OPERATOR THEORETIC IMAGE CODING

by

HONG PUH

A dissertation submitted to the Graduate Faculty in Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

The City University of New York

1996

UMI Number: 9707145

This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

June 5, 1996
Date

Patrick L Combettes
Chair of Examining Committee

Sept. 25, 1996
Date

Gerard G. Lowen
Executive Officer

PROFESSOR JOSEPH BARBA

PROFESSOR KENNETH SOBEL

PROFESSOR RICHARD TOLIMIERI

DOCTOR M. REHA CIVANLAR

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

ABSTRACT

OPERATOR THEORETIC IMAGE CODING

by

Hong Puh

Advisor: Professor Patrick L. Combettes

Versatile coding techniques are required to face the increasing demand of modern digital communication technology for efficient digital image transmission and storage schemes. In this dissertation, a unified framework for iterative image coding is introduced. In this framework, each basic feature of an image is individually encoded into a nonexpansive operator defined on the image space. This operator admits as fixed point set the class of images possessing the feature in question. Consequently, an image is associated with a family of operators which are specified during the encoding process, while the decoding process consists of finding a common fixed point of these operators. Decoding is achieved via a powerful parallel algorithm which proceeds by extrapolated relaxations of weighted averages of variable blocks of operators. This approach generalizes several coding techniques, in particular fractal coding -- which employs a single contractive operator – and set theoretic coding – which employs convex projection operators. The effectiveness and the flexibility of the proposed operator theoretic framework is illustrated through numerical simulations on grayscale images.

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my mentor, Prof. P. L. Combettes, not only for his guidance, but also for his assistance in all phases of this work. Thanks are also given to the members of the examing committee, Dean G. Lowen, Professor J. Barba, Professor K. Sobel, Professor R. Tolimieri, and Doctor M. R. Civanlar.

I would like to give special thanks to Prof. J. Barba, director of the CMIPS laboratory, for his encouragements during the past three years and for making his facilities available to me. Thanks also go to all the staff of the CMIPS lab.

This work was partially supported by the National Science Foundation under grant MIP-9308609.

This dissertation is dedicated to my parents with respect and love.

Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 Overview

The goal of this dissertation is to develop, study, and implement a novel approach to image coding – operator theoretic image coding (OTIC).

The basic idea underlying this approach is to employ operators defined on the image space to characterize the basic features of the image to be encoded. This is achieved by associating a nonexpansive operator with each feature of the input image, any image possessing the feature under consideration being left invariant by this operator. Thus, the complex features of an image are individually encoded into a family of operators. To recover the input at the decoder's end, the operators are employed repeatedly to construct a sequence of images converging to one of their common fixed points, i.e., to an image possessing all the features selected by the encoder. Any such image is an acceptable solution to the decoding problem.

1.2 Motivation

As mentioned in the editorial of the February 1995 issue of the *Proceedings of the IEEE,* an historical transition is taking place in the development of digital image and video technology. Almost on a daily basis, new products and services based on video technology are advertized. With the continuing growth of multimedia communications technology, demand for efficient digital image transmission and storage is increasing rapidly, especially in areas such as teleconferencing, digital broadcast codec, video telephony, facsimile transmission of printed material, graphics, or transmission of remote-sensing images obtained by satellites and reconnaissance aircrafts. Another area of demand is data storage, where large image databases must be efficiently stored, e.g., archiving of medical images, multispectral images, finger prints, and drawings.

A major obstacle in many applications is the vast amount of data involved in the direct representation of a digital image. For example, a digitized version of a single color picture at TV resolution contains on the order of one million bytes; 35mm resolution requires ten times that amount. The use of digital images is often not viable due to high storage and transmission costs, even when image acquisition and display devices are quite affordable.

Technically, a key that opens the door of digital imagery to a wider range of applications is data compression technology. Image compression or coding is the process of representing an image in a way that realizes a desired objective such as analog-to-digital conversion, low bit-rate transmission, or message encryption. Coding techniques rely on mathematical theories to realize reliable and efficient algorithms. To answer the needs of modern image coding, more advanced theories and more synthesized methods must be developed. It is the goal of this research to build a novel framework for iterative image coding that extend existing image coding techniques, based on theoretical results from nonlinear functional analysis.

1.3 Research Methodology

1.3.1 Basic Principles

In image analysis, the complex perceptual feature of an image can be decomposed into a family of basic features. The philosophy underlying our approach to coding is that it is more effective to faithfully represent an image by coding its basic features individually rather than by coding it as a whole directly. This point of view raises several questions:

- How to extract and represent the basic image features,
- How to encode those features, and then,
- How to effectively recover the input image from its feature codes.

A suitable mathematical formalism must be set up to answer these questions.

