. One can easily observe that the decoded images with estimation are very close to those without estimation. Thus, unlike the adaptive-partition fractal coding schemes which introduce blocking effect on the smooth regions, the proposed adjacent block parameters estimation scheme does not produce additional blocking artifacts on the decoded images.

Observation 3: It is found that, for the same image, the number of blocks estimated are roughly the same for the conventional and orthogonalized PIFS. That means the percentages of these blocks are quite independent of the affine transformation used. Therefore both gives very similar performance in terms of bit rate. One interesting thing is the conventional PIFS gives slightly more blocks in joint optimization but lesser in predictive coding compared with the orthogonalized counterpart.

Observation 4: The structure of the PIFS, no matter conventional or orthogonalized, is by no means modified in our proposed method. The domain block matching is the same as those in most existing schemes. Therefore all fast algorithms for encoding and decoding devised for the existing schemes can be applied. Moreover, no time-consuming block matching is needed for those estimated blocks and so the encoding time is shorter.

Table 3.2: Number of range blocks to be estimated by joint optimization and coded by predictive coding in the conventional PIFS

Image	Dimension	Total number of blocks	Number of blocks estimated by joint optimization	Number of blocks coded by predictive coding
Lena	512×512	4096	410	526
Fruits	480×512	3840	404	629
Flower	480×512	3840	447	989
Tiffany	512×512	4096	564	606

Table 3.3: Performance of the proposed estimation method using the conventional PIFS

Image	Without estimation		With estimation	
	bpp	PSNR	bpp	PSNR
Lena	0.42	31.0	0.37	31.0
Fruits	0.42	31.7	0.36	31.6
Flower	0.42	32.6	0.33	32.5
Tiffany	0.42	29.9	0.35	29.8

Table 3.4: Number of range blocks to be estimated by joint optimization and coded by predictive coding in the orthogonalized PIFS

Image	Dimension	Total number of blocks	Number of blocks estimated by joint optimization	Number of blocks coded by predictive coding
Lena	512×512	4096	373	565
Fruits	480×512	3840	399	653
Flower	480×512	3840	436	1019
Tiffany	512×512	4096	540	637

Table 3.5: Performance of the proposed estimation method using the orthogonalized PIFS

Image	Without estimation		With estimation	
	bpp	PSNR	bpp	PSNR
Lena	0.42	31.66	0.37	31.63
Fruits	0.42	32.18	0.36	32.16
Flower	0.42	33.13	0.33	33.10
Tiffany	0.42	32.47	0.34	32.44

Figure 3.5: The locations of those blocks using joint optimization (white blocks) and those estimated by predictive coding (black blocks) of image Lena in the proposed estimation scheme using the conventional PIFS.

Figure 3.6: The locations of those blocks using joint optimization (white blocks) and those estimated by predictive coding (light-gray blocks) of image Fruits in the proposed estimation scheme using the conventional PIFS.

(a) original image Lena

(b) quadtree partition: 0.40 bpp, PSNR=30.95dB

(c) non-adaptive partition without estimation: 0.42 bpp, PSNR=31.0dB

(d) non-adaptive partition with estimation: 0.37 bpp, PSNR=31.0dB

Figure 3.7: Experimental result of image Lena.

(a) original image Fruits

(b) quadtree partition: 0.37 bpp, PSNR=31.0dB

(c) non-adaptive partition without estimation: 0.42 bpp, PSNR=31.7dB

(d) non-adaptive partition with estimation: 0.36 bpp, PSNR=31.6dB

Figure 3.8: Experimental result of image Fruits.

3.5 Summary

In this chapter, an estimation scheme is proposed to reduce the bits required in specifying the block parameters of the smooth regions on an image based on the PIFS of constant range block size. The criterion to select those blocks for estimation is the δ-minimum edge difference (δ-MED) condition. If the δ-MED criterion is satisfied, the adjacent blocks are likely of low activities. A joint optimization is performed on these adjacent blocks. If it does not give a satisfactory results, predictive coding is applied instead. Conventional domain block searching is used for those failing to satisfy the criterion.

Owing to the use of constant range block size, most blockiness of the decoded images are now hidden in the high-activity regions that, owing to the masking effect of HVS, are not easily observed. This novel attractor coding method can effectively code an image suffered with lesser blocking artifacts and achieve the compression ratio compatible to those using the variable block size. Since the structure of the transformation $\mathbf{T}$ is not modified in the proposed estimation method, all fast algorithms devised for the existing coding methods can be applied directly.

Chapter 4

Attractor Coding using Lapped Partitioned Iterated Function Systems

In the last chapter an estimation scheme is proposed to hide most of the blockiness such that the human eyes cannot easily spot them out even though the blocking effect still exists. The basic structure of the transformation is by no means altered in the proposed estimation scheme. An image is still partitioned into disjoint range blocks. Owing to the partitioning of an image into disjoint range blocks, blocking artifacts which is highly visibly annoying to human visual system occur.

One solution to this problem is to allow range blocks to overlap. IFS with overlaps has gained interests in theoretical studies of multifractal structure of self-similar measures [15]. However, little has been done in constructive and algorithmic direction. Reusens [43] proposed to use overlapping range blocks and average the pixel values on the overlapping regions during decoding. However, the image details within the overlapping regions are blurred because of the pixel averaging. Therefore the overlapping regions are restricted to the boundary pixels of blocks only in order to minimize

the undesirable blurring effect. It is expected that the blockiness removal is not so effective and considerable blockiness still exists in the decoded image. On the other hand, Forte and Vrscay [19, 20] found that such blurring can be avoided if the global collage distance for all blocks is minimized simultaneously instead of minimizing collage distance for each block independently. The collage minimization is formulated as a system of equations involving offset and scaling parameters and becomes a quadratic programming (QP) problem in these two parameters. However, a very large system of equations is involved in this optimization and so a large amount of computations and memories is used in encoding.

In this work, we propose a new coding scheme using partitioned iterated function systems with lapped range blocks (LPIFS) [25]. The invention of this novel coding method is basically motivated by Malvar's lapped orthogonal transform (LOT) [33] and its two generalized forms, GenLOT [42] and the biorthogonal local trigonometric bases [29]. Traditional transform coding methods like JPEG partition an image into disjoint blocks and apply one kind of energy-packing transforms on each of the blocks. The transform coefficients of each block are quantized and coded. Blocking artifacts thus occur in these methods. Based on this observation, LOT was proposed by Malvar to solve this problem. The transform is applied on overlapping blocks and the transform coefficients are optimized to alleviate the blockiness. Moreover, each of the bases in the LOT are orthogonal to each other and thus no aliasing occurs.

In this novel coding method using LPIFS, an image is no longer partitioned into disjoint range blocks. Instead, each range block laps with its adjacent blocks through a weighting operator which acts as a window diminishing in magnitudes towards its window boundaries. Therefore the blocking artifacts are removed through this smooth weighting window. The transformation parameters are computed in such a way that most undesirable blurring is eliminated. Therefore most image details are preserved and no large system of equations is needed.

4.1 Lapped Partitioned Iterated Function Systems

In this section, a novel mathematical formulation of PIFS in which the range blocks overlap is introduced. We term this new formulation of the PIFS as the lapped partitioned iterated function systems (LPIFS). Up to the author's knowledge, the same formulation is not presented or invented by the others. The fundamental difference between LPIFS and PIFS is the inclusion of the weighting operator on each overlapping range block that does not appear in the formulation of PIFS.

The image $\mathcal{I}$, being a summation of disjoint range blocks in the conventional PIFS, is now expressed as a weighted summation of all range blocks under the LPIFS settings. In order to highlight the property of overlapping in the image sub-blocks in LPIFS, a tilde is added to distinguish those of the same kinds in the PIFS. The range pool and domain pool are alternatively defined as follows:

Definition 4.1 *(Range pool) An image $\mathcal{I}$ is partitioned into a number of square range blocks of size $2^{B+1}\times 2^{B+1}$ The collection $\tilde{\mathsf{R}}$ of all overlapping range block $\tilde{\mathcal{R}}_{i,j}$, i.e.,*

$$\{\tilde{\mathcal{R}}_{i,j} : \tilde{\mathcal{R}}_{i,j} \equiv \mathbf{B}_{i,j}^{B+1}\mathcal{I} \text{ with } i = 2^B p, j = 2^B q, 0 \leq p, q < 2^{N-B-1}\}.$$

is defined as the range pool in the LPIFS.

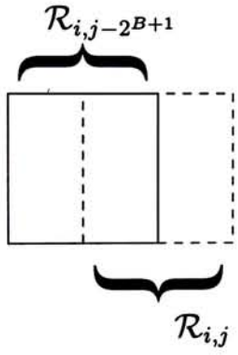


Figure 4.1: The overlapping range blocks in $\tilde{\mathsf{R}}$.

Remark 4.1 *Under this definition, each $\tilde{\mathcal{R}}_{i,j}$ contains $2^{B+1}\times 2^{B+1}$ pixels and adjacent range blocks are just 2^B pixels apart. Every block overlaps each of its adjacent blocks*

with half of its support. The size of each range block is four times of its counterpart in the PIFS and the number of range blocks is $(2^{N-B}-1)\times(2^{N-B}-1)$.

Remark 4.2 *Every sub-block* $\mathbf{B}^B_{2^B p,2^B q}\mathcal{I}$ *(with* $0 \le p,q < 2^{N-B}$*) in* $\mathcal{I}$ *is now covered by four range blocks with this lapped partition, except those lying on the image boundary. To be more precise, each of* $\mathbf{B}^B_{2^B p,2^B q}\mathcal{I}$, $0 < p,q < 2^{N-B}-1$, *is covered by four range blocks. Each of the boundary sub-blocks,* $\mathbf{B}^B_{2^B p,0}\mathcal{I}$ *and* $\mathbf{B}^B_{0,2^B q}\mathcal{I}$ *with* $0 < p,q < 2^{N-B}-1$, *is covered two range blocks and each of the four sub-blocks,* $\mathbf{B}^B_{k,l}\mathcal{I}$ *with* $k,l \in \{0, 2^{N-B}-1\}$, *at the corners is covered by one range block.*

Definition 4.2 *(Domain pool) A domain pool or library* D *is defined as*

$$\mathsf{D} \equiv \{\mathcal{D}_{k,l} : \mathcal{D}_{k,l} \equiv \mathbf{B}^{B+2}_{k,l}\mathcal{I} \text{ with } k = 2^B p, l = 2^B q, 0 \le p,q < 2^{N-B-2}\}.$$

Remark 4.3 D *contains domain blocks* $\mathcal{D}_{k,l}$ *of size* $2^{B+2}\times 2^{B+2}$ *to keep the decimated* $\mathcal{D}_{k,l}$ *to be* $2^{B+1}\times 2^{B+1}$. *It must be noted that any collection of sub-blocks of size larger than a range block is a valid library. This particular choice of* D *is just for the ease of implementation.*

4.1.1 Weighting Operator

The LPIFS is formulated in a similar fashion as the convenient PIFS. All operators employed in the convenient PIFS appears in the present formulation. The major modification is the introduction of the weighting operator $\mathbf{W}$ introduced here.

Definition 4.3 *(Weighting operator)* $\mathbf{W} : \mathbb{R}^{2^{B+1}\times 2^{B+1}} \to \mathbb{R}^{2^{B+1}\times 2^{B+1}}$ *is defined as a linear operator which multiplies each element in a given sub-block of size* $2^{B+1}\times 2^{B+1}$ *by the corresponding weight* $w_{u,v} \in \mathbb{R}$ *with* $u,v \in \{0,1,\ldots,2^{B+1}-1\}$*:*

$$\mathbf{W}\left([i_{u,v}]_{0\le u,v<2^{B+1}}\right) \equiv [w_{u,v}i_{u,v}]_{0\le u,v<2^{B+1}} \tag{4.1}$$

with the following constraint:

$$w_{u,v} + w_{u+2^B,v} + w_{u,v+2^B} + w_{u+2^B,v+2^B} = 1, \quad \forall\ 0 \leq u, v < 2^B. \tag{4.2}$$

The function of $\mathbf{W}$ is to assign weights to the pixels in the affine-transformed domain block before it is inserted into the image. There are altogether $2^{B+1} \times 2^{B+1}$ numbers of $w_{u,v}$ in $\mathbf{W}$. The weights $w_{u,v}$ are chosen to provide smooth overlapping across block boundaries in order to minimize the blocking artifacts. There is only one constraint, i.e.,(4.2), on the choice of $w_{u,v}$: the sum of $w_{u,v}$ for different adjacent range blocks on the same spatial location must be one.

Proposition 4.1 *The weighting operator* $\mathbf{W}$ *in definition 4.3 satisfies the following matrix identity:*

$$\sum_{k,l \in \{0,2^B\}} \mathbf{B}^B_{k,l} \mathbf{W} \mathbf{U}^{B+1} = \mathbf{U}^B. \tag{4.3}$$

and the following operator identity:

$$\sum_{k,l \in \{0,2^B\}} \mathbf{B}^B_{k,l} \mathbf{W} \mathbf{B}^{B+1}_{m,n} \mathbf{B'}^B_{m+k,n+l} = \mathbf{I}, \quad \forall 0 \leq m, n < 2^{N-B-1}, \tag{4.4}$$

where $\mathbf{I} : \mathbb{R}^{2^B \times 2^B} \rightarrow \mathbb{R}^{2^B \times 2^B}$ *is the identity operator that maps an element to itself.*

Proof: *For the matrix identity,*

$$\begin{aligned}
\sum_{k,l \in \{0,2^B\}} \mathbf{B}^B_{k,l} \mathbf{W} \mathbf{U}^{B+1} &= \sum_{k,l \in \{0,2^B\}} \mathbf{B}^B_{k,l} \left[w_{u,v} \right]_{0 \leq u,v < 2^{B+1}} \\
&= \sum_{k,l \in \{0,2^B\}} \left[w_{k+u,l+v} \right]_{0 \leq u,v < 2^B} \\
&= \left[\sum_{k,l \in \{0,2^B\}} w_{k+u,l+v} \right]_{0 \leq u,v < 2^B} \\
&= \left[1 \right]_{0 \leq u,v < 2^B} \\
&= \mathbf{U}^B
\end{aligned}$$

For the operator identity, consider $[x_{u,v}]_{0\le u,v<2^B} \in \mathbb{R}^{2^B\times 2^B}$,

$$\begin{aligned}\sum_{k,l\in\{0,2^B\}} \mathbf{B}^B_{k,l}\mathbf{W}\mathbf{B}^{B+1}_{m+k,n+l}\mathbf{B}'^B_{m+k,n+l}[x_{u,v}]_{0\le u,v<2^B} &= \sum_{k,l\in\{0,2^B\}} \mathbf{B}^B_{k,l}[w_{k+u,l+v}x_{u,v}]_{0\le u,v<2^B}\\ &= \left[\sum_{k,l\in\{0,2^B\}} w_{k+u,l+v}x_{u,v}\right]_{0\le u,v<2^B}\\ &= [x_{u,v}]_{0\le u,v<2^B} \quad \square\end{aligned}$$