1.3.2 Operator Theoretic Formalization

Mathematically, a digital image h can be viewed as a point (or a vector) in a euclidean image space Ξ. An image feature or property can be described by a constraint Ψ_i on the vectors of Ξ. Inspired by nonlinear operator theory, we choose to represent the constraint Ψ_i by a nonexpansive operator T_i which will admit as fixed points all the points that satisfy

Ψ_i, i.e.,

$$(\forall x \in \Xi)\ \ x \text{ satisfies } \Psi_i \quad \Leftrightarrow \quad T_i(x) = x. \tag{1.1}$$

Consequently, all those possible feature operators constitute an image code. This operator theoretic formalism is close to that of convex set theoretic estimation [11], where each constraint is associated with a closed and convex feature set in the image space

$$S_i = \{x \in \Xi \mid x \text{ satisfies } \Psi_i\}. \tag{1.2}$$

More precisely, (1.1) can be put in the set theoretic format (1.2) by letting S_i be the set of fixed points of T_i, namely[1]

$$S_i = \operatorname{Fix} T_i \triangleq \{x \in \Xi \mid T_i(x) = x\}. \tag{1.3}$$

Thus, if the input image h is satisfactorily described by a family of features or constraints $(\Psi_i)_{i\in I}$, it will be equally well represented by the family of operators $(T_i)_{i\in I}$ obtained in (1.1) or, equivalently, by the fixed point sets (feature sets) $(S_i)_{i\in I}$ obtained in (1.2)-(1.3). Since the original image remains invariant under the operators $(T_i)_{i\in I}$, the decoding process amounts to finding one of their common fixed points, i.e.

$$\text{Find } x \in \Xi \quad \text{such that} \quad (\forall i \in I)\ \ T_i(x) = x. \tag{1.4}$$

Alternatively, the problem can be posed as that of finding a point in the intersection of the fixed point sets, i.e.

$$\text{Find } x \in S = \bigcap_{i\in I} S_i. \tag{1.5}$$

[1]As will be seen in Chapter 3, the nonexpansivity of T_i guarantees that S_i is closed and convex.

The fixed point formulation (1.4) will be preferred to the conceptually equivalent set theoretic formulation (1.5) for most algorithms for solving the latter require the operators of projections onto the sets, which are often difficult to determine and computationally expensive to implement. On the other hand, algorithms for solving the former require only the knowledge of the nonexpansive operators $(T_i)_{i \in I}$, which are much more general and readily available.

1.3.3 Plan of Work

Three main points will be studied to develop the proposed operator theoretic image coding (OTIC) framework.

- Construction of a pool of nonexpansive operators to constitute a rich codebook and parametrization of this codebook.
- Selection of a code to represent the image, i.e., selection of a family of nonexpansive operators from the codebook.
- Numerical solution of (1.4), i.e., construction of a common fixed point of a family of nonexpansive operators.

1.4 Features

The advantages of the proposed operator theoretic approach are the following.

- It provides a general, abstract, and flexible problem formulation.

- Each feature of the image (or a subblock thereof) can be treated individually and easily encoded into a nonexpansive operator.
- Common fixed points of the operators can be constructed iteratively through efficient parallel algorithms.
- High compression rates can be achieved since complex image features are simply encoded into indices of operators.

1.5 Outline of the Dissertation

In Chapter 2, we review several important image coding techniques, namely transform coding, predictive coding, and vector quantization. Recent iterative techniques closely related to our work are also discussed, namely fractal coding and set theoretic coding. In Chapter 3, the mathematical foundation supporting our operator theoretic approach is laid. Important theorems on the convergence of iterative fixed point algorithms are given and proved. We proceed to Chapter 4 with a presentation of useful nonexpansive operators and a discussion of the computations related to the enforcement of the constraints they represent. Chapter 5 systematically describes the design of an operator theoretic codec (coder/decoder). Specific issues such as encoding, parametrization, and decoding are addressed. Chapter 6 covers a detailed numerical implementation of a general OTIC image codec whose performance is compared to that of a fractal codec. Chapter 7 concludes the dissertation and outlines potential extensions and directions for further research.

Chapter 2

REVIEW OF IMAGE CODING TECHNIQUES

2.1 General Overview

A digital image is specified by a two dimensional real array whose entries are called pixels and digital video is a sequence of digital images. Image and video coding requires a reduction of the average number of bits used per pixel while maintaining a good subjective visual quality rather than an accurate match between the original and encoded images. Numerous compression techniques have been developed in response to the rapid growth of the fields of applications of digital imagery, e.g., transform coding, predictive coding, hybrid coding, and their adaptive versions [5], [10], [20], [59]. These techniques usually exploit psychovisual as well as statistical redundancies in the image data to reduce the bit rate. Thanks to the development of recent mathematical theories and the availability of efficient computing facilities, there has been a tendency to combine different classical coding techniques with modern digital signal processing techniques in order to obtain greater coding efficiency.

There are a number of ways to classify image coding techniques. Among them are:

1. Classification according to the domains where the processing takes place [10]:
   (a) Transform coding
   (b) Predictive coding
   (c) Hybrid coding
2. Classification according to compression rates [2], [33]:
   (a) Low rate coding
   (b) Very low rate coding
3. Classification according to the methods of quantization [20], [39]:
   (a) Scalar quantization
   (b) Vector quantization
   (c) Combined scalar and vector quantization
4. Classification according to the coding models [2], [48]:
   (a) Non-model compression
   (b) Predictor model compression
   (c) Transformation model compression

We review here briefly the major techniques related to our research work.

2.2 Transform Coding

Signal transformation is a major tool in a number of coding methods. Coding in the transform – as opposed to the pixel – domain has several advantages. A first advantage of transforms is their whitening property: in the transform domain, the transform coefficients are decorrelated so that redundancies are eliminated. This is equivalent to diagonalizing the covariance matrix of the signal. However, the only transform that achieves exact diagonalization, the Karhunen-Loève transform (KLT) [20], [26], is usually impractical. Many other transforms come close to diagonalization and are therefore popular, e.g., the discrete cosine transform (DCT) [44]. The DCT was proposed in [1] as an approximation for the KLT of a first-order Gauss-Markov process with large positive correlation coefficients. Overall, the DCT has proven to be a robust approximation to the KLT, and it is used in several digital image standards [55], [60], and video compression standards [32], [55]. The second advantage is that the new domain is often more appropriate for quantization using a perceptual criterion. The third advantage of transform coding is that the previous properties come at a low computational price. The transform decomposition itself is computed using fast algorithms, and quantization in the transform domain is often a simple scalar quantization process.

The Fourier transform has often been used for image coding but it has now been largely superseded by other transforms which have the advantage of higher coding efficiency and which require only real number manipulations, e.g., the DCT. For an image with high inter-pixel correlation, many of the high-order spectral coefficients will be small and may either be encoded with very few bits or deleted completely. In other words, a lot of redundancy in the input signal can be removed [10]. Compared with predictive coding, which will be introduced next, this method is relatively complex, and its implementation has benefited greatly from recent advances in high-speed digital hardware. It is also somewhat less sensitive to errors than predictive coding.

2.3 Predictive Coding

Among standard methods, predictive coding is the main alternative to transform coding for image data compression [10]. Its principle is very simple. Since the image data source is assumed to be highly correlated on the average, pixels in a same neighborhood will tend to have similar amplitudes. We may therefore use the values of one or more previously coded elements in the same line, or in a previous lines, or frames to form a prediction of the present pixel [25]. Given the statistical nature of the image, we expect, again on the average, that the prediction - usually based on linear models - will be satisfactory and that the magnitude of the difference signal formed by subtracting the prediction from the actual value of the present pixel will be small. This signal is then encoded and transmitted or stored. In order to reconstruct the pixel values, the same prediction process is carried out by the decoder, and the difference signal is added to it.

2.4 Vector Quantization

Vector quantization is a very popular data compression technique for reducing transmission and storage bit rates. Over the past ten years, it has proved a valuable coding technique in a variety of applications, especially in voice and image coding [3], [20], [39]. In practical applications, images are partitioned into sub-blocks prior to being quantized for the sake of reducing complexity. Therefore, vector quantization is usually a block-wise processing technique.

Vector quantization is so-named as opposed to scalar quantization. Its coding structure was developed by Shannon in his theoretical development of source coding with a fidelity criterion [51]. The basic idea is that coding systems can perform better if they operate on vectors or groups of symbols (such as speech samples or image pixels) rather than on individual symbols or samples. Shannon called vector quantizers "block source codes with fidelity criterion" and they have also been called "block quantizers". On the contrary, scalar quantization coding techniques perform on individual samples of waveforms, e.g., on either pixels of images for spatial domain coding and on transform coefficients for transform coding. These techniques are not optimal to the extent that the processed samples are still somehow correlated. According to Shannon's rate-distortion theory, a better performance is always achievable in theory by coding vectors instead of scalars, even though the data source is memoryless [39].

Vector quantization rounds off groups of numbers together rather than one at a time. To specify a basic vector quantizer, one needs a codebook, the set of all possible reproduction vectors. Thus, vector quantization entails the mapping of vectors into binary vectors, which index a limited number of possible reproduction. The decoding process is just a simple procedure of table look-up.

A feature of vector quantization is that high compression ratios are possible with relatively small block sizes. The use of smaller block sizes in block coding has been known to lead to better subjective quality. A second feature is that the decoder is methodologically very simple to implement, making vector quantization attractive for single-encoder, multi-decoder applications such as videotext and archiving. However, vector quantizers are seriously hampered by complexity. In general, design complexity and encoding complexity are exponentially proportional to the codebook size and the dimension of the input vectors. By far, efforts have focused mostly on optimal vector quantizers [5], [20], efficient search techniques [29], and the development of a variety of applications of vector quantization to images in the spatial domain, the predictive domain, the transform domain, and combinations of these, known as hybrid domains [10], [25].

2.5 Fractal Coding

Image fractal coding recently received much attention [18]. The notion of Fractal Image Compression, according to which real-life objects or images can be modeled by attractors of two-dimensional affine transformations, was first proposed in [4]. However, fractal theory alone does not provide any constructive procedure for the "encoding" of a grayscale image in an automated way. An automated coding scheme was proposed in [23] where piecewise affine contractive transformations making use of the partial self-transformability of images were defined.

The basic idea of fractal image coding is that the input image h is viewed as the unique fixed point of a contractive operator T defined on the image space Ξ. It can therefore be represented by the operator T. Thanks to the Banach Contraction Theorem, an encoded image can be recovered as the limit of the sequence of successive approximations $(T^n(x_0))_{n\geq 0}$, where x_0 is an arbitrary image [23], [24]. This scheme actually relies on the following assumptions:

1. A contractive transformation T can be found, which admits h as its fixed point.
2. Image redundancies can be efficiently captured and exploited through the process of self-transformation.
3. The encoded image can be closely approximated by the image obtained after a finite number of iterations of an image operator.