Using the weighting operator $\mathbf{W}$ and the range pool $\tilde{\mathsf{R}}$, any image $\mathcal{I}$ can be expressed as the weighted sum of all range blocks in $\tilde{\mathsf{R}}$. However, remark 4.2 tells us that not all $2^B\times 2^B$ sub-blocks $\mathbf{B}^B_{2^Bp,2^Bq}\mathcal{I}$ (with $0 \le p,q < 2^{N-B}$) are covered by exactly four range blocks $\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}$. The weights assigned to the boundary range blocks need some modifications in order to provide real weights for the boundary blocks. Accordingly, for those sub-blocks covered by just two range blocks, i.e, $\mathbf{B}^B_{2^Bp,0}\mathcal{I}$ and $\mathbf{B}^B_{0,2^Bq}\mathcal{I}$ with $0 < p,q < 2^{N-B}-1$, every pixel within these sub-blocks is the sum of two weighted pixels only. Therefore the sum of two weighted pixels is divided by the sum of two weights. For example, $\mathbf{B}^B_{0,2^Bq}\mathcal{I}$ where $0 < q < 2^{N-B}$ is a sub-block just covered by two range blocks, namely, $\tilde{\mathcal{R}}_{0,2^Bq}$ and $\tilde{\mathcal{R}}_{0,2^B(q-1)}$. Therefore $\mathbf{W}$ for those $\tilde{\mathcal{R}}_{0,2^Bq}$ with $0 < q < 2^{N-B-1}$ are changed to map each $i_{u,v}$ to $\frac{w_{u,v}}{w_{u,v}+w_{u,2^B+v}}i_{u,v}$ for $0 \le u < 2^B$ and map other $i_{u,v}$ to $w_{u,v}i_{u,v}$. For the sub-blocks $\mathbf{B}^B_{k,l}\mathcal{I}$ with $k,l \in \{0, 2^{N-B}-1\}$, each of these four sub-blocks is covered by just one range block. Therefore the weights on these blocks are divided by themselves, i.e., the weights are $\frac{w_{u,v}}{w_{u,v}} = 1$. With the weights on the boundary blocks modified, corollary 4.1 follows:

Corollary 4.1 *Any image* $\mathcal{I} \in \{\mathcal{I}_i\}$ *can be expressed as the weighted summation of all range blocks: (compare with (2.10))*

$$\mathcal{I} \equiv \sum_{\tilde{\mathcal{R}}_{i,j}\in\tilde{\mathsf{R}}} \mathbf{B}'^{B+1}_{i,j}\mathbf{W}(\tilde{\mathcal{R}}_{i,j}). \tag{4.5}$$

The use of special techniques to handle boundaries of a finite-length signal is not new in the scope of signal processing. The bases in the LOT and the GenLOT of the

boundary blocks are altered [33, 42]. In discrete wavelet transform (DWT), one way to handle the boundaries of the finite-length signals is to apply the boundary filters for perfect reconstruction (PR) [49].

4.1.2 Mathematical Formulation of the LPIFS

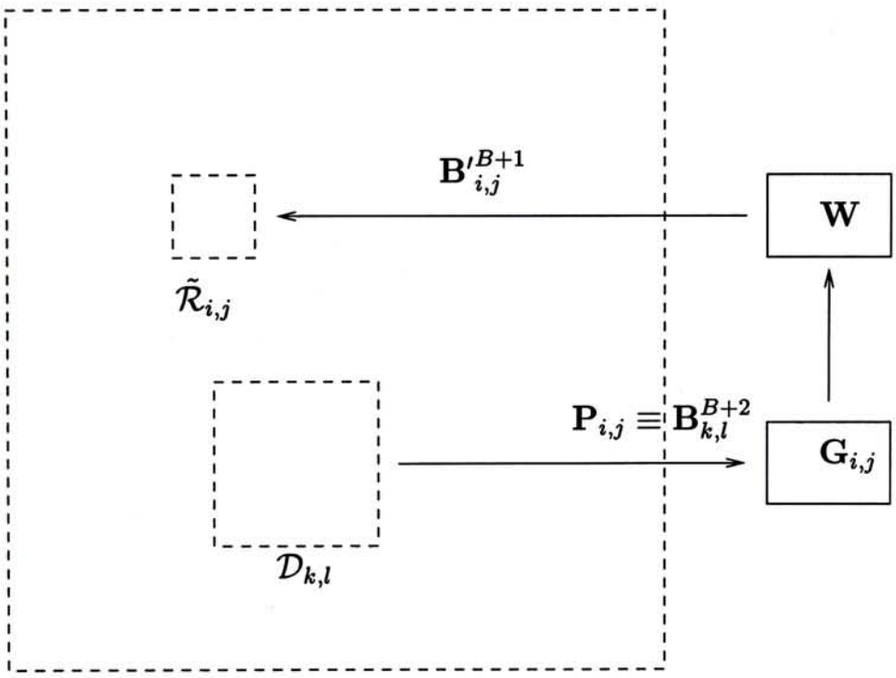


Figure 4.2: Affine transformation on a range block in the LPIFS.

Affine transformation is defined on each range block that maps a domain block within the same image to the specific range block. Affine transformation of either the conventional PIFS (2.11) or the orthogonalized PIFS (2.20) can serve this purpose.

For every $\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}$ an affine transformation is defined in the following way:

1. Define $\mathbf{P}_{i,j} : \{\mathcal{I}_i\} \rightarrow \mathbb{R}^{2^{B+2} \times 2^{B+2}}$ to be an operator which gets a domain block $\mathcal{D}_{k,l}$ from D, i.e., $\mathbf{P}_{i,j} \equiv \mathbf{B}^{B+2}_{k,l}$.

2. Let $\mathbf{G}_{i,j} : \mathbb{R}^{2^{B+2} \times 2^{B+2}} \rightarrow \mathbb{R}^{2^{B+1} \times 2^{B+1}}$ be the affine transformation which approximates $\tilde{\mathcal{R}}_{i,j}$ by the affine-transformed domain block $\mathcal{D}_{k,l}$. $\mathbf{G}_{i,j}$ can be defined in

one of the two ways:

$$\mathbf{G}_{i,j}(\mathcal{D}_{k,l}) = s_{i,j} \times \mathbf{DI}_{i,j}(\mathcal{D}_{k,l}) + o_{i,j} \times \mathbf{U}^{B+1} \quad \text{(Conventional PIFS)} \quad (4.6)$$

$$\mathbf{G}_{i,j}(\mathcal{D}_{k,l}) = s_{i,j} \times \mathbf{ODI}_{i,j}(\mathcal{D}_{k,l}) + o_{i,j} \times \mathbf{U}^{B+1} \quad \text{(Orthogonalized PIFS)} (4.7)$$

with $\mathbf{D}$ and $\mathbf{I}_{i,j}$ defined in the same way as (2.11) and (2.20) by replacing B by $B+1$ in their definitions. $\mathbf{O}$ is defined in the definition 2.6 with B replaced by $B+1$.

3. Apply $\mathbf{W}$ defined in definition 4.1 on the block $\mathbf{G}_{i,j}(\mathcal{D}_{k,l})$.

4. Define $\mathbf{B}'^{B+1}_{i,j} : \mathbb{R}^{2^{B+1}\times 2^{B+1}} \rightarrow \{\mathcal{I}_i\}$ which puts the resultant block at the range block location in the image.

Similar to the overall transformation $\mathbf{T}$ in the PIFS, the overall transformation on the image space is constructed by taking all affine transformation $\mathbf{G}_{i,j}$ on each of the range blocks together. Since the range blocks overlap, every pixel value on the overlapping regions is the result of weighting of all covering range blocks by $\mathbf{W}$.

Definition 4.4 *(Overall transformation) The overall transformation* $\tilde{\mathbf{T}} : \{\mathcal{I}_i\} \rightarrow \{\mathcal{I}_i\}$ *of the proposed LPIFS is defined as:*

$$\tilde{\mathbf{T}}\mathcal{I} \equiv \sum_{\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}} \mathbf{B}'^{B+1}_{i,j} \mathbf{W} \mathbf{G}_{i,j} \mathbf{P}_{i,j}(\mathcal{I}), \quad \forall \mathcal{I} \in \{\mathcal{I}_i\} \quad (4.8)$$

with $\mathbf{G}_{i,j}$ *defined by either (4.6) or (4.7).*

This is the overall transformation on the image space that is used in the proposed attractor image coding. The idea of the proposed coding method is similar to those using the PIFS: optimizing the collage by tuning the parameters of the transformation. Several properties of the transformation $\tilde{\mathbf{T}}$ are discussed before we go to cover the details of the coding method in the next section.

If all weights $\omega_{u,v}$ are chosen *a priori*, each $\mathbf{G}_{i,j}$ has the same number of transformation parameters as its counterpart in $\mathbf{T}$ defined in (2.12), i.e., $\mathbf{G}_{i,j}$ consists of the four parameters:

1. The scaling coefficients $s_{i,j}$,
2. The offset coefficients $o_{i,j}$,
3. The isometry $\mathbf{I}_{i,j}$ and
4. The domain block chosen $\mathbf{P}_{i,j}$.

These four parameters of each $\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}$ completely specify the transformation $\tilde{\mathbf{T}}$. Remark 4.1 states that there are altogether $(2^{N-B}-1)\times(2^{N-B}-1)$ number of $\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}$. Therefore the total number of parameters in specifying $\tilde{\mathbf{T}}$ is $4\times(2^{N-B}-1)\times(2^{N-B}-1)$.

Proposition 4.2 *$\tilde{\mathbf{T}}$ is an affine transformation on the image space $\{\mathcal{I}_i\}$ using either (4.6) or (4.7) as the definition of $\mathbf{G}_{i,j}$. Moreover, the contractivity of $\tilde{\mathbf{T}}$ is controlled solely by the linear term and is independent of the constant term.*

Proof: *By putting (4.6) into (4.8), $\tilde{\mathbf{T}}$ can be rewritten as*

$$\tilde{\mathbf{T}}\mathcal{I} \equiv \left(\sum_{\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}} s_{i,j} \times \mathbf{B}'^{B+1}_{i,j} \mathbf{W} \mathbf{D} \mathbf{I}_{i,j} \mathbf{P}_{i,j} \right) \mathcal{I} + \sum_{\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}} o_{i,j} \times \mathbf{B}'^{B+1}_{i,j} \mathbf{W} \mathbf{U}^{B+1}. \tag{4.9}$$

If (4.7) is used instead, $\tilde{\mathbf{T}}$ is given by

$$\tilde{\mathbf{T}}\mathcal{I} \equiv \left(\sum_{\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}} s_{i,j} \times \mathbf{B}'^{B+1}_{i,j} \mathbf{W} \mathbf{O} \mathbf{D} \mathbf{I}_{i,j} \mathbf{P}_{i,j} \right) \mathcal{I} + \sum_{\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}} o_{i,j} \times \mathbf{B}'^{B+1}_{i,j} \mathbf{W} \mathbf{U}^{B+1}. \tag{4.10}$$

The first part of both equations (4.9) and (4.10) is the linear term and the second part is the constant term. Therefore $\tilde{\mathbf{T}}$ is an affine transformation on the image space $\{\mathcal{I}_i\}$.

Being an affine transformation, we denote $\tilde{\mathbf{T}}_L$ and $\tilde{\mathbf{T}}_C$ as the linear part and the constant part of $\tilde{\mathbf{T}}$ respectively, i.e., $\tilde{\mathbf{T}}\mathcal{I} \equiv \tilde{\mathbf{T}}_L\mathcal{I} + \tilde{\mathbf{T}}_C, \quad \forall \mathcal{I} \in \{\mathcal{I}_i\}$. Then, $\forall \mathcal{I}_1, \mathcal{I}_2 \in$

$\{\mathcal{I}_i\}$ and any positive integer k,

$$\begin{aligned} d(\tilde{\mathbf{T}}^k\mathcal{I}_1, \tilde{\mathbf{T}}^k\mathcal{I}_2) &\equiv \|\tilde{\mathbf{T}}^k\mathcal{I}_1 - \tilde{\mathbf{T}}^k\mathcal{I}_2\| \\ &= \| \left(\tilde{\mathbf{T}}_L^k\mathcal{I}_1 + \sum_{i=0}^{k-1} \tilde{\mathbf{T}}_L^i\tilde{\mathbf{T}}_C\right) - \left(\tilde{\mathbf{T}}_L^k\mathcal{I}_2 + \sum_{i=0}^{k-1} \tilde{\mathbf{T}}_L^i\tilde{\mathbf{T}}_C\right) \| \\ &= \| \left(\tilde{\mathbf{T}}_L^k\mathcal{I}_1 - \tilde{\mathbf{T}}_L^k\mathcal{I}_2\right) \| \end{aligned} \tag{4.11}$$

where $\mathbf{T}^k$ denotes the k-th iteration of the transformation $\mathbf{T}$. Therefore we can see that the contractivity condition in definition 2.1 depends only on $\tilde{\mathbf{T}}_L$ and is independent of $\tilde{\mathbf{T}}_C$. □

Up to this stage, the weights $w_{u,v}$ are allowed to choose from any real values. The admission of different choice of $\mathbf{W}$ makes LPIFS a generalized technique to include PIFS as a special case.

Proposition 4.3 *If the weights $w_{u,v}$ in $\mathbf{W}$ are chosen as follows:*

$$w_{u,v} = \begin{cases} 1 & \text{if } \frac{1}{4}\times 2^{B+1} \le u, v < \frac{3}{4}\times 2^{B+1}, \\ 0 & \text{otherwise.} \end{cases}$$

Then, LPIFS becomes PIFS with non-overlapping range blocks of size $2^B \times 2^B$ except those at the boundaries.

Proof: *From (4.8), $\tilde{\mathbf{T}}$ is the summation of $\mathbf{B'}_{i,j}^{B+1}\mathbf{W}\mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I})$ for each $\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}$. If $\mathbf{W}$ is chosen as above, $\mathbf{B'}_{i,j}^{B+1}\mathbf{W}$ can be written as $\mathbf{B'}_{i+2^{B-1},j+2^{B-1}}^{B}$. Each pixel in $\tilde{\mathbf{T}}\mathcal{I}$ is now covered by just one transformed block. No range block overlapping occurs in this case. Therefore the range pool $\tilde{\mathsf{R}}$ can be rewritten to have non-overlapping range blocks. The size of these blocks is $2^B \times 2^B$ except those lying on the boundaries. As the weights of the boundary range blocks are modified, these boundary blocks are larger than $2^B \times 2^B$. They extend their boundaries to the image boundaries. The range blocks at the four corners are $(\frac{3}{2}\times 2^B)\times(\frac{3}{2}\times 2^B)$. The other boundary range blocks are $2^B \times (\frac{3}{2}\times 2^B)$.*

Therefore it is the transformation of the PIFS with $2^B \times 2^B$ range blocks except those lying on the boundaries. □

Other than the PIFS, the overlapping PIFS proposed in [43] can also be seen as a special case of the LPIFS. These overlapping range blocks are obtained by extending the non-overlapping range blocks of the PIFS by one pixel at the four sides. These overlapping range blocks are easily formulated in our range pool $\tilde{\mathsf{R}}$ if the weights $w_{u,v}$ are chosen as

$$w_{u,v} = \begin{cases} 1 & \text{if} \quad 2^{B-1} < u,v < 3{\times}2^{B-1} - 1, \\ 0.5 & \text{if} \quad u \in \{2^{B-1}-1, 2^{B-1}, 3{\times}2^{B-1}-1, 3{\times}2^{B-1}\} \text{ and} \\ & \quad\; 2^{B-1} < v < 3{\times}2^{B-1} - 1, \\ 0.5 & \text{if} \quad v \in \{2^{B-1}-1, 2^{B-1}, 3{\times}2^{B-1}-1, 3{\times}2^{B-1}\} \text{ and} \\ & \quad\; 2^{B-1} < u < 3{\times}2^{B-1} - 1, \\ 0.25 & \text{if} \quad u,v \in \{2^{B-1}-1, 2^{B-1}, 3{\times}2^{B-1}-1, 3{\times}2^{B-1}\}, \\ 0 & \text{otherwise,} \end{cases} \tag{4.12}$$

One can find that these weights are obtained by extending the non-zero weights in proposition 4.3 by one pixel at the four sides. Therefore proposition 4.4 is obtained.

Proposition 4.4 *The overlapping PIFS described in [43] is a special case of the LPIFS by defining the weights in* $\mathbf{W}$ *by (4.12).*

Proposition 4.4 tells us that the PIFS is a special case of the LPIFS. Interestingly, the LPIFS can be expressed as the weighted summation of the PIFS. The details are described in proposition 4.5.

Proposition 4.5 *Let* $\mathbf{T}_m : \{\mathcal{I}_i\} \rightarrow \{\mathcal{I}_i\}, m = 0,1,2,3$ *be four transformations of the PIFS defined as follows:*

$$\mathbf{T}_m\mathcal{I} \equiv \sum_{\mathcal{R}_{i,j} \in R_m} \mathbf{B}'^{B+1}_{i,j} \mathbf{G}_{i,j} \mathbf{P}_{i,j}(\mathcal{I}), \quad \forall \mathcal{I} \in \{\mathcal{I}_i\} \tag{4.13}$$

with their corresponding range pool $R_m, m = 0, 1, 2, 3$, *defined as follows:*

$$\begin{aligned}
\mathsf{R}_1 &\equiv \{\mathcal{R}_{i,j} : \mathcal{R}_{i,j} = \mathbf{B}_{i,j}^{B+1}\mathcal{I} \text{ with } i = 2^{B+1}p, j = 2^{B+1}q\} \\
\mathsf{R}_2 &\equiv \{\mathcal{R}_{i,j} : \mathcal{R}_{i,j} = \mathbf{B}_{i,j}^{B+1}\mathcal{I} \text{ with } i = 2^{B+1}p + 2^B, j = 2^{B+1}q\} \\
\mathsf{R}_3 &\equiv \{\mathcal{R}_{i,j} : \mathcal{R}_{i,j} = \mathbf{B}_{i,j}^{B+1}\mathcal{I} \text{ with } i = 2^{B+1}p, j = 2^{B+1}q + 2^B\} \\
\mathsf{R}_4 &\equiv \{\mathcal{R}_{i,j} : \mathcal{R}_{i,j} = \mathbf{B}_{i,j}^{B+1}\mathcal{I} \text{ with } i = 2^{B+1}p + 2^B, j = 2^{B+1}q + 2^B\} \qquad (4.14)
\end{aligned}$$

where p, q *are non-negative integers with* $0 \le i, j < 2^N$. *All* R_m *contain range blocks of size* $2^{B+1} \times 2^{B+1}$ *and* $\mathbf{T}_m$ *share the same library* D *in LPIFS.* $\mathbf{P}_{i,j}$ *and* $\mathbf{G}_{i,j}$ *for each range block are defined in the same manner as (4.6) or (4.7) in LPIFS. Then* $\tilde{\mathbf{T}}$ *can be expressed as:*

$$\tilde{\mathbf{T}}\mathcal{I} \equiv \sum_{m=0}^{3} \left[\left(\sum_{\mathcal{R}_{i,j} \in \mathsf{R}_m} \mathbf{B}'^{B+1}_{i,j} \mathbf{W} \mathbf{B}^{B+1}_{i,j} \right) \mathbf{T}_m \mathcal{I} \right]. \qquad (4.15)$$

Proof: *Obviously,* $\tilde{\mathsf{R}}$ *can be partitioned into four disjoint subsets* $\mathsf{R}_m, m = 0, 1, 2, 3$ *in (4.14), i.e.,* $\tilde{\mathsf{R}} \equiv \cup_{m=0}^{3} \mathsf{R}_m$. *From (4.8),*

$$\begin{aligned}
\tilde{\mathbf{T}}\mathcal{I} &= \sum_{\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}} \mathbf{B}'^{B+1}_{i,j} \mathbf{W} \mathbf{G}_{i,j} \mathbf{P}_{i,j}(\mathcal{I}) \\
&= \sum_{m=0}^{3} \sum_{\mathcal{R}_{i,j} \in \mathsf{R}_m} \mathbf{B}'^{B+1}_{i,j} \mathbf{W} \mathbf{G}_{i,j} \mathbf{P}_{i,j}(\mathcal{I}) \\
&= \sum_{m=0}^{3} \sum_{\mathcal{R}_{i,j} \in \mathsf{R}_m} \mathbf{B}'^{B+1}_{i,j} \mathbf{W} \mathbf{B}^{B+1}_{i,j} \mathbf{B}'^{B+1}_{i,j} \mathbf{G}_{i,j} \mathbf{P}_{i,j}(\mathcal{I}) \\
&= \sum_{m=0}^{3} \left[\left(\sum_{\mathcal{R}_{i,j} \in \mathsf{R}_m} \mathbf{B}'^{B+1}_{i,j} \mathbf{W} \mathbf{B}^{B+1}_{i,j} \right) \left(\sum_{\mathcal{R}_{i,j} \in \mathsf{R}_m} \mathbf{B}'^{B+1}_{i,j} \mathbf{G}_{i,j} \mathbf{P}_{i,j}(\mathcal{I}) \right) \right] \\
&= \sum_{m=0}^{3} \left[\left(\sum_{\mathcal{R}_{i,j} \in \mathsf{R}_m} \mathbf{B}'^{B+1}_{i,j} \mathbf{W} \mathbf{B}^{B+1}_{i,j} \right) \mathbf{T}_m \mathcal{I} \right]. \quad \square
\end{aligned}$$

4.2 Attractor Coding using the LPIFS

In the last section, the formulation of the LPIFS is introduced. Based on the LPIFS, the task of the attractor coding is to parameterize the LPIFS such that its fixed point

is close to the original image. Theorem 2.2 tells us that we can minimize the distance between the original image and its collage, i.e., $d(\tilde{\mathbf{T}}\mathcal{I}, \mathcal{I})$ in order to give a fixed point close to the original. In the PIFS, the range blocks are disjoint and so the optimum parameters on each range block can be found independent of the other parts of an image. However, in the proposed LPIFS, the range blocks are no longer disjoint. The question is whether the independent collage minimization in PIFS can give the closest collage to the original image in the case of LPIFS. The answer is no.

The problem comes from the weighting operator $\mathbf{W}$. Collage minimization over each range block produces error for several reasons. The domain block pool may be unable to produce perfect match for a range block. Even if such block is found, quantization of the transformation parameters sacrifices the optimality of the parameters found and thus gives quantization noise. If we simply apply the independent processing of range blocks in the LPIFS, the weighting operator $\mathbf{W}$ mixes up the errors of the adjacent blocks on the overlapping regions [43]. Thus the weighting operation further increases the collage distance. Therefore the ordinary encoding procedure cannot produce the closest collage in this case and the decoded image may suffer from severe blurring of the image details.

Algorithm 2 Proposed Attractor Coding Algorithm using the LPIFS

for each $\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}$ **do**
  Compute $\mathbf{W}\tilde{\mathcal{R}}_{i,j}$
  Compute $\mathbf{AW}\mathcal{R}_{i,j}$
  $\min_{s_{i,j}, o_{i,j}, \mathbf{I}_{i,j}, \mathcal{D}_{k,l}} d(\mathbf{WG}_{i,j}\mathbf{P}_{i,j}(\mathcal{I}), \mathbf{AW}\mathcal{R}_{i,j})$
  store $s_{i,j}, o_{i,j}, \mathbf{I}_{i,j}, \mathbf{P}_{i,j}$
end for

In the proposed attractor coding using LPIFS, the weights in $\mathbf{W}$ are carefully chosen to provide smooth overlapping. The proposed algorithm processes each block of the image from left to right and from top to bottom just like the conventional case. For each range block $\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}$ the preprocessing operator $\mathbf{A}$ is applied first to modify the given block to take the errors of its adjacent blocks into account. No large system

of equations adopted in [20] is needed since the proposed preprocessing decouples the system of equations into a number of independent equations. Each range block can be processed independently now. Then, the weighted collage distance over the aliasing-eliminated block is minimized by tuning the parameters of the affine transformation $\mathbf{G}$, i.e.,

$$\min_{s_{i,j},o_{i,j},\mathbf{I}_{i,j},\mathcal{D}_{k,l}} d(\mathbf{W}\mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I}), \mathbf{A}\mathbf{W}\mathcal{R}_{i,j}). \tag{4.16}$$

These parameters $(s_{i,j}, o_{i,j}, \mathbf{I}_{i,j}, \mathcal{D}_{k,l})$ constitute the image code in the proposed method. Weighted distance is used in the minimization because the block pixels contribute in different weights in approximating the range block. Therefore, using weighted distance can give a weighted optimal solution. In the decoder all parameters are retrieved from the code to reconstruct the LPIFS. The decoded image is the fixed point obtained by recursively applying the transformation $\tilde{\mathbf{T}}$ on any initial image. The proposed attractor coding algorithm is listed in algorithm 2. In the following subsections, the choice of weighting operator $\mathbf{W}$, range block preprocessing $\mathbf{A}$ and the decoder convergence are clearly defined and analyzed.

4.2.1 Choice of Weighting Operator

$\mathbf{W}$ controls the amount of contributes of different transformed blocks to a pixel. A good choice of weights $w_{u,v}$ is crucial to the overall performance of LPIFS. The weighting operator should be optimized for the nature of general images. There are altogether $2^{B+1} \times 2^{B+1}$ number of $w_{u,v}$ in $\mathbf{W}$. It is difficult to tune the optimized values of all weights $w_{u,v}$. Instead, our approach is to deduce all weights which are sufficiently good in image coding. Here we list several properties for a good choice of $w_{u,v}$ should possess:

Property 1: $\mathbf{W}$ should provide smooth transition from one range block to another. In other words, $\mathbf{W}$ works like a window over a range block providing smooth overlapping for several blocks over the same region. Therefore the elements of the matrix $\mathbf{W}\mathbf{U}^{B+1}$

should be diminishing gradually from the center towards the boundaries:

$$w_{u,v} \leq w_{u,v+1}, \quad 0 \leq u < 2^{B+1}, \quad 0 \leq v < 2^B - 1,$$
$$w_{u,v} \leq w_{u+1,v}, \quad 0 \leq v < 2^{B+1}, \quad 0 \leq u < 2^B - 1.$$

Property 2: Each $w_{u,v}$ represents the weight of a pixel in the transformed domain block towards an image location. It is natural to assign non-negative weights for all pixels in a block, i.e, $w_{u,v} \geq 0$. Consider the case that the weights satisfy property 1 with some are negative. That means there is a gradual transition from the central positive weights towards the boundary negative weights. In this case, the absolute values of $w_{u,v}$ decrease from the center of $\mathbf{WU}^{B+1}$ and then increase at the boundaries. That means in the weighted collage minimization (4.16) less weights are assigned to some pixels that do not locate at the block boundaries. The weighted minimization may produce a domain block that does not fit very well at the lightly-weighted pixels which do not locate at the block boundaries. Ringing errors may thus occur in the image. Therefore all $w_{u,v}$ are chosen to be non-negative. Together with the constraints (4.2), $|w_{u,v}| \leq 1$ for all $0 \leq u, v < 2^{B+1}$.

Property 3: $w_{u,v}$ is symmetric in the sense that the matrix $\mathbf{WU}^{B+1}$ is invariant up to reflections and rotations. The image $\tilde{\mathbf{T}}\mathcal{I}$ is formed by many weighted copies of transformed domain blocks. If $\mathbf{W}$ is not symmetric, the image may contain some periodic fluctuations which is highly disturbing to HVS. Therefore symmetry of weights is important in applying the LPIFS to image coding. If the matrix $\mathbf{WU}^{B+1}$ is invariant up to reflections and rotations, then

$$\begin{aligned} & w_{u,v} = w_{2^{B+1}-u,v} = w_{u,2^{B+1}-v} = w_{2^{B+1}-u,2^{B+1}-v} \\ = \; & w_{v,u} = w_{2^{B+1}-v,u} = w_{v,2^{B+1}-u} = w_{2^{B+1}-v,2^{B+1}-u}, \qquad \forall\, 0 \leq u, v < 2^B. \quad (4.17) \end{aligned}$$

Thus the number of parameters in specifying $\mathbf{W}$ is reduced to $\frac{2^{B+1} \times 2^{B+1}}{8}$.

Property 4: $\mathbf{W}$ provides smooth transition from one range block to another. Therefore, the weights at the boundaries of $\mathbf{W}$ should be small in magnitudes, i.e., $w_{u,v} \approx 0$

at the boundary.

There are many choices of $\mathbf{W}$ that fulfill these four properties. In our studies, the cosine weights is proposed to be used for the weights $w_{u,v}$.

Definition 4.5 *(Cosine weights) Cosine weights are a $2^{B+1}\times 2^{B+1}$ matrix with its elements $w_{u,v}$ defined as follows:*

$$w_{u,v} = \left(1 - \frac{\cos\frac{2u+1}{2^{B+1}}\pi}{\cos\frac{\pi}{2^{B+1}}}\right)\left(1 - \frac{\cos\frac{2v+1}{2^{B+1}}\pi}{\cos\frac{\pi}{2^{B+1}}}\right), \quad \forall\ 0 \le u, v < 2^B \tag{4.18}$$

with

$$w_{u,v} = w_{2^{B+1}-u,v} = w_{u,2^{B+1}-v} = w_{2^{B+1}-u,2^{B+1}-v}, \quad \forall\ 0 \le u, v < 2^B.$$

The cosine weights in definition (4.5) satisfies all four desirable properties. It is constructed by using the linear combination of a real values and the first basis of the 2^{B+1} LOT [33]. (see fig. 4.3) The derivation and the reasons for this choice are discussed in the following paragraphs.

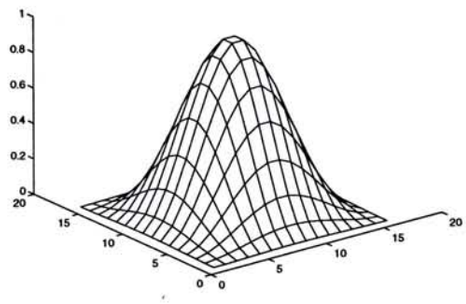


Figure 4.3: The plot of 16×16 weights $w_{i,j}$.