As can be seen, the central issue of fractal coding is to construct an appropriate contractive operator, i.e., an operator $T: \Xi \to \Xi$ such that

$$(\exists k \in]0,1[)(\forall (x,y) \in \Xi^2) \quad \|T(x) - T(y)\| \leq k\|x - y\|, \tag{2.1}$$

where $\|\cdot\|$ is the norm of Ξ. The ideal goal is to find a contractive operator which leaves the original image h fixed, that is

$$h = T(h). \tag{2.2}$$

The original image h is then recovered through the Picard iterations [63]

$$(\forall n \in \mathbb{N}) \quad x_{n+1} = T(x_n), \tag{2.3}$$

where x_0 is any image. However, in practice, it is hard to find an operator T that satisfies (2.2) exactly. A more practical approach is to select a contraction T_i from a finite pool of contractions $(T_j)_{j\in J}$ so that h is the best approximate fixed point, i.e.

$$\|h - T_i(h)\| = \min_{j\in J} \|h - T_j(h)\|. \tag{2.4}$$

It is easy to show that the reconstruction error is bounded as follows [4]

$$(\forall n \in \mathbb{N}) \ \ \|h - x_n\| \leq \frac{1}{1-k}\|h - T(h)\| + k^n\|h - x_0\|. \tag{2.5}$$

The second term on the right hand side can be ignored when n becomes sufficiently large. Recently, it was shown in [34] that fractal image coding can be viewed as a generalized predictive coding scheme. It can yield higher prediction gains than conventional predictive coding by using noncausal predictors and long-term predictors.

Some comments are in order:

1. To ensure the existence and the uniqueness of a fixed point, a contractive operator has to be employed. On the other hand, one can usually get a T whose fixed point only approximates h as in (2.4). Thus, the restriction to a single contractive operator cannot be expected to give accurate results. Instead, it limits the possible choices from other – possibly more effective – operators.

2. In terms of convergence rate, the expression

$$(\forall n \in \mathbb{N}) \ \|h - x_n\| \leq \frac{k^n}{1-k}\|x_1 - x_0\|, \tag{2.6}$$

shows that k should be small. This further restricts the choice of operators within the class of contractive mappings.

3. The input image is encoded into a single operator T. It is therefore often difficult to adequately model a broad spectrum of image features or properties.

4. The speed of convergence of the rudimentary Picard iteration scheme (2.3) is known to be poor.

2.6 Set Theoretic Coding

Following the principles of set theoretic estimation [11], a signal h can be viewed as a point in an abstract space Ξ. Each constraint Ψ_i available about h confines it to the set

$$S_i = \{x \in \Xi \mid x \text{ satisfies } \Psi_i\}. \tag{2.7}$$

In estimation problems, the constraints usually arise from *a priori* information and from the observed data. One thus obtains a family of property sets $(S_i)_{i \in I}$ whose intersection S represents the class of feasible solutions for the problem.

By applying the above formalism to the image coding problem, the encoding consists of specifying the property sets whereas the decoding consists of finding a point in their intersection. The set theoretic approach was first introduced to image coding in [50] to effectively combine pixel domain and frequency domain coding. More recently, [56] elaborated the concept of set theoretic DCT coding based on scalar quantization. A set theoretic approach has also been adopted to remove blocking effects due to block-wise coding in JPEG decoding [49], [61], in the post-processing of VQ decoding [38], and to conceal damaged block transform coded images [54]. These works are therefore not set theoretic coders *per se* (where the code would actually consist of sets) but the cascade of a standard coding technique and a set theoretic image enhancement scheme. In these studies, the successive projections POCS algorithm – proposed in [7] and popularized in [62] – was used to find a feasible image. For m sets, POCS is described by the algorithm

$$(\forall n \in \mathbb{N}) \quad x_{n+1} = P_{n\,(\text{modulo}\, m)+1}(x_n), \tag{2.8}$$

where P_i is the operator of projection onto S_i. All in all, the set theoretic approach in coding is still in its infancy and the concept of a set theoretic coder has yet to be systematically developed.

Set theoretic image coding has at least two advantages. First, it is a coding scheme which encodes sets instead of points in the image space. Encoding a set is easier since a set can be specified by a constraint that gives explicit information about the original image. Moreover, by construction, the input image does belong to the feasibility set and it is therefore carried over to the decoder. By contrast, the commitment to a single point will inevitably incur information loss. As a result, more freedom and possibilities are given to devise a robust decoding scheme. Secondly, when dealing with input images containing several features of various perceptual or geometrical features, one can encode those features individually. This opens the possibility of effectively coding images with complex features. Sophisticated solution algorithms are described in [8], [11], [13], and [30].

Chapter 3

MATHEMATICAL FOUNDATIONS

3.1 Scope

In this chapter, we introduce the basic mathematical tools relevant to operator theoretic coding. Our notations are as follows: $\mathbb{N}$ is the set of nonnegative integers, $\mathbb{N}^*$ the set of positive integers, $\mathbb{R}$ the set of real numbers, and $\mathbb{C}$ the set of complex number. Ξ denotes a euclidean (finite-dimensional real Hilbert) space, d its distance, $\langle \cdot \mid \cdot \rangle$ its scalar product, and $\|\cdot\|$ its norm. We use T or F to represent an operator defined from Ξ into itself. Id is the identity operator. 1_S denotes the indicator function of the set S, which is defined as

$$(\forall x \in \Xi)\ 1_S(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{if } x \notin S. \end{cases} \tag{3.1}$$

The closed ball of center x and radius γ in Ξ is denoted by $B(x, \gamma)$. $(x_n)_{n \geq 0}$ is a sequence of points in Ξ indexed on $\mathbb{N}$. For detailed accounts of nonlinear functional analysis, convex analysis, and fixed point theory, consult [17], [21], [28], [47], [63].