The reason for this choice becomes clear when the block matching is examined in terms of inner product. All range blocks, $\mathbf{U}^{B+1}$, and the decimated domain blocks are seen as elements in an inner product space of dimension $2^{B+1}\times 2^{B+1}$. The weighted collage minimization in (4.16) can be seen as the approximation of each range block by the linear combination of the decimated domain block $\mathbf{ODI}_{i,j}(\mathcal{D}_{k,l})$ and the matrix $\mathbf{U}^{B+1}$. Without loss of generality, assuming that the orthogonalized affine transformation $\mathbf{G}_{i,j}$

in (4.7) is used, theorem 2.3 tells us that a range block $\tilde{\mathcal{R}}_{i,j}$ is best approximated by the linear span of the domain block and $\mathbf{U}^{B+1}$ when the error is orthogonal to the approximation, i.e.,

$$< \mathbf{W}\tilde{\mathcal{R}}_{i,j} - \mathbf{WG}_{i,j}(\mathcal{D}_{k,l}), \mathbf{WG}_{i,j}(\mathcal{D}_{k,l}) >= 0$$

where $\mathbf{W}\tilde{\mathcal{R}}_{i,j} - \mathbf{WG}_{i,j}(\mathcal{D}_{k,l})$ is the approximation error. A good choice of weights $w_{i,j}$ must minimize this error for most range blocks of general images. In other words, nearly all energies of most range blocks should be concentrated in the $\mathbf{WG}_{i,j}(\mathcal{D}_{k,l})$ such that its orthogonal complement, i.e., the error, is minimized. Expanding $\mathbf{WG}_{i,j}(\mathcal{D}_{k,l})$, we get

$$\mathbf{WG}_{i,j}(\mathcal{D}_{k,l}) = s_{i,j} \times \mathbf{WODI}_{i,j}(\mathcal{D}_{k,l}) + o_{i,j} \times \mathbf{WU}^{B+1}.$$

This is only the approximation of just one weighted range block only. Every sub-block $\mathbf{B}^B_{i,j}\mathcal{I}$ (with $i = 2^B p, j = 2^B q, 0 \leq p, q < 2^{N-B} - 1$) in $\mathcal{I}$ is covered by four range blocks. That means $\mathbf{B}^B_{i,j}\mathcal{I}$ is approximated by four weighted decimated domain blocks and weighted $\mathbf{U}^B$:

$$\mathbf{B}^B_{i,j}\mathcal{I} \approx \sum_{m,n \in \{0,2^B\}} \left(s_{i-m,j-n}\mathbf{B}^B_{m,n}\mathbf{WODI}_{i-m,j-n}(\mathcal{D}_{k,l}) + o_{i-m,j-n}\mathbf{B}^B_{m,n}\mathbf{WU}^{B+1} \right).$$

Therefore a range block is no longer just covered by a DC component and a decimated domain block that is found to be statistically incompatible to general image characteristics [51]. The first term in the summation depends on the domain block chosen. The domain block is chosen from the domain pool and such a large number of different domain blocks are difficult to be characterized by an image model. It is expected that finding the optimized weights of $\mathbf{W}$ such that most of the energy are contained in $\mathbf{WODI}_{i-m,j-n}(\mathcal{D}_{k,l})$ is a very complicated task. On the other hand, the second term depends on the weights only. Our aim is to find the weights such that nearly all energies of a block concentrate in the four blocks $\mathbf{WU}^{B+1}$.

It is well known that Karhunen-Loève transform (KLT) is the optimal transform in the sense that KLT concentrates maximum energies into the first few bases. In

processing digital images, discrete cosine transform (DCT) is a good energy-compacting transform to replace KLT in most images [28]. Most energies are concentrated in the first two bases of DCT and all other bases are nearly negligible. If the weights in $\mathbf{W}$ are designed to be the summation of the first two bases of DCT, we expect that most of energies of a block can be concentrated in $\mathbf{W}\mathbf{U}^{B+1}$. Therefore, the first basis of the lapped orthogonal transform (LOT) [33] of length 2^{B+1} that is the linear combination of the first two bases of DCT of length 2^B is chosen to construct $\mathbf{W}$.

We are now going to derive the cosine weights in definition 4.5. The weights $w_{u,v}$ with $0 \leq u, v < 2^B$ are derived here and the remaining are obtained by the symmetry described in property 3. Starting from the first basis of LOT of length 2^{B+1}:

$$\sqrt{\frac{1}{2^B}} - \sqrt{\frac{1}{2^{B-1}}} \cos \frac{2u+1}{2^{B+1}} \pi, \quad \forall\, 0 \leq u < 2^B.$$

It is only left half of the basis of length 2^B. The right half is the mirror image of this half. The coefficients at the two boundaries are negative that is not desirable for the weighting operator $\mathbf{W}$ by property 2. Therefore the entire basis is shifted by a constant such that all coefficients are non-negative. However how much should the basis shift? Property 4 tells us to choose the amount of shifting in the way that the boundary coefficients to be zero. As the boundary coefficient ($u = 0$) has the least magnitude, the shifted basis is obtained by deducting each coefficient by the boundary coefficient ($u = 0$):

$$\sqrt{\frac{1}{2^B}} - \sqrt{\frac{1}{2^{B-1}}} \cos \frac{2u+1}{2^{B+1}} \pi - \left(\sqrt{\frac{1}{2^B}} - \sqrt{\frac{1}{2^{B-1}}} \cos \frac{1}{2^{B+1}} \pi \right), \quad \forall\, 0 \leq u < 2^B,$$

and the new shifted basis is

$$\sqrt{\frac{1}{2^{B-1}}} \left(\cos \frac{\pi}{2^{B+1}} - \cos \frac{2u+1}{2^{B+1}} \pi \right), \quad \forall\, 0 \leq u < 2^B.$$

Therefore the weights $w_{u,v}$ are the product of two separable one-dimensional basis:

$$\frac{1}{2^{B-1}} \left(\cos \frac{\pi}{2^{B+1}} - \cos \frac{2u+1}{2^{B+1}} \pi \right) \left(\cos \frac{\pi}{2^{B+1}} - \cos \frac{2v+1}{2^{B+1}} \pi \right), \quad \forall\, 0 \leq u, v < 2^B.$$

The cosine weights are nearly obtained. However the constraint (4.2) is not fulfilled in this stage. All weights are multiplied by the same constant in the way that the resultant $w_{u,v}$ satisfy the constraint (4.2) and the cosine weights in definition 4.3 are obtained. Experiments have been performed that the cosine weights is undoubtedly better than the rectangular window.

4.2.2 Range Block Preprocessing

The collage minimization takes into account of the error of the previous processed blocks by modifying the current range block before the collage minimization. We term such a process as preprocessing. No large system of equations are needed since the proposed preprocessing decouples the system of quadratic equations into a number of independent equations. Each range block can be processed independently with the aliasing error eliminated. The cause of aliasing is analyzed first followed by the formulation of the aliasing elimination $\mathbf{A}$.

Consider a range block $\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}$ within an image $\mathcal{I}$. This block laps with its three adjacent range blocks, namely $\tilde{\mathcal{R}}_{i-2^B,j-2^B}$, $\tilde{\mathcal{R}}_{i-2^B,j}$ and $\tilde{\mathcal{R}}_{i,j-2^B}$, in the proposed LPIFS. As the proposed LPIFS processes each range block in the lexicographical order, the other three range blocks have already been processed when the proposed algorithm starts to process $\tilde{\mathcal{R}}_{i,j}$. Consider the four sub-blocks of size $2^B \times 2^B$ within $\tilde{\mathcal{R}}_{i,j}$. The three adjacent range blocks overlap with $\tilde{\mathcal{R}}_{i,j}$ on the sub-blocks $\mathbf{B}^B_{i,j}\mathcal{I}$, $\mathbf{B}^B_{i+2^B,j}\mathcal{I}$ and $\mathbf{B}^B_{i,j+2^B}\mathcal{I}$. Large error on these three blocks occurs if the transformation parameters of $\tilde{\mathcal{R}}_{i,j}$ are not found properly. The overall error $\tilde{e}_{i,j}$ between the original image and its

collage on the sub-block $\mathbf{B}_{i,j}^{B}\mathcal{I}$ is given by

$$\begin{aligned}\tilde{e}_{i,j} = \mathbf{B}_{i,j}^{B}\mathcal{I} \; &- \; \mathbf{B}_{0,0}^{B}\mathbf{W}\mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I}) \\ &- \; \mathbf{B}_{0,2^B}^{B}\mathbf{W}\mathbf{G}_{i,j-2^B}\mathbf{P}_{i,j-2^B}(\mathcal{I}) \\ &- \; \mathbf{B}_{2^B,0}^{B}\mathbf{W}\mathbf{G}_{i-2^B,j}\mathbf{P}_{i-2^B,j}(\mathcal{I}) \\ &- \; \mathbf{B}_{2^B,2^B}^{B}\mathbf{W}\mathbf{G}_{i-2^B,j-2^B}\mathbf{P}_{i-2^B,j-2^B}(\mathcal{I}) \end{aligned} \tag{4.19}$$

or in more compact form:

$$\tilde{e}_{i,j} = \mathbf{B}_{i,j}^{B}\mathcal{I} - \sum_{k,l\in\{0,2^B\}} \mathbf{B}_{k,l}^{B}\mathbf{W}\mathbf{G}_{i-k,j-l}\mathbf{P}_{i-k,j-l}(\mathcal{I}). \tag{4.20}$$

Similarly, define $\tilde{e}_{i,j+2^B}$, $\tilde{e}_{i+2^B,j}$ and $\tilde{e}_{i+2^B,j+2^B}$ as the error on the sub-blocks $\mathbf{B}_{i,j}^{B}\mathcal{I}$, $\mathbf{B}_{i+2^B,j}^{B}\mathcal{I}$ and $\mathbf{B}_{i,j+2^B}^{B}\mathcal{I}$ within $\tilde{\mathcal{R}}_{i,j}$ respectively:

$$\tilde{e}_{i,j+2^B} = \mathbf{B}_{i,j}^{B}\mathcal{I} - \sum_{k,l\in\{0,2^B\}} \mathbf{B}_{k,l}^{B}\mathbf{W}\mathbf{G}_{i-k,j+2^B-l}\mathbf{P}_{i-k,j+2^B-l}(\mathcal{I}), \tag{4.21}$$

$$\tilde{e}_{i+2^B,j} = \mathbf{B}_{i+2^B,j}^{B}\mathcal{I} - \sum_{k,l\in\{0,2^B\}} \mathbf{B}_{k,l}^{B}\mathbf{W}\mathbf{G}_{i+2^B-k,j-l}\mathbf{P}_{i+2^B-k,j-l}(\mathcal{I}), \tag{4.22}$$

$$\tilde{e}_{i+2^B,j+2^B} = \mathbf{B}_{i+2^B,j+2^B}^{B}\mathcal{I} - \sum_{k,l\in\{0,2^B\}} \mathbf{B}_{k,l}^{B}\mathbf{W}\mathbf{G}_{i+2^B-k,j+2^B-l}\mathbf{P}_{i+2^B-k,j+2^B-l}(\mathcal{I}) \tag{4.23}$$

and the optimal encoder should minimize these four errors on every $\tilde{\mathcal{R}}_{i,j}$, i.e.,

$$\min \sum_{k,l\in\{0,2^B\}} \|\tilde{e}_{i+k,j+l}\|^2. \tag{4.24}$$

If no preprocessing is applied, only direct weighted minimization is used to find the block parameters in the proposed LPIFS, i.e., $\min_{s_{i,j},o_{i,j},\mathbf{I}_{i,j},\mathcal{D}_{k,l}} d(\mathbf{W}\mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I}), \mathbf{W}\mathcal{R}_{i,j})$. For the range block $\tilde{\mathcal{R}}_{i,j}$ the distance between the collage and the original image on $\tilde{\mathcal{R}}_{i,j}$ is minimized. If we let $e_{i,j}$ be the error on the sub-block $\mathbf{B}_{i,j}^{B}\mathcal{I}$ produced by the transformation for $\tilde{\mathcal{R}}_{i,j}$, i.e,

$$e_{i,j} \equiv \mathbf{B}_{0,0}^{B}\mathbf{W}\tilde{\mathcal{R}}_{i,j} - \mathbf{B}_{0,0}^{B}\mathbf{W}\mathbf{G}_{i,j}\mathbf{P}_{i,j}(\mathcal{I}). \tag{4.25}$$

Similarly, the errors for the other three range blocks covering $\mathbf{B}^B_{i,j}\mathcal{I}$ are:

$$e_{i,j-2^B} \equiv \mathbf{B}^B_{0,2^B}\mathbf{W}\tilde{\mathcal{R}}_{i,j-2^B} - \mathbf{B}^B_{0,2^B}\mathbf{W}\mathbf{G}_{i,j-2^B}\mathbf{P}_{i,j-2^B}(\mathcal{I}), \tag{4.26}$$

$$e_{i-2^B,j} \equiv \mathbf{B}^B_{2^B,0}\mathbf{W}\tilde{\mathcal{R}}_{i-2^B,j} - \mathbf{B}^B_{2^B,0}\mathbf{W}\mathbf{G}_{i-2^B,j}\mathbf{P}_{i-2^B,j}(\mathcal{I}), \tag{4.27}$$

$$e_{i-2^B,j-2^B} \equiv \mathbf{B}^B_{2^B,2^B}\mathbf{W}\tilde{\mathcal{R}}_{i-2^B,j-2^B} - \mathbf{B}^B_{2^B,2^B}\mathbf{W}\mathbf{G}_{i-2^B,j-2^B}\mathbf{P}_{i-2^B,j-2^B}(\mathcal{I}). \tag{4.28}$$

Then, we can work out the upper bound for (4.19):

$$\begin{aligned}
\|\tilde{e}_{i,j}\| &= \|\mathbf{B}^B_{i,j}\mathcal{I} - \sum_{k,l\in\{0,2^B\}} \mathbf{B}^B_{k,l}\mathbf{W}\mathbf{G}_{i-k,j-l}\mathbf{P}_{i-k,j-l}(\mathcal{I})\| \\
&= \| \sum_{k,l\in\{0,2^B\}} \mathbf{B}^B_{k,l}\mathbf{W}\mathbf{B}'^B_{k,l}\mathbf{B}^B_{i,j}\mathcal{I} - \sum_{k,l\in\{0,2^B\}} \mathbf{B}^B_{k,l}\mathbf{W}\mathbf{G}_{i-k,j-l}\mathbf{P}_{i-k,j-l}(\mathcal{I})\| \\
&= \| \sum_{k,l\in\{0,2^B\}} \mathbf{B}^B_{k,l}\mathbf{W}\tilde{\mathcal{R}}_{i-k,j-l} - \sum_{k,l\in\{0,2^B\}} \mathbf{B}^B_{k,l}\mathbf{W}\mathbf{G}_{i-k,j-l}\mathbf{P}_{i-k,j-l}(\mathcal{I})\| \\
&\leq \sum_{k,l\in\{0,2^B\}} \| \left(\mathbf{B}^B_{k,l}\mathbf{W}\tilde{\mathcal{R}}_{i-k,j-l} - \mathbf{B}^B_{k,l}\mathbf{W}\mathbf{G}_{i-k,j-l}\mathbf{P}_{i-k,j-l}(\mathcal{I})\right) \| \\
&= \|e_{i,j}\| + \|e_{i,j-2^B}\| + \|e_{i-2^B,j}\| + \|e_{i-2^B,j-2^B}\| \\
&\equiv \sum_{k,l\in\{0,2^B\}} \|e_{i-l,j-k}\|.
\end{aligned}$$

Equality holds only when these four errors are parallel, i.e., there exists $k_1, k_2, k_3 \in \mathbb{R}$ such that

$$e_{i,j} = k_1 e_{i,j-2^B} = k_2 e_{i-2^B,j} = k_3 e_{i-2^B,j-2^B}. \tag{4.29}$$

Thus proposition 4.6 follows:

Proposition 4.6 *If only weighted collage minimization is applied without any range block preprocessing, the overall error $\tilde{e}_{i,j}$ between the original image and its collage on the sub-block $\mathbf{B}^B_{i,j}\mathcal{I}$ is upper bounded by*

$$\|\tilde{e}_{i,j}\| \leq \sum_{k,l\in\{0,2^B\}} \|e_{i-l,j-k}\|$$

with $e_{i-l,j-k}$, $k,l \in \{0, 2^B\}$, defined in (4.25) to (4.28).

Proposition 4.6 gives the account of the problem when the collage is found only by the weighted collage minimization. The collage produced from the direct minimization

on individual range block can only give the upper error bound that is generally not tight enough. The upper bound is not tight because the condition in (4.29) is not guaranteed and thus not satisfied in most of the cases. It means that the encoder used in existing PIFS is not the optimal one for the LPIFS. Some modifications on the encoder must be made to achieve a closer collage. One obvious way is to carry out the minimization of (4.24) directly. However, it depends on four range blocks and thus all these errors on the whole image becomes a system of quadratic equations [20]. The equations depend on the others and make the solutions difficult to be found in a simple manner. Instead, a technique called range block preprocessing is proposed to solve this problem.

Preprocessing is proposed to achieve a tighter bound of the error in (4.19). Given an range block $\tilde{\mathcal{R}}_{i,j}$, the following three modifications on the three sub-blocks of $\tilde{\mathcal{R}}_{i,j}$ are performed:

$$\begin{aligned}
\mathbf{B}^B_{0,0}\mathbf{W}\tilde{\mathcal{R}}_{i,j} \leftarrow \quad & \mathbf{B}^B_{0,0}\tilde{\mathcal{R}}_{i,j} \quad -\mathbf{B}^B_{0,2^B}\mathbf{W}\mathbf{G}_{i,j-2^B}\mathbf{P}_{i,j-2^B}(\mathcal{I}) \\
& \qquad\qquad\quad -\mathbf{B}^B_{2^B,0}\mathbf{W}\mathbf{G}_{i-2^B,j}\mathbf{P}_{i-2^B,j}(\mathcal{I}) \\
& \qquad\qquad\quad -\mathbf{B}^B_{2^B,2^B}\mathbf{W}\mathbf{G}_{i-2^B,j-2^B}\mathbf{P}_{i-2^B,j-2^B}(\mathcal{I}) \\
\mathbf{B}^B_{0,2^B}\mathbf{W}\tilde{\mathcal{R}}_{i,j} \leftarrow \quad & \mathbf{B}^B_{0,2^B}\mathbf{W}\tilde{\mathcal{R}}_{i,j} \quad -\mathbf{B}^B_{2^B,2^B}\mathbf{W}\mathbf{G}_{i-2^B,j}\mathbf{P}_{i-2^B,j}(\mathcal{I}) \\
\mathbf{B}^B_{2^B,0}\mathbf{W}\tilde{\mathcal{R}}_{i,j} \leftarrow \quad & \mathbf{B}^B_{2^B,0}\mathbf{W}\tilde{\mathcal{R}}_{i,j} \quad -\mathbf{B}^B_{2^B,2^B}\mathbf{W}\mathbf{G}_{i,j-2^B}\mathbf{P}_{i,j-2^B}(\mathcal{I})
\end{aligned}$$

where $\leftarrow$ denotes the assignment. These three modifications on $\mathbf{W}\tilde{\mathcal{R}}_{i,j}$ can be expressed compactly by defining an operator $\mathbf{A}$ on the weighted range block.

Definition 4.6 *(Range block preprocessing operator)* $\mathbf{A} : \mathbb{R}^{2^{B+1}\times 2^{B+1}} \rightarrow \mathbb{R}^{2^{B+1}\times 2^{B+1}}$:

$$\begin{aligned}
\mathbf{A}\mathbf{W}\tilde{\mathcal{R}}_{i,j} \;=\; & \mathbf{B}'^B_{0,0}\left(\mathbf{B}^B_{0,0}\tilde{\mathcal{R}}_{i,j} - \sum_{\substack{k,l\in\{0,2^B\}\\ \{k,l\}\neq\{0,0\}}} \mathbf{B}^B_{k,l}\mathbf{W}\mathbf{G}_{i-k,j-l}\mathbf{P}_{i-k,j-l}(\mathcal{I})\right) \\
+ \; & \mathbf{B}'^B_{0,2^B}\left(\mathbf{B}^B_{0,2^B}\mathbf{W}\tilde{\mathcal{R}}_{i,j} - \mathbf{B}^B_{2^B,2^B}\mathbf{W}\mathbf{G}_{i-2^B,j}\mathbf{P}_{i-2^B,j}(\mathcal{I})\right) \\
+ \; & \mathbf{B}'^B_{2^B,0}\left(\mathbf{B}^B_{2^B,0}\mathbf{W}\tilde{\mathcal{R}}_{i,j} - \mathbf{B}^B_{2^B,2^B}\mathbf{W}\mathbf{G}_{i,j-2^B}\mathbf{P}_{i,j-2^B}(\mathcal{I})\right) \\
+ \; & \mathbf{B}'^B_{2^B,2^B}\mathbf{B}^B_{2^B,2^B}\mathbf{W}\tilde{\mathcal{R}}_{i,j} \qquad (4.30)
\end{aligned}$$

where $\mathbf{W}\tilde{\mathcal{R}}_{i,j} \in \mathbb{R}^{2^{B+1}\times 2^{B+1}}$ *and the range of* $\mathbf{B}'^{B}_{k,l}$, $k,l \in \{0, 2^B\}$ *is modified to* $\mathbb{R}^{2^{B+1}\times 2^{B+1}}$ *(not* $\mathbb{R}^{2^N\times 2^N}$ *as usual).*

By preprocessing the weighted range block by $\mathbf{A}$ before the collage minimization, the weighted minimization becomes (4.16). In plain words, the preprocessing $\mathbf{A}$ takes the errors accumulated from the previous blocks into the present minimization. Thus, the collage distance being minimized will not suffer from mixing up with the errors produced by the previous blocks.

4.2.3 Decoder Convergence Analysis

In the proposed coding method, the decoded image is the fixed point of the transformation $\tilde{\mathbf{T}}$ and it is found by iterative method. Thus it is natural to ask the question whether $\tilde{\mathbf{T}}$ is a contraction. If $\mathbf{G}_{i,j}$ is defined by (4.6), the sufficient condition for $\tilde{\mathbf{T}}$ to be a contraction is very similar to that of the conventional PIFS.

Proposition 4.7 $\tilde{\mathbf{T}} : \{\mathcal{I}_i\} \rightarrow \{\mathcal{I}_i\}$ *is a strictly contractive transformation respect to the supremum metric if* $\mathbf{G}_{i,j}$ *is defined by (4.6) and* $|s_{i,j}| < 1$ *for all* $\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}$. *(compared with theorem 2.4)*

Proof: *If the supremum metric is used, for any* $\mathcal{I}_1, \mathcal{I}_2 \in \{\mathcal{I}_i\}$,

$$\begin{aligned} d(\tilde{\mathbf{T}}\mathcal{I}_1, \tilde{\mathbf{T}}\mathcal{I}_2) &\equiv \|\tilde{\mathbf{T}}\mathcal{I}_1 - \tilde{\mathbf{T}}\mathcal{I}_2\| \\ &= \|\tilde{\mathbf{T}}_L(\mathcal{I}_1 - \mathcal{I}_2)\| \\ &= \max_{u,v} |i_{u,v}^{\tilde{\mathbf{T}}_L}| \end{aligned}$$

where $i_{u,v}^{\tilde{\mathbf{T}}_L} \in \tilde{\mathbf{T}}_L$ *and* $\tilde{\mathbf{T}}_L$ *is the linear part of* $\tilde{\mathbf{T}}$. *If* $\mathbf{G}_{i,j}$ *is defined by (4.6), then each* $i_{u,v}^{\tilde{\mathbf{T}}_L} \in \tilde{\mathbf{T}}_L$ *can be written as*

$$i_{u,v}^{\tilde{\mathbf{T}}_L} = \sum_{m,n} \left(w_{m,n} s_{m,n} \frac{\sum_{p_m,q_n} (i^1_{p_m,q_n} - i^2_{p_m,q_n})}{4} \right)$$

for some $w_{m,n}$, $s_{m,n}$ *and* $i^1_{p_m,q_n} - i^2_{p_m,q_n} \in \mathcal{I}_1 - \mathcal{I}_2$. *The summation over* (m,n) *is due to the weighting operator* $\mathbf{W}$ *and the other is due to the averaging and decimation operator* $\mathbf{D}$. *Hence*

$$\begin{aligned}
d(\tilde{\mathbf{T}}\mathcal{I}_1, \tilde{\mathbf{T}}\mathcal{I}_2) &= \max_{u,v} |i^{\tilde{\mathbf{T}}_L}_{u,v}| \\
&= \left| \sum_{m,n} \left(w_{m,n} s_{m,n} \frac{\sum_{p_m,q_n} (i^1_{p_m,q_n} - i^2_{p_m,q_n})}{4} \right) \right| \\
&\leq \left| \sum_{m,n} \left(w_{m,n} \left| s_{m,n} \frac{\sum_{p_m,q_n} (i^1_{p_m,q_n} - i^2_{p_m,q_n})}{4} \right| \right) \right| \\
&\leq \left| \sum_{m,n} w_{m,n} \right| \max_{p_m,q_n} \left| s_{m,n} \frac{\sum_{p_m,q_n} (i^1_{p_m,q_n} - i^2_{p_m,q_n})}{4} \right| \\
&= \max_{m,n} \left| s_{m,n} \frac{\sum_{p_m,q_n} (i^1_{p_m,q_n} - i^2_{p_m,q_n})}{4} \right| \\
&\leq \max_{p,q} \left| \frac{\sum_{p,q} (i^1_{p,q} - i^2_{p,q})}{4} \right| \\
&\leq \max_{p,q} |i^1_{p,q} - i^2_{p,q}| \\
&\equiv d(\mathcal{I}_1, \mathcal{I}_2).
\end{aligned}$$

Hence $\tilde{\mathbf{T}}$ *is a strictly contractive transformation.* □

If the orthogonalized $\mathbf{G}_{i,j}$ is used instead, i.e., (4.6), the contractivity is not easily verified by deducing the contractivity condition in definition 2.1. However, experiments show that convergence of the iterative decoding method is obtained for all cases under our settings.

4.3 Local Domain Block Searching

In the last section, the attractor coding method based on LPIFS is discussed. Just like PIFS, for each range block, a domain block searching is required to find the best fit in encoding. The number of blocks in the domain pool is usually very large. It takes a lot of computations and time in block matching if exhaustive searching is performed. Moreover, many bits are used to represent all domain block locations in such a large

pool. If some domain blocks have higher probabilities to give a good match, fewer bits should be allocated to represent their locations in order to reduce the bits required. They should also have higher priority in matching in order to reduce the encoding time. Then the problem is what kind of searching method and how to represent the block locations are appropriate for attractor coding. This section addresses this problem and a local domain block searching algorithm is devised and analyzed.

Roughly speaking, the proposed searching method starts from the block with the same center of the range block, i.e., the local domain block. If it does not result in a good match, it will search blocks further away from the range block until a right domain block is found. Similar local searching algorithms have been proposed and verified by experiments [4, 5, 22, 47]. The effectiveness of this searching method replies on the fact that the local domain block can fit the range block in most cases. Therefore the encoding time can be shorter and fewer bits are needed to represent the block locations. Barthel *et. al.* suggested that the best block is usually the one overlaps with the range block. Experiments performed by Frigaard *et. al.* suggested the otherwise. The location of the best block is quite random but the local domain block usually can give a good match even though it is not the best one in the pool. In the following section, a theoretical analysis is provided which shows that the domain block overlapping with the range block can usually provide a nearly-best fit.

4.3.1 Theoretical Foundation

The analysis is based on the first-order Gauss-Markov model. Consider a vector $\vec{X} \equiv (x_0, x_1, \ldots, x_{2^{B+1}-1})$ of length 2^{B+1} with each $x_i \in \vec{X}$ being a random variable. Assume $\vec{X}$ satisfies the first-order Gauss-Markov process with correlation coefficient ρ. Based on $\vec{X}$, define another vector $\vec{Y}$ of length 2^B as follows:

$$\vec{Y} \equiv (x_{2^{B-1}}, x_{2^{B-1}+1}, \ldots, x_{3\times 2^{B-1}-1}).$$

Obviously, the elements in $\vec{Y}$ is extracted from the center of $\vec{X}$. Under this settings, $\vec{Y}$ simulates the range block in attractor coding. $\vec{X}$, being longer than $\vec{Y}$, simulates the local domain block. We are now going to see whether the decimated $\vec{X}$ can fit $\vec{Y}$ well up to the affine transformation. Following the encoding procedure of attractor coding, $\vec{X}$ is first decimated by 2 to form $\vec{X}_{\downarrow 2}$ of length 2^B as follows:

$$\vec{X}_{\downarrow 2} \equiv (\frac{x_0 + x_1}{2}, \frac{x_2 + x_3}{2}, \ldots, \frac{x_{2^{B+1}-2} + x_{2^{B+1}-1}}{2}).$$

Then an affine transformation is applied on $\vec{X}_{\downarrow 2}$. The two parameters in affine transformation are tuned such that $\vec{X}_{\downarrow 2}$ is close to $\vec{Y}$. Since each $x_i \in \vec{X}$ is zero mean (by the assumption of the model), the optimal offset should be zero. Therefore only one scaling parameter $s \in \mathbb{R}$ remains in the discussions. Hence $\vec{Y}$ is now approximated by $s\vec{X}_{\downarrow 2}$ and the distance between $\vec{Y}$ and $s\vec{X}_{\downarrow 2}$ is given by

$$\begin{aligned} d(\vec{Y}, s\vec{X}_{\downarrow 2}) &\equiv \mathbf{E}\left((\vec{Y} - s\vec{X}_{\downarrow 2})(\vec{Y} - s\vec{X}_{\downarrow 2})'\right) \\ &= \mathbf{E}\left(Tr\left[(\vec{Y} - s\vec{X}_{\downarrow 2})'(\vec{Y} - s\vec{X}_{\downarrow 2})\right]\right) \\ &= Tr\left[\mathbf{E}(\vec{Y}'\vec{Y})\right] - s \times Tr\left[\mathbf{E}(\vec{Y}'\vec{X}_{\downarrow 2})\right] - s \times Tr\left[\mathbf{E}(\vec{X}'_{\downarrow 2}\vec{Y})\right] + Tr\left[\mathbf{E}(\vec{X}'_{\downarrow 2}\vec{X}_{\downarrow 2})\right] \end{aligned}$$

where $'$ and Tr denote the transpose operator of a vector and the trace of a square matrix respectively. By expanding the elements in each matrix and expressing them in terms of ρ, $d(\vec{Y}, s\vec{X}_{\downarrow 2})$ can be written as follows:

$$\begin{aligned} d(\vec{Y}, s\vec{X}_{\downarrow 2}) &= s2^{B-1}(1+\rho) - 2s \sum_{i=0}^{2^{B-1}-1} \rho^i(1+\rho) + 2^B \\ &= 2^B + s(1+\rho)\left(2^{B-1} - \frac{2(1-\rho^{2^{B-1}-1})}{1-\rho}\right). \end{aligned} \tag{4.31}$$

$d(\vec{Y}, s\vec{X}_{\downarrow 2})$ tends to zero as $\rho \to 1$ when s is chosen to be one. Hence we get the following proposition.