3.2 Basic Operators

We start this section with some important definitions.

3.2.1 Definitions

Definition 3.1 *The sequence* $(x_n)_{n\geq 0}$ *is said to be convergent to* x *if*

$$\|x_n - x\| \to 0 \quad \text{as } n \to +\infty, \tag{3.2}$$

which is represented by $x_n \to x$.

Definition 3.2 $\operatorname{Fix} T$ *denotes the fixed point set of an operator* $T : \Xi \to \Xi$, *which is defined as*

$$\operatorname{Fix} T = \{x \in \Xi \mid x = T(x)\}. \tag{3.3}$$

Definition 3.3 *A nonempty subset* C *of* Ξ *is said to be an affine subspace if there exists a point* y *in* Ξ *such that*

$$S = \{x + y \mid x \in S'\}, \tag{3.4}$$

where S' *is a vector subspace of* Ξ. *Equivalently,* S *is an affine subspace if*

$$(\forall (x, y) \in \Xi^2)(\forall \alpha \in \mathbb{R}) \; (1 - \alpha)x + \alpha y \in S. \tag{3.5}$$

When α *is restricted to* $[0, 1]$ *in (3.5),* S *is called a convex set. Thus, affine subspaces are special cases of convex sets.*

Definition 3.4 *Let $f : C \to \mathbb{R}$ be a function, where C is a convex subset of Ξ. Then f is convex if*

$$(\forall (x,y) \in C^2)(\forall \alpha \in [0,1])\ f((1-\alpha)x + \alpha y) \leq (1-\alpha)f(x) + \alpha f(y). \tag{3.6}$$

We now define some general classes of linear and nonlinear operators and show relationships existing among them. Those are contractive operators, nonexpansive operators, firmly nonexpansive operators, projectors onto closed and convex sets, and affine operators. For details see [21] and [63].

Definition 3.5 *An operator $T\colon \Xi \to \Xi$ is said to have Lipschitz constant k if*

$$(\forall (x,y) \in \Xi^2)\ \ \|T(x) - T(y)\| \leq k\|x - y\|. \tag{3.7}$$

If $k \in\ ]0,1[$, T is said to be contractive; if $k = 1$, T is said to be nonexpansive. Obviously, nonexpansivity includes contractivity as a special case.

Definition 3.6 *An operator $T\colon \Xi \to \Xi$ is said to be firmly nonexpansive if*

$$(\forall (x,y) \in \Xi^2)\ \|T(x) - T(y)\|^2 \leq \langle T(x) - T(y) \mid x - y\rangle. \tag{3.8}$$

To see the relationship between firm nonexpansivity and nonexpansivity, we invoke the Cauchy-Schwarz inequality, which gives

$$(\forall (x,y) \in \Xi^2)\quad \langle T(x) - T(y) \mid x - y\rangle \leq \|T(x) - T(y)\| \cdot \|x - y\|. \tag{3.9}$$

Then we can derive inequality (3.7) for $k = 1$ from (3.8). Therefore, firm nonexpansivity implies nonexpansivity. Another commonly used operator is the projector onto a convex set in Ξ, which is defined as follows.

Definition 3.7 *Let A be a nonempty closed and convex set. A point $P_A(x)$ in A is said to be the projection of a point x onto A if*

$$\|x - P_A(x)\| = \inf_{z \in A} \|x - z\|. \tag{3.10}$$

The projection operator P_A is characterized by

$$(\forall x \in \Xi)(\forall z \in A) \quad \langle x - P_A(x) \mid z - P_A(x) \rangle \leq 0. \tag{3.11}$$

The next two propositions explicit the relationship between projection operators and firmly nonexpansive operators.

Proposition 3.1 *The projector onto a nonempty closed and convex set A is firmly nonexpansive.*

Proof. Using inequality (3.11), we get

$$(\forall (x,y) \in \Xi^2) \quad \langle x - P_A(x) \mid P_A(y) - P_A(x) \rangle \leq 0. \tag{3.12}$$

Likewise

$$(\forall (x,y) \in \Xi^2) \quad \langle y - P_A(y) \mid P_A(x) - P_A(y) \rangle \leq 0. \tag{3.13}$$

Adding the above two inequalities gives

$$(\forall (x,y) \in \Xi^2) \quad \|P_A(x) - P_A(y)\|^2 \leq \langle P_A(x) - P_A(y) \mid x - y \rangle, \tag{3.14}$$

and the proof is complete. □

A related result is the following.

Proposition 3.2 *[13] A firmly nonexpansive operator T with a fixed point is a projector onto a closed and convex set if and only if* $(\forall x \in \Xi)\ \ T(x) \in \operatorname{Fix} T$.

Definition 3.8 *An operator $T : \Xi \to \Xi$ is called an affine operator if*

$$(\forall (x,y) \in \Xi^2)(\forall \alpha \in \mathbb{R}) \quad T((1-\alpha)x + \alpha y) = (1-\alpha)T(x) + \alpha T(y). \tag{3.15}$$

It is called a linear operator if

$$(\forall (x,y) \in \Xi^2)(\forall \alpha \in \mathbb{R}) \quad T(\alpha x + y) = \alpha T(x) + T(y). \tag{3.16}$$

It can be seen that a linear operator is a special kind of affine operator from the following expression. T is an affine operator if and only if

$$(\forall x \in \Xi) \quad T(x) = L(x) + y, \tag{3.17}$$

where $y \in \Xi$ and L is a linear operator [47].

Definition 3.9 *An affine operator $T : \Xi \to \Xi$ is called an isometry if*

$$(\forall (x,y) \in \Xi^2) \quad \|T(x) - T(y)\| = \|x - y\|. \tag{3.18}$$

Thus, an isometry is nonexpansive.

We summarize the above relationships as follows:

projector

↓

linear isometry contractive firmly nonexpansive

↓ ↙ ↘ ↓ ↙

affine nonexpansive

3.2.2 Firmly Nonexpansive Operators

Now we examine and prove some important properties of firmly nonexpansive operators. An interesting relationship between firm-nonexpansivity and nonexpansivity is exploited.

Proposition 3.3 *Consider a firmly nonexpansive operator* $F : \Xi \to \Xi$ *with a fixed point. Then,*

$$(\forall z \in \operatorname{Fix} F)(\forall x \in \Xi) \quad \langle x - F(x) \mid z - F(x) \rangle \leq 0. \tag{3.19}$$

$$(\forall z \in \operatorname{Fix} F)(\forall x \in \Xi) \quad \langle x - F(x) \mid x - z \rangle \geq \|x - F(x)\|^2. \tag{3.20}$$

$$(\forall z \in \operatorname{Fix} F)(\forall x \in \Xi) \quad \|F(x) - z\|^2 \leq \|x - z\|^2 - \|x - F(x)\|^2. \tag{3.21}$$

Proof. By virtue of (3.8) in Definition 3.6, if z is a fixed point of F, then

$$(\forall x \in \Xi) \ \langle F(x) - z \mid x - z \rangle \geq \|F(x) - z\|^2. \tag{3.22}$$

Hence,

$$(\forall x \in \Xi)\ \langle F(x) - z \mid x - F(x) \rangle = \langle F(x) - z \mid x - z \rangle - \|F(x) - z\|^2 \geq 0. \tag{3.23}$$

This proves (3.19). (3.20) follows from (3.19) since

$$\begin{aligned} \langle x - F(x) \mid z - x \rangle &= \langle x - F(x) \mid z - F(x) \rangle + \langle x - F(x) \mid F(x) - x \rangle \\ &\leq -\|x - F(x)\|^2. \end{aligned} \tag{3.24}$$

Finally, (3.21) follows also from (3.19) since

$$\begin{aligned} \|x - z\|^2 &= \|x - F(x) + F(x) - z\|^2 \\ &= \|x - F(x)\|^2 + \|F(x) - z\|^2 + 2\langle x - F(x) \mid F(x) - z \rangle \\ &\geq \|x - F(x)\|^2 + \|F(x) - z\|^2. \end{aligned} \tag{3.25}$$

□

In the process of applying an operator, one can employ a useful variation called relaxation. It will be shown that relaxations can often help improve the convergence of certain algorithms. Mathematically, for a point $x \in \Xi$, a relaxation of $T(x)$ can be represented as

$$\begin{aligned} T'(x) &= (1 - \lambda)x + \lambda T(x) \\ &= x + \lambda(T(x) - x), \end{aligned} \tag{3.26}$$

where $\lambda > 0$ is called a relaxation coefficient. The range of relaxation varies and is determined so as to guarantee the convergence of the algorithm. If $\lambda < 1$, $T'(x)$ will be a point on the segment $[x, T(x)]$ and we obtain an underrelaxation. If $\lambda = 1$, no relaxation takes place. If $\lambda > 1$, we obtain an overrelaxation. In particular, for $\lambda = 2$, $T'(x)$ will be the mirror image of x with respect to $T(x)$.

Proposition 3.4 *Let $T : \Xi \to \Xi$ be a nonexpansive operator. Then for every β in $[0,1]$, $T' = (1-\beta)\mathrm{Id} + \beta T$ is nonexpansive. Moreover,* $\mathrm{Fix}\, T = \mathrm{Fix}\, T'$.

Proof. By the nonexpansivity of T, for every (x, y) in Ξ^2, we have

$$\begin{aligned} \|T'(x) - T'(y)\| &= \|(1-\beta)(x-y) + \beta(T(x) - T(y))\| \\ &\leq \|(1-\beta)(x-y)\| + \|\beta(T(x) - T(y))\| \\ &= (1-\beta)\|x-y\| + \beta\|T(x) - T(y)\| \\ &\leq (1-\beta)\|x-y\| + \beta\|x-y\| \\ &= \|x-y\|. \end{aligned} \tag{3.27}$$

Hence, T' is nonexpansive. $\mathrm{Fix}\, T = \mathrm{Fix}\, T'$ follows at once from

$$(\forall x \in \Xi)\ x = T(x) \iff (1-\beta)x + \beta T(x) = x. \tag{3.28}$$

□

The following result provides a very important relationship between nonexpansive operators and firmly nonexpansive operators. Our proof is similar to that given in [21].