Proposition 4.8 *If $\vec{X} \equiv (x_0, x_1, \ldots, x_{2^{B+1}-1})$ is a vector of length 2^{B+1} satisfying the first-order Gauss-Markov process, then $\vec{X}$ is a good match to the vector $\vec{Y} \equiv (x_{2^{B-1}}, x_{2^{B-1}+1}, \ldots, x_{3\times 2^{B-1}-1})$ up to the decimation by two followed by the affine transformation if the correlation coefficient ρ tends to one.*

4.3.2 Local Block Searching Algorithm

For each $\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}$, the proposed algorithm starts from the local block first and searches the blocks further away from $\tilde{\mathcal{R}}_{i,j}$. To be more precise, the domain pool in definition 4.2 is reformulated as four nested domain pools as follows:

Definition 4.7 *(Nested domain pools) For each* $\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}$,

$$\begin{aligned}
\mathsf{D}_0 &\equiv \{\mathcal{D}_{k,l} : k = i - 2^B, l = j - 2^B\} \\
\mathsf{D}_1 &\equiv \{\mathcal{D}_{k,l} : k = i + 2^B p, l = j + 2^B q, -2 \le p, q \le 2\} \\
\mathsf{D}_2 &\equiv \{\mathcal{D}_{k,l} : k = i + 2^B p, l = j + 2^B q, -8 \le p, q \le 8\} \\
\mathsf{D}_3 &\equiv \{\mathcal{D}_{k,l} : k = 2^B p, l = 2^B q, 0 \le p, q < 2^{N-B-2}\}
\end{aligned}$$

where $\mathcal{D}_{k,l} \equiv \mathbf{B}_{k,l}^{B+2}\mathcal{I}$.

Obviously, $\mathsf{D}_0 \subset \mathsf{D}_1 \subset \mathsf{D}_2 \subset \mathsf{D}_3$ and $\cup_{i=0}^{3}\mathsf{D}_i = \mathsf{D}$. Moreover, the definitions of $\mathsf{D}_i, i = 0, 1, 2, 3$, depend on the position of $\tilde{\mathcal{R}}_{i,j}$.

Rather than exhaustively searching all domain blocks in D, D_0 is searched first, then D_1, D_2 and finally D_3 till an appropriate domain block is found. If all four domain pools are searched for each range block, it is equivalent to simply using D but in different searching order of the domain blocks only. Moreover, when the search pool is enlarged, a better fit may be found in the cost of more bits to represent the block locations. That means there is a trade-off between decoded image quality and compression ability. Not all pools are needed for each range block. A stopping criterion is required to define the desired domain block. Therefore we formulate this problem as the optimization of a set of rate-distortion functions.

Definition 4.8 *(Rate-distortion function) For each* $\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}$, $R_{i,j}(D_{i,j})$ *is the number of bits required for* $\tilde{\mathcal{R}}_{i,j}$ *to achieve the mean square error* $D_{i,j}$.

By defining $R_{i,j}(D_{i,j})$ for each $\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}$, an optimization problem is formulated on how to allocate bits to different $\tilde{\mathcal{R}}_{i,j}$ under the bit budget R such that the overall distortion

is minimized:

$$\min \sum_{\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}} D_{i,j} \quad \text{with the constraint} \quad \sum_{\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}} R_{i,j}(D_{i,j}) \leq R \tag{4.32}$$

where the minimization is taken on all rate-distortion function $R_{i,j}(D_{i,j})$ for all $\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}$. This is a typical optimization that appears in microeconomic theory [30]. Lagrange showed that the solution of (4.32) is equivalent to that of the following optimization problem:

$$\min \left(\sum_{\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}} D_{i,j} + \lambda \left(\sum_{\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}} R_{i,j}(D_{i,j}) - R \right) \right) \tag{4.33}$$

where $\lambda \in \mathbb{R}$ is called the Lagrange multiplier. By taking partial derivatives of (4.33) with respect to each $R_{i,j}(D_{i,j})$, we get

$$-\frac{\partial}{\partial R_{i,j}(D_{i,j})} D_{i,j} = \lambda \quad \text{for any} \quad \tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}. \tag{4.34}$$

(4.34) implies that the solution of (4.32) satisfies the marginal condition: the bits should be allocated to those $\tilde{\mathcal{R}}_{i,j}$ such that the reduction of the distortion is maximized. There have been many coding algorithms based on efficient bit allocations. Many of them relies on the time-consuming iterative methods to find the optimal allocations [4, 52]. However, there is still no method to ensure the solution to be global optimal. In this work, a non-iterative but sub-optimal method is devised for the block searching of the proposed attractor coding method.

Roughly speaking, the block searching algorithm tunes the domain pools of $\tilde{\mathcal{R}}_{i,j}$ such that all marginal distortion, i.e., $\frac{\partial}{\partial R_{i,j}(D_{i,j})} D_{i,j}$, are approximately the same (as $-\lambda$). The proposed method starts from small $R_{i,j}$ and enlarges the domain pool progressively until (4.34) is approximately satisfied. To be more precise, the local domain block searching algorithm is formulated as follows: a predefined quality factor $\lambda \in \mathbb{R}$ is set up to control the overall bit rates. For each $\tilde{\mathcal{R}}_{i,j} \in \tilde{\mathsf{R}}$, collage minimization is performed using D_0 as the search region:

$$d_0 = \min_{s_{i,j}, o_{i,j}, \mathbf{I}_{i,j}, \mathcal{D}_{k,l}} d(\mathbf{W}\mathbf{G}_{i,j}\mathcal{D}_{k,l}, \mathbf{A}\mathbf{W}\mathcal{R}_{i,j}) \quad \text{for} \quad \mathcal{D}_{k,l} \in \mathsf{D}_0. \tag{4.35}$$

and r_0 bits are required to store all parameters, namely $s_{i,j}, o_{i,j}, \mathbf{I}_{i,j}, \mathcal{D}_{k,l}$. Initialize the counter $i = 1$ and perform the following minimization followed by computing the optimality condition:

$$d_i = \min_{s_{i,j}, o_{i,j}, \mathbf{I}_{i,j}, \mathcal{D}_{k,l}} d(\mathbf{W}\mathbf{G}_{i,j}\mathcal{D}_{k,l}, \mathbf{A}\mathbf{W}\mathcal{R}_{i,j}) \quad \text{for} \quad \mathcal{D}_{k,l} \in \mathsf{D}_i \tag{4.36}$$

and r_i bits are used to store $s_{i,j}, o_{i,j}, \mathbf{I}_{i,j}, \mathcal{D}_{k,l}$. Checking the optimality criterion:

$$-\frac{d_i - d_{i-1}}{r_i - r_{i-1}} \leq \lambda. \tag{4.37}$$

If it is not the case, i is incremented by one to $i+1$ and repeat (4.36) and (4.37) until the criterion is satisfied or D_3 is reached. Once the procedure is completed, the bit rate $R_{i,j}$ and the distortion $D_{i,j}$ are r_{i-1} and d_{i-1} respectively if the criterion is satisfied by D_i. Otherwise, they are r_3 and d_3 respectively that implies the use of D_3.

4.4 Experimental Results

The proposed attractor coding methods using LPIFS are experimented on several images. Each of the images is partitioned into 16×16 overlapping range blocks, i.e., $B = 3$. Domain pool in definition 4.2 is used with $B = 3$, i.e.,

$$\mathsf{D} \equiv \{\mathcal{D}_{k,l} : \mathcal{D}_{k,l} \equiv \mathbf{B}^5_{k,l}\mathcal{I} \text{ with } k = 2^3 p, l = 2^3 q, 0 \leq p, q < 2^{N-5}\}. \tag{4.38}$$

With $B = 3$, the weights in $\mathbf{W}$ is as follows:

$$w_{u,v} = \left(1 - \frac{\cos\frac{2u+1}{16}\pi}{\cos\frac{\pi}{16}}\right)\left(1 - \frac{\cos\frac{2v+1}{16}\pi}{\cos\frac{\pi}{16}}\right), \quad \forall\, 0 \leq u, v < 8 \tag{4.39}$$

with $w_{u,v} = w_{16-u,v} = w_{u,16-v} = w_{16-u,16-v}$. Both definitions of $\mathbf{G}_{i,j}$, (4.6) and (4.7), are experimented. The range of the scaling parameters in the orthogonalized case is $(-1.5, 1.5)$. In order to examine the effects of the LPIFS in the overall coding performance. No local domain block searching is used at this stage. The block parameters are quantized and coded in the same way of the PIFS as described in section 2.5.2.

The coding results are tabulated in Table 4.1 and 4.2. The decoded images, Lena, are given in fig. 4.5 and the zoom-in view of the decoded images are provided in fig. 4.7.

The following observations are found in the experiments:

Observation 1: It can be found that attractor coding using the LPIFS results in similar PSNRs to those using the PIFS at the same bit rates for all four images but better visual quality for all images tested. In fact, the decoded images of the LPIFS suffer from lesser blocking artifacts than those of the PIFS since the range blocks are no longer disjoint in this case. Fig. 4.7 shows the zoom-in views of the decoded image Lena. It can be seen that the blocking effect is more prominent in the one produced from the PIFS. However there are nearly no blockiness in the case of the LPIFS.

Observation 2: As the structure of the LPIFS differs from that of the PIFS, it is necessary to study the decoder convergence of the LPIFS. It is advantageous that the fixed point of $\tilde{\mathbf{T}}$ can emerge in a few iterations. It implies lesser computations and shorter time for decoding. Therefore the errors versus the number of iterations for four images are plotted in fig. 4.4. It is found that, for $\mathbf{G}_{i,j}$ defined by (4.6), the number of iterations for convergence is image-dependent. It is shown that less than twenty iterations are sufficient for the transformation $\tilde{\mathbf{T}}$ in LPIFS to converge to its fixed point. It is about twice the number of iterations required in the PIFS which requires about ten in general [18]. The convergence rate of the LPIFS with $\mathbf{G}_{i,j}$ defined by (4.7) is detailed in the latter part of this section.

Observation 3: When one compares two definitions of $\mathbf{G}_{i,j}$, (4.6) and (4.7), it is found that (4.7) can give a better performance in terms of bit rates and PSNRs. It is mainly due to the constraint on the magnitudes of the scaling parameters in (4.6). No constraints are imposed on those scaling parameters by using (4.7). Therefore, the collage optimization would not suffer from optimal solution owing to the constrainted magnitudes of one of the block parameters.

After studying various aspects of the coding method using the LPIFS, the local

domain block searching is now embedded into the coding method. (4.7) is used as the definition of $\mathbf{G}_{i,j}$. Moreover, not all eight isometries are used in the domain block matching for all range blocks. It was reported that the PSNRs of the decoded image does not increase much by incorporating eight isometries but the bit rate increases owing to the use of extra 3 bits for representing the isometries [45]. Therefore no isometry is used for those domain blocks in $\mathsf{D}_i, i = 0, 1, 2$. For D_3, four isometries by 90° rotations are attempted first. If it does not satisfy the rate-distortion optimal condition, the remaining four isometries obtained by rotations and reflections are used to give the best match. Since the offset can be seen as the weighted DC of each range block, the offset parameters of adjacent blocks exhibit strong correlation. The most simplest way to exploit the redundancies, DPCM is applied followed by adaptive arithmetic coding [6]. The range of the scaling coefficients is $(-2, 2)$. Each of them is uniformly quantized and 5 bits are used for representation followed adaptive arithmetic coding. Table 4.3 and Fig. 4.8 show the bit rate against the PSNRs of four images by tuning the values of the Lagrange multiplier λ. Fig. 4.9 shows the decoded images of Lena and Fruits using different Lagrange multiplier λ.

The following observations are found in the experiments:

Observation 1: λ serves as a quality factor for the proposed attractor coding method. The larger the λ is, the smaller the bit rate is required in the cost of reduced PSNR for all images. (see table 4.3) If λ is chosen to be a small number, the rate-distortion optimality criterion (4.37) can be satisfied by some domain block with large error in the range block approximation. Therefore, more blocks are chosen from the small domain pools which require less bits in specifying them.

Observation 2: In order to study the distributions of domain blocks, the numbers of blocks chosen from different domain pools are listed in table 4.4 for the image Lena. It can be observed that most domain blocks are drawn from the smallest pool D_0. More blocks are taken from D_0 as λ increases. When λ is large, over half of the total number

of blocks are taken from D_0 that shows the local block searching can save a lot of bits by reduce the searching regions. Moreover, the encoding time is shorter as most range blocks need only a few domain blocks in collage optimization.

Observation 3: It can be found from fig. 4.9 that the decoded images does not severely suffer from blocking artifacts like those of the PIFS. However, when the bit rate decreases, it can be found from fig. 4.9 that some parts of the decoded images become blurred. It is expected nearly all range blocks choose its local domain block as λ becomes large. Therefore the collage is not very tight to the original image.

Observation 4: The convergence rate of the iterative decoding procedure is studied. The number of iterations required to reach the fixed point of different images are tabulated in table 4.5 It is found that the number of iterations is not very dependent on λ chosen but image only. Less than ten iterations are required for all images that are relatively a small number compared with the LPIFS with (4.6) as the definition of $\mathbf{G}_{i,j}$. (see fig. 4.4)

Observation 5: The inclusion of the local domain block searching does not affect the structure of the LPIFS. Moreover, the encoding algorithm without any local block searching, i.e., algorithm 2, can be seen as a special case of the one with local block searching. By putting λ to be a negative number, the rate-distortion optimality criterion (4.37) is not possible to be satisfied. Therefore, the local domain block searching becomes an exhaustive searching over the whole domain pool D in definition 4.2.

Table 4.1: Numerical results of the attractor coding using the LPIFS with (4.6) as the definition of $\mathbf{G}_{i,j}$ using the quantization and coding methods described in section 2.5.2 for the range block parameters

Image	LPIFS		PIFS	
	bpp	PSNR	bpp	PSNR
Flower	0.41	33.29	0.42	32.56
Fruits	0.41	31.54	0.42	31.74
Lena	0.41	31.43	0.42	31.01
Tiffany	0.41	31.20	0.42	31.35

Table 4.2: Numerical results of the attractor coding using the LPIFS with (4.7) as the definition of $\mathbf{G}_{i,j}$ using the quantization and coding methods described in section 2.5.2 for the range block parameters

Image	LPIFS		PIFS	
	bpp	PSNR	bpp	PSNR
Flower	0.41	33.96	0.42	32.10
Fruits	0.41	32.08	0.42	32.16
Lena	0.41	32.03	0.42	31.63
Tiffany	0.41	31.60	0.42	31.89

Table 4.3: Numerical results of the attractor coding using the LPIFS with (4.7) as the definition of $\mathbf{G}_{i,j}$ with local domain block searching

Image	bpp	PSNR	λ
Flower	0.236	31.23	100
	0.243	31.30	80
	0.252	31.41	45
	0.273	31.59	15
	0.284	31.67	6.5
Fruits	0.257	30.41	80
	0.264	30.54	60
	0.273	30.63	45
	0.307	30.84	15
	0.329	30.95	6.5

Image	bpp	PSNR	λ
Lena	0.248	30.26	80
	0.253	30.34	60
	0.279	30.51	15
	0.295	30.57	9.5
	0.310	30.60	6.5
Tiffany	0.236	30.63	80
	0.242	30.67	60
	0.249	30.72	45
	0.273	30.86	15
	0.299	30.92	6.5

Table 4.4: Statistics of the domain blocks chosen in different $D_i, i = 0, 1, 2, 3$ together with different number of isometries in the proposed local domain block searching

Image	Domain pool chosen, number of isometries used					λ
	$D_0, 1$	$D_1, 1$	$D_2, 1$	$D_3, 4$	$D_3, 8$	
Flower	1774	505	167	696	575	6.5
	1977	463	193	617	467	15
	2329	459	193	437	299	45
	2547	397	164	368	241	80
	2630	391	155	328	213	100
Fruits	1203	519	158	996	841	6.5
	1486	515	184	857	675	15
	1945	500	197	635	440	45
	2097	487	199	569	365	60
	2230	470	201	502	314	80
Lena	1431	738	259	879	662	6.0
	1480	718	270	863	638	6.5
	1646	679	261	794	589	9.5
	1862	635	264	691	517	15
	2041	589	263	620	456	25
	2263	589	244	492	381	45
	2362	575	239	455	338	60
	2479	573	215	385	317	80
Tiffany	1570	731	219	795	654	6.5
	1969	664	254	613	469	15
	2387	639	260	406	277	45
	2537	591	243	356	242	60
	2657	580	237	305	190	80

Table 4.5: Number of iterations required for the iterative decoding procedure of the coding method using the LPIFS with $\mathbf{G}_{i,j}$ defined by (4.7) with different values of λ (the number in the parenthesis is the number of iterations required in the LPIFS with $\mathbf{G}_{i,j}$ defined by (4.6))

Image	Number of iterations	λ
Flower (11)	8	100
	6	80
	8	45
	7	15
	7	6.5
Fruits (8)	4	80
	4	60
	4	45
	4	15
	4	6.5

Image	Number of iterations	λ
Lena (8)	8	80
	8	60
	8	15
	8	9.5
	8	6.5
Tiffany (17)	9	80
	9	60
	7	45
	9	15
	7	6.5

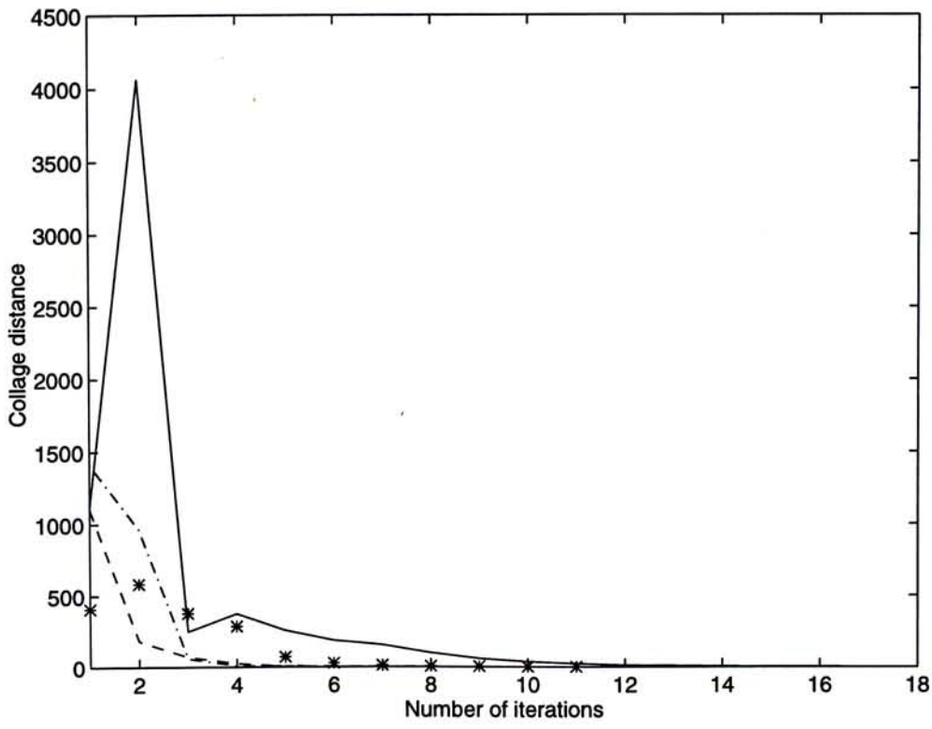


Figure 4.4: The plot of the collage distance against the number of iterations in the iterative decoding procedure of the coding method using the LPIFS with $\mathbf{G}_{i,j}$ defined by (4.6) for images, Flower (star), Fruits (dash), Lena (dashdot), and Tiffany (solid).

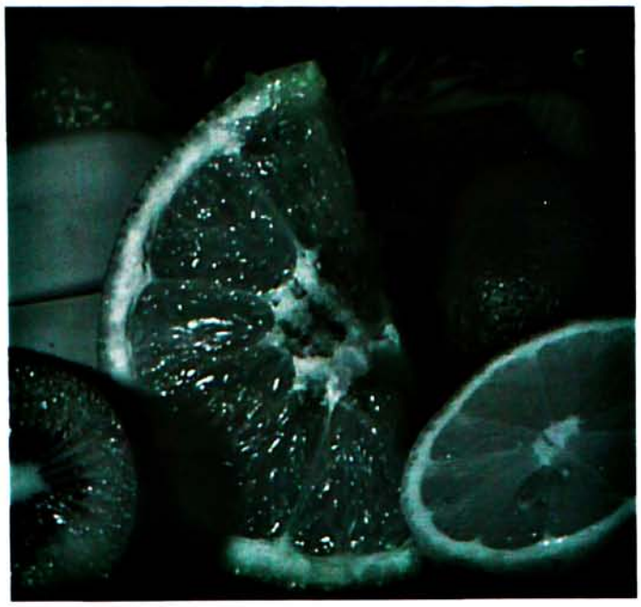
(a) original image Lena

(b) decoded image of the coding method using the PIFS: 0.42 bpp, PSNR=31.01dB

(c) decoded image of the coding method using the LPIFS with $\mathbf{G}_{i,j}$ defined by (4.6) : 0.41 bpp, PSNR=31.43dB

(d) decoded image of the coding method using the LPIFS with $\mathbf{G}_{i,j}$ defined by (4.7): 0.41 bpp, PSNR=32.03dB

Figure 4.5: Experimental result of image Lena.



(a) original image Fruits

(b) decoded image of the coding method using the PIFS: 0.42 bpp, PSNR=31.74dB

(c) decoded image of the coding method using the LPIFS with $\mathbf{G}_{i,j}$ defined by (4.6) : 0.41 bpp, PSNR=31.54dB

(d) decoded image of the coding method using the LPIFS with $\mathbf{G}_{i,j}$ defined by (4.6): 0.41 bpp, PSNR=32.08dB

Figure 4.6: Experimental result of image Fruits.

(a) Zoom-in view of the image in fig. 4.5 (b)

(b) Zoom-in view of the image in fig. 4.5 (d)

Figure 4.7: Zoom-in views of the decoded image Lena.

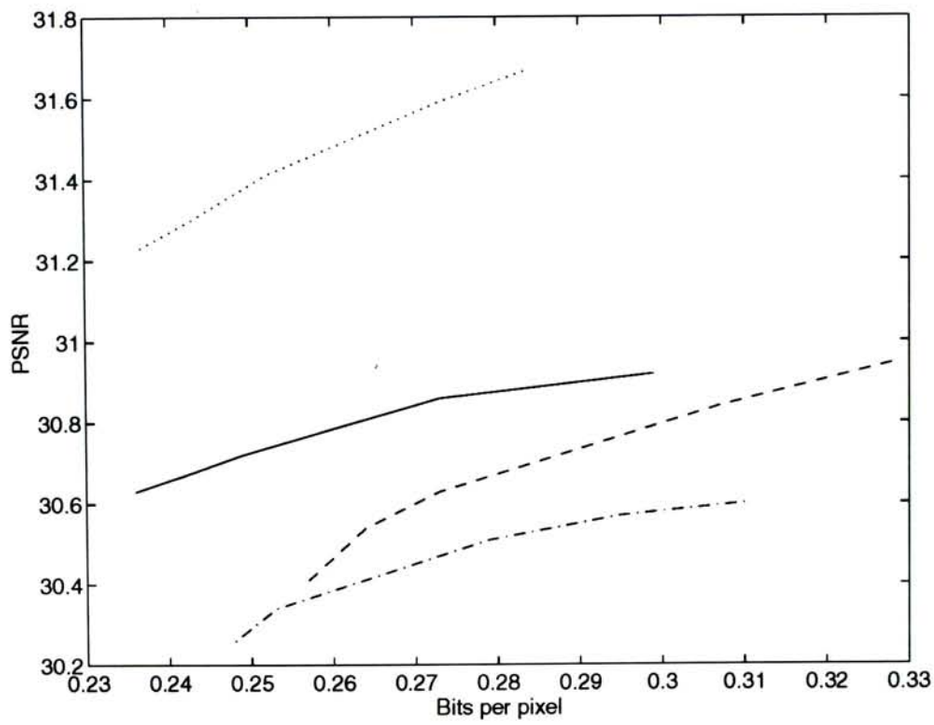


Figure 4.8: The plot of the PSNRs against the bits per pixel of the coding method using the LPIFS with $\mathbf{G}_{i,j}$ defined by (4.7) with different values of λ for the images, Flower (dotted), Fruits (dashed), Lena (dashdot), and Tiffany (solid).

(a) decoded image Lena of the coding method using the LPIFS with $\mathbf{G}_{i,j}$ defined by (4.6): $\lambda = 6.5$, 0.31bpp, PSNR=30.60dB

(b) decoded image Lena of the coding method using the LPIFS with $\mathbf{G}_{i,j}$ defined by (4.7): $\lambda = 80$, 0.25bpp, PSNR=30.26dB

(c) decoded image Fruits of the coding method using the LPIFS with $\mathbf{G}_{i,j}$ defined by (4.6): $\lambda = 6.5$, 0.329bpp, PSNR=30.95dB

(d) decoded image Fruits of the coding method using the LPIFS with $\mathbf{G}_{i,j}$ defined by (4.7): $\lambda = 80$, 0.257bpp, PSNR=30.41dB

Figure 4.9: Experimental result of images, Lena and Fruits, using the coding method using the LPIFS with local domain block searching.

4.5 Summary

In this chapter, a novel attractor image coding using the LPIFS is proposed. The major modification of the LPIFS over the PIFS is the inclusion of a weighting operator $\mathbf{W}$ in the affine transformation of each overlapping range block. Every range block laps with its adjacent blocks in the LPIFS. The proposed LPIFS can be seen as a generalized technique to include the existing PIFS as a special case.

In applying the LPIFS to image coding, the encoding procedure is similar to the existing method except the preprocessing of each range block before performing the domain block matching. The range block preprocessing can effectively eliminate the errors of the processed adjacent blocks and thus a closer collage can be obtained without involving any large system of equations. In order to provide smooth overlapping of the adjacent blocks, the cosine weights are chosen for the weighting operator that satisfy all desirable properties to be a good weighting operator. Our experiments show that the decoded images suffer from very low blocking effects only.

Furthermore, the local domain block searching is embedded into the encoder. this kind of searching is analyzed using the Gauss-Markov image model and experimented using real images. It is found that the use of this local block searching can effectively reduce the bits required in specifying the domain block addresses and shorten the encoding time.

Chapter 5

Conclusion

5.1 Original Contributions

Attractor image coding is still under investigations on both theories and implementations by many researchers. My work is just an opening of some potential research directions in this field. The main contributions, together with the corresponding publications, of this thesis are summarized as follows (in the order they are treated in the text, not in the order of importance):

1. **Analysis of the δ-MED criterion:**

   The theoretical foundation of the δ-MED criterion, on which the proposed adjacent block parameter estimations built, is presented based on the Gauss-Markov image model. It can be shown that if the range block satisfies the δ-MED criterion, it is likely to locate at the most smooth regions of an image where the proposed estimatin scheme performs.

2. **Attractor coding with adjacent block parameter estimations** [24]:

   Attractor coding embedded with adjacent block parameter estimations is proposed to alleviate the blocking effects. Experimental results have shown that the

proposed method can effectively code the block parameters of the smooth regions and conceal the undesirable blockiness.

3. **Lapped partitioned iterated function systems (LPIFS):**

   A novel image transformation called the LPIFS is formulated based on the existing PIFS. By tuning the weights in the weighting operator, the PIFS can be seen as a special case of the proposed LPIFS. Moreover, several properties of the LPIFS are investigated.

4. **Attractor coding using the LPIFS** [25]:

   A novel attractor coding method using the LPIFS is presented and examined. It is shown that most blockiness that appears in the decoded image of the PIFS is eliminated in that of the LPIFS. The number of iterations required in decoding is within ten that is acceptable for image coding.

5. **Local domain block searching with rate-distortion optimal stopping criterion**:

   A local domain block searching has been experimented and examined using the Gauss-Markov image model. It is found that this kind of searching is suitable for general images. Most range block can find a suitable domain block in a small domain pool. Encoding time becomes shorter and the number of bits in specifying the domain block addresses is reduced.