Proposition 3.5 *An operator $F : \Xi \to \Xi$ is firmly nonexpansive if and only if the operator $T = 2F - \mathrm{Id}$ is nonexpansive. Moreover,* $\mathrm{Fix}\, F = \mathrm{Fix}\, T$. *Alternatively, an operator $T : \Xi \to \Xi$ is nonexpansive if and only if the operator $F = (T + \mathrm{Id})/2$ is firmly nonexpansive.*

Proof. For every operator $T : \Xi \to \Xi$, and for every (x, y) in Ξ^2, we can derive the very useful equality:

$$\|x-y\|^2 - \|T(x) - T(y)\|^2 = 4\langle F(x) - F(y) \mid (x - F(x)) - (y - F(y))\rangle, \tag{3.29}$$

where $F = (T + \mathrm{Id})/2$. (3.29) holds since

$$\begin{aligned}
\|x-y\|^2 &- \|T(x)-T(y)\|^2 \\
&= \|x-y\|^2 + \langle T(x)-T(y) \mid x-y\rangle - \langle x-y \mid T(x)-T(y)\rangle \\
&\quad -\|T(x)-T(y)\|^2 \\
&= \langle T(x)+x-T(y)-y \mid x-y\rangle \\
&\quad -\langle T(x)+x-T(y)-y \mid T(x)-T(y)\rangle \\
&= \langle (x+T(x))-(y+T(y)) \mid (x-T(x))-(y-T(y))\rangle. \qquad (3.30)
\end{aligned}$$

First, let us show that $F = (T + \mathrm{Id})/2$ satisfies (3.8) and, thus, is firmly nonexpansive. Since T is nonexpansive, we have

$$\begin{aligned}
\langle F(x)-F(y) \mid x-y\rangle &= \left\langle \frac{(T(x)-T(y)}{2} + \frac{x-y}{2} \mid x-y \right\rangle \\
&= \frac{1}{2}\langle T(x)-T(y) \mid x-y\rangle + \frac{1}{2}\|x-y\|^2 \\
&\geq \frac{1}{4}\|T(x)-T(y)\|^2 + \frac{1}{2}\langle T(x)-T(y) \mid x-y\rangle \\
&\quad +\frac{1}{4}\|x-y\|^2 \\
&= \frac{1}{4}\|T(x)+x-T(y)-y\|^2 \\
&= \|F(x)-F(y)\|^2. \qquad (3.31)
\end{aligned}$$

Now, observe (3.29): T is nonexpansive if and only if the left-hand side is nonnegative while F is firmly nonexpansive if and only if the right-hand side is nonnegative. Therefore, the nonexpansivity of T is equivalent to the firm-nonexpansivity of F. Finally, by virtue of Proposition 3.4, $\mathrm{Fix}\, T = \mathrm{Fix}\, F$. □

By combining Proposition 3.4 and Proposition 3.5, we obtain the following.

Proposition 3.6 *Let C be a nonempty closed and convex subset of Ξ. For any nonexpansive operator $T : C \to C$, and every α in $]0, 1/2]$, the operator $F = (1-\alpha)\mathrm{Id} + \alpha T$ is firmly nonexpansive. Moreover,* $\mathrm{Fix}\, T = \mathrm{Fix}\, F$.

Proof. Fix $\beta \in]0, 1]$ and let $\alpha = \beta/2$. We have

$$\frac{\mathrm{Id}}{2} + \frac{(1-\beta)\mathrm{Id} + \beta T}{2} = (1-\alpha)\mathrm{Id} + \alpha T. \tag{3.32}$$

By virtue of Proposition 3.4 and Proposition 3.5,

$$\begin{array}{lll} T \text{ nonexpansive} \implies & (1-\beta)\mathrm{Id} + \beta T & \text{nonexpansive} \\ & \Updownarrow & \\ & (1-\alpha)\mathrm{Id} + \alpha T & \text{firmly nonexpansive.} \end{array}$$

Finally, $\mathrm{Fix}\, T = \mathrm{Fix}\, F$ follows directly from Proposition 3.4. ☐

Therefore, we summarize:

sufficiently underrelaxed nonexpansivity

↓

firm nonexpansivity

↓

nonexpansivity.

Proposition 3.6 reveals an important relationship between nonexpansivity and firm nonexpansivity. It also provides a simple procedure for constructing a firmly nonexpansive operator based on any nonexpansive one, while preserving the fixed point set. This is of special interest since algorithms based on firmly nonexpansive operators will be seen to enjoy richer convergence properties.

3.2.3 Nonexpansive Operators and Their Fixed Points

In this section, we discuss basic properties of nonexpansive operators and their fixed points. We start with the problem of the existence of fixed points for nonexpansive operators.

3.2.3.1 Existence of Fixed Points

First, let us examine the existence of fixed points of a single nonexpansive operator. A nonexpansive operator does not necessarily have a fixed point. An example is the translation operator $T: \ x \mapsto x + g$, where $g \in \Xi$ is a nonzero vector, for which $\operatorname{Fix} T = \emptyset$. A general method for constructing fixed point free nonexpansive operators is given in [21, Example 12.4]. On the other hand, a fixed point of a nonexpansive operator may not be unique. An extreme case is $T = \mathrm{Id}$, for which $\operatorname{Fix} T = \Xi$.

Proposition 3.7 *Let $T: \ C \to C$ be a nonexpansive operator, where C is a nonempty, closed, bounded, and convex subset of Ξ. Then,*

i) $\operatorname{Fix} T \neq \emptyset$*;*

ii) $\operatorname{Fix} T$ *is a closed and convex set.*

Proof. The proof of i) can be found in [41] or [63]. ii): From Proposition 3.5, it is equivalent to examine the fixed point set of the firmly nonexpansive operator $F = (\mathrm{Id} +$

$T)/2$. According to (3.19), we have

$$\operatorname{Fix} F = \{z \in \Xi \mid (\forall \in \Xi) \quad \langle x - F(x) \mid z - F(x)\rangle \leq 0\}. \tag{3.33}$$

Indeed, any $z \in \operatorname{Fix} T$ satisfies (3.19) thanks to Proposition 3.3. Conversely, if the inequality in (3.19) holds for some z and every $x \in \Xi$, then it must hold when $x = z$, which yields $\|z - F(z)\|^2 \leq 0$, i.e., $z \in \operatorname{Fix} F$. Now, observe that for every $(y_1, y_2) \in (\operatorname{Fix} F)^2$ and $\alpha \in [0, 1]$, the point $z = (1-\alpha)y_1 + \alpha y_2$ satisfies (3.19) too. Thus, it is a fixed point of F. Hence, $\operatorname{Fix} F$ is convex. To show the closedness of $\operatorname{Fix} T$, let us take an arbitrary sequence $(z_n)_{n\geq 0} \subset \operatorname{Fix} T$ such that $z_n \to z$. We must show that $z \in \operatorname{Fix} T$. Since $(\forall n \in \mathbb{N}) \quad z_n \in \operatorname{Fix} T \subset C$ we have $z \in C$ due to the closedness of C. Therefore, $T(z)$ is well defined. Next, invoking the nonexpansivity of T, we get

$$\begin{aligned} \|T(z) - z\| &= \|T(z) - T(z_n) + z_n - z\| \\ &\leq \|T(z) - T(z_n)\| + \|z_n - z\| \\ &\leq 2\|z - z_n\| \to 0 \quad \text{as } n \to +\infty. \end{aligned} \tag{3.34}$$

Hence, $T(z) = z$. We conclude that $\operatorname{Fix} T$ is closed. □

3.2.3.2 Existence of Common Fixed Points

Even if two nonexpansive operators possess fixed points individually, they may not share any of these fixed points. The following theorem gives a simple sufficient condition for the existence of common fixed points.

Proposition 3.8 *[28] Let C be a closed bounded and convex subset of Ξ and let $(T_i)_{i\in I}$ be a family a commuting nonexpansive operators from C into itself. Then the family $(T_i)_{i\in I}$*

has a common fixed point.

Recall that two operators T_1 and T_2 defined from C into itself commute if

$$(\forall x \in C) \ \ T_1 \circ T_2(x) = T_2 \circ T_1(x). \tag{3.35}$$

Therefore, it is usually a strong condition. Less restrictive existence conditions can be found in [43].

3.2.3.3 The Set of Common Fixed Points

The following result appears to be known (see [35] for the case of composition and [12] for the case of convex combination). We provide here detailed proofs.

Theorem 3.1 *Let $(T_i)_{i\in I}$ be a finite family of m firmly nonexpansive operators from Ξ into Ξ, with fixed point sets $(S_i)_{i\in I}$. Let T_{com} denote their composition and T_{cc} their convex combination, i.e., $T_{cc} = \sum_{i\in I} w_i T_i$, where $(\forall i \in I)\ w_i > 0$ and $\sum_{i\in I} w_i = 1$. Then,*

i) T_{cc} is firmly nonexpansive.

ii) If $S = \bigcap_{i\in I} S_i \neq \varnothing$, then $\operatorname{Fix} T_{com} = \operatorname{Fix} T_{cc} = S$.

Proof. i): The T_is are firmly nonexpansive. Hence, for every (x, y) in Ξ^2, we have

$$\begin{aligned}
\langle T_{cc}(x) - T_{cc}(y) \mid x - y\rangle &= \sum_{i\in I} w_i \langle T_i(x) - T_i(y) \mid x - y\rangle \\
&\geq \sum_{i\in I} w_i \|T_i(x) - T_i(y)\|^2 \\
&\geq \|\sum_{i\in I} w_i (T_i(x) - T_i(y))\|^2 \\
&= \|T_{cc}(x) - T_{cc}(y)\|^2.
\end{aligned} \tag{3.36}$$

The first inequality follows from (3.8) and the second holds due to the convexity of the function $\|\cdot\|^2$. Thus, T_{cc} is firmly nonexpansive. ii): We first show $S \subset \operatorname{Fix} T_{com}$ and $S \subset \operatorname{Fix} T_{cc}$. Let $x \in S$. Then,

$$(\forall i \in I)\ T_i(x) = x. \tag{3.37}$$

Hence, we obtain

$$\begin{cases} T_{com}(x) = T_m \circ T_{m-1} \circ \cdots \circ T_1(x) = x \\ T_{cc}(x) = \sum_{i\in I} w_i T_i(x) = \sum_{i\in I} w_i x = x. \end{cases} \tag{3.38}$$

Therefore, $S \subset \operatorname{Fix} T_{com}$ and $S \subset \operatorname{Fix} T_{cc}$. Conversely, let us first show $\operatorname{Fix} T_{cc} \subset S$. Fix $z \in S$ and $x \in \operatorname{Fix} T_{cc}$. Then $x - T_{cc}(x) = 0$. Hence, using (3.20), we get

$$\begin