5.2 Subjects for Future Research

1. **Fast encoding algorithm for the attractor coding using the LPIFS :**

   The most time-consuming part in the encoding using the LPIFS is the weighted collage minimization. One possible research direction is to formulate a better LPIFS such that fast algorithm of the weighted collage minimization exists. This

approach is mainly inspired by the LOT which is designed for sufficiently smooth in block overlapping and fast in implementation.

2. **Including the quadtree partition and other block geometries in the LPIFS :**

   The LPIFS is formulated to extract square domain blocks of size four times larger than a range block. One possible generalization is to include quadtree partition and other block geometries in the domain pool. The basic formulation of the LPIFS is by no means altered by the elements in the domain pool. Better coding results are expected.

3. **Hybrid coding :**

   It is possible to combine several existing image coding techniques into a single coding method. The two proposed methods can be slightly modified and embedded into some existing hybrid methods like [5, 37, 46].

4. **Applying the proposed methods to color images :**

   Though all proposed methods are devised for gray-scale digital images, it is obvious of practical interest to extend the methods to color images. The extension to color images should not raise any big theoretical problems though several implementation issues must be resolved.

Chapter A

Fundamental Definitions

In this appendix, we give the definitions of the abstract spaces and the examples of these spaces mentioned in this thesis.

Definition A.1 *(Metric space) A metric space $(\mathcal{M}, d)$ is a set $\mathcal{M}$ equipped with a metric $d : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ satisfying the following three conditions:*

- $\forall x, y \in \mathcal{M}, d(x, y) = 0 \Longrightarrow x = y.$
- $\forall x, y \in \mathcal{M}, d(x, y) = d(y, x).$
- $\forall x, y, z \in \mathcal{M}, d(x, z) \leq d(x, y) + d(y, z).$

Definition A.2 *(Normed space) A normed space is a vector space $\mathcal{X}$ in which every element $x \in \mathcal{X}$ is associated a nonnegative real number $\|x\|$, called the norm of x, in such a way that*

- $\|x + y\| \leq \|x\| + \|y\|, \quad \forall x, y \in \mathcal{X}.$
- $\|\alpha x\| = |\alpha| \|x\|, \quad \forall x \in \mathcal{X}$ *and* $\alpha \in \mathbb{R}.$
- $\|x\| > 0$ *for* $x \neq 0.$

Definition A.3 *(Inner product space) An inner product space $\mathcal{H}$ is a vector space in which each pair of vectors $x, y \in \mathcal{H}$ is associated a real number $< x, y >$, called the inner product of x and y, such that the following rules hold:*

- $< x, y >=< y, x >, \quad \forall x, y \in \mathcal{H}.$
- $< x + y, z >=< x, z > + < y, z >, \quad \forall x, y \text{ and } z \in \mathcal{H}.$
- $< \alpha x, y >= \alpha < x, y >, \quad \forall x \in \mathcal{H} \text{ and } \alpha \in \mathbb{R}.$
- $< x, x >\geq 0, \quad \forall x \in \mathcal{H}.$
- $< x, x >= 0, \quad \forall x \in \mathcal{H}.$

Remark A.1 *It must be noted that both the definitions of a normed space and a inner product space assume the use of real vector space.*

Example A.1 *Any vector space $\mathbb{R}^N$ is an inner product space by defining $< \vec{x}, \vec{y} >= \vec{x}'\vec{y}$ where $\vec{x}, \vec{y} \in \mathbb{R}^N$. $'$ denotes the transpose of a vector.*

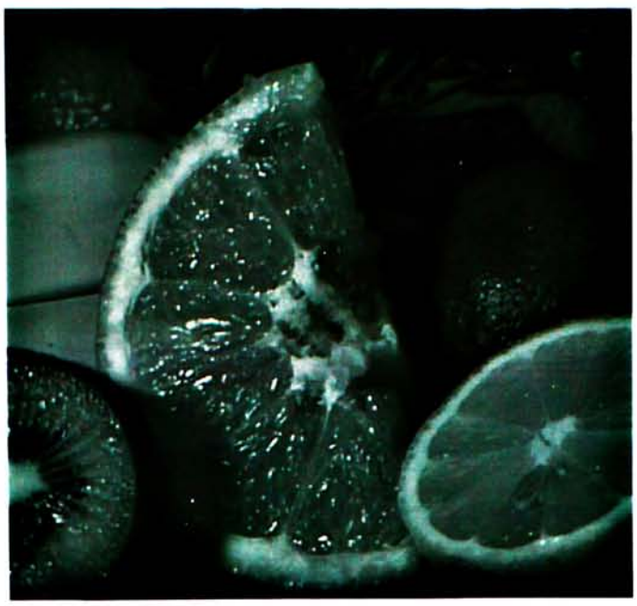
Example A.2 *For the vector space $\mathcal{X}$ with vectors of length N and each element of a vector is a random variable. $\mathcal{X}$ is an inner product space by defining $< \vec{X}_1, \vec{X}_2 >$ as*

$$< \vec{X}_1, \vec{X}_2 >= \mathbf{E}(\sum_i x_i^1 x_i^2)$$

where x_i^j is the i-th element of $\vec{X}_j, j = 1, 2$ and $\mathbf{E}$ is the expectation operator.

Appendix B

Original Images



(a) Flower

(b) Fruits

(c) Lena

(d) Tiffany

Bibliography

[1] Z. Barahav, D. Malah, and E. Karnin, "Hierarchical interpretation of fractal image coding and its applications," in *Fractal Image Compression: Theory and Application*, Yuval Fisher, Ed., chapter 5, pp. 91–117. Springer-Verlag, New York, 1995.

[2] Michael F. Barnsley and Stephen G. Demko, Eds., *Chaotic Dynamics and Fractals*, vol. 2 of *Notes and reports in mathematics in science and engineering*, Academic Press, London, 1st edition, 1986.

[3] Michael F. Barnsley, Arnaud Jacquin, Francois Malassenet, Laurie Reuter, and Alan D. Sloan, "Harnessing chaos for image synthesis," *Computer Graphics*, vol. 22, no. 4, pp. 131–140, Aug. 1988.

[4] Kai Uwe Barthel, "Entropy constrained fractal image coding," in *Proceedings of the NATO ASI on Fractal Image Encoding and Analysis*, Trondheim, Norway, July 1995.

[5] Kai Uwe Barthel, Jörg Schüttemeyer, Thomas Voyé, and Peter Noll, "A new image coding technique unifying fractal and transform coding," in *Proceedings ICIP-94 (IEEE International Conference on Image Processing)*, Austin, Texas, Nov. 1994, pp. 112–116.

[6] T.C. Bell, J.G. Cleary, and I.H. Witten, *Text Compression*, Prentice Hall, New York, 1990.

[7] W. K. Cham and R. J. Clarke, "DC coefficient restoration in transform image coding," *IEE Proceedings*, vol. 131, no. 7r, pp. 709–713, Dec. 1984.

[8] Bing Cheng and Xiaokun Zhu, "Multiresolution approximation of fractal transform," Technical Report UKC/IMS/96/79, Institute of Mathematics and Statistics, University of Kent, Canterbury, United Kingdom, 1996.

[9] Steven S. O. Choy, Yuk-Hee Chan, and Wan-Chi Siu, "Reduction of block-transform image coding artifacts by local statistics of transform coefficients," *IEEE Signal Processing Letters*, vol. 4, no. 1, pp. 5–7, Jan. 1997.

[10] R. J. Clarke, *Transform Coding of Images*, Academic Press, London, 1985.

[11] R. J. Clarke, *Digital Compression of Still Images and Video*, Academic Press, London, 1995.

[12] Thomas M. Cover and Joy A. Thomas, *Elements of Information Theory*, Wiley, New York, 1991.

[13] Geoffrey M. Davis, "A wavelet-based analysis of fractal image compression," to appear in *IEEE Transactions on Image Processing*, 1997.

[14] Franck Davoine, Marc Antonini, Jean-Marc Chassery, and Michel Barlaud, "Fractal image compression based on Delaunay triangulation and vector quantization," *IEEE Transactions on Image Processing*, vol. 5, no. 2, pp. 338–346, Feb. 1996.

[15] Ai-Hua Fan, Ka-Sing Lau, and Sze-Man Ngai, "Iterated function systems with overlaps," Research Report CUHK-96-33(107), Department of Mathematics, The Chinese University of Hong Kong, Shatin, Hong Kong, 1996.

[16] Y. Fisher, "Fractal image compression with quadtrees," in *Fractal Image Compression: Theory and Application*, Yuval Fisher, Ed., chapter 3, pp. 55–77. Springer-Verlag, New York, 1995.

[17] Y. Fisher and S. Menlove, "Fractal encoding with HV partitions," in *Fractal Image Compression: Theory and Application*, Yuval Fisher, Ed., chapter 6, pp. 119–136. Springer-Verlag, New York, 1995.

[18] Yuval Fisher, "Fractal image compression," Tech. Rep. vol 12 pp 7.1-7.19, Department of Mathematics, Technion Israel Institute of Technology, 1992, SIGGRAPH '92 COURSE NOTES.

[19] Bruno Forte and Edward Vrscay, "Solving the inverse problem for function/image approximation using iterated function systems: I theoretical basis," *Fractals*, vol. 2, no. 3, pp. 325–334, 1994.

[20] Bruno Forte and Edward Vrscay, "Solving the inverse problem for function/image approximation using iterated function systems: II algorithms and computations," *Fractals*, vol. 2, no. 3, pp. 335–346, 1994.

[21] Bruno Forte and Edward Vrscay, "Theory of generalized fractal transforms," in *Proceedings of the NATO ASI on Fractal Image Encoding and Analysis*, Trondheim, July 1995.

[22] Carsten Frigaard, Jess Gade, Thomas Hemmingsen, and Torben Sand, "Image compression based on a fractal theory," Internal Report S701, Institute for Electronic Systems, Aalborg University, Aalborg, Denmark, 1994.

[23] John C. Hart, "Fractal image compression and the inverse problem of recurrent iterated function systems," *IEEE Computer Graphics and Applications*, vol. 16, no. 4, pp. 25–33, July 1996.

[24] Hau Lai Ho and Wai Kuen Cham, "Fractal image compression with adjacent block parameter estimations," in *Proceedings IEEE Region Ten Conference '96 - Digital Signal Processing Applications*, Perth, Western Australia, Nov. 1996, pp. 793–797.

[25] Hau Lai Ho and Wai Kuen Cham, "Attractor image coding using lapped partitioned iterated function systems," in *Proceedings ICASSP-97 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, Munich, Germany, Apr. 1997, vol. 4, pp. 2773–2776.

[26] Arnaud E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Transactions on Image Processing*, vol. 1, no. 1, pp. 18–30, Jan. 1992.

[27] Arnaud E. Jacquin, "Fractal image coding: A review," *Proceedings of the IEEE*, vol. 81, no. 10, pp. 1451–1465, Oct. 1993.

[28] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, University of California, Davis, 1989.

[29] B. Jawerth and W. Sweldens, "Biorthogonal smooth local trigonometric bases," *J. Fourier Anal. Appl.*, vol. 2, no. 2, pp. 109–133, 1995.

[30] Geoffrey A. Jehle, *Advanced Microeconomic Theory*, Prentice Hall, Englewood Cliffs, N.J., 1991.

[31] E. Kreyszig, *Introductory Functional Analysis with Applications*, John Wiley & Sons, 1978.

[32] Lars M. Lundheim, "A discrete framework for fractal signal modeling," in *Fractal Image Compression: Theory and Application*, Yuval Fisher, Ed., chapter 7, pp. 137–151. Springer-Verlag, New York, 1995.

[33] H.S. Malvar and D.H. Staelin, "The LOT: Transform coding without blocking effects," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 4, pp. 553–559, Apr. 1989.

[34] Benoit B. Mandelbrot, *The Fractal Geometry of Nature*, W.H. Freeman, New York, 1977.

[35] Peter R. Massopust, *Fractal Functions, Fractal Surfaces, and Wavelets*, Academic Press, San Diego, 1994.

[36] Shigenobu Minami and Avideh Zakhor, "An optimization approach for removing blocking effects in transform coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 2, pp. 74–82, Apr. 1995.

[37] Donald M. Monro, "A hybrid fractal transform," in *Proceedings ICASSP-93 (IEEE International Conference on Acoustics, Speech and Signal Processing)*, 1993, vol. 5, pp. 169–172.

[38] Geir E. Øien, *L_2-optimal Attractor Image Coding with Fast Decoder Convergence*, Ph.D. thesis, The Norwegian Institute of Technology, Trondheim, Norway, Apr. 1993.

[39] Geir E. Øien and Skjalg Lepsøy, "A class of fractal image coders with fast decoder convergence," in *Fractal Image Compression: Theory and Application*, Yuval Fisher, Ed., chapter 8, pp. 153–175. Springer-Verlag, New York, 1995.

[40] Thomas P. O'Rourke and Robert L. Stevenson, "Improved image decompression for reduced transform coding artifacts," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 6, pp. 490–499, Dec. 1995.

[41] Dan C. Popescu, Alex Dimca, and Hong Yan, "A nonlinear model for fractal image coding," *IEEE Transactions on Image Processing*, vol. 6, no. 3, pp. 373–382, Mar. 1997.

[42] Ricardo L. De Queiroz, *On Lapped Transform*, Ph.D. thesis, University of Texas at Arlington, Arlington, Texas, Dec. 1994.

[43] Emmanuel 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)*, Adelaide, Australia, Apr. 1994, vol. 5, pp. 569–572.

[44] Walter Rudin, *Functional Analysis*, McGraw-Hill, New York, 2nd edition, 1991.

[45] Dietmar Saupe, "The futility of square isometries in fractal image compression," in *Proceedings ICIP-96 (IEEE International Conference on Image Processing)*, Lausanne, Switzerland, Sept. 1996, vol. I, pp. 161–164.

[46] Nuygen T. Thao, "A hybrid fractal-DCT coding scheme for image compression," in *Proceedings ICIP-96 (IEEE International Conference on Image Processing)*, Lausanne, Switzerland, Sept. 1996, vol. I, pp. 169–172.

[47] Nuygen T. Thao, "Local search fractal image compression for fast integrated implementation," in *Proceedings ISCAS '97 (IEEE International Symposium on Circuits and Systems)*, Hong Kong, June 1997, vol. V, pp. 1333–1336.

[48] F.W. Tse, *DC Coefficient Restoration for Transform Image Coding*, M.Phil. thesis, The Chinese University of Hong Kong, Shatin, Hong Kong, June 1996.

[49] Martin Vetterli and Jelena Kovacevic, *Wavelets and Subband Coding*, Prentice Hall, Englewood Cliffs, N.J., 1995.

[50] Greg Vines, *Signal Modelling with Iterated Function Systems*, Ph.D. thesis, Georgia Institute of Technology, Atlanta, Georgia, May 1993.

[51] B. Wohlberg and G. de Jager, "Fractal coding performance for first-order Gauss-Markov models," *Electronics Letters*, vol. 32, no. 5, pp. 441–442, Feb. 1996.

[52] Z. Xiong, K. Ramchandran, and M.T. Orchard, "Space-frequency quantization for wavelet image coding," to appear in *IEEE Transactions on Image Processing*, May 1997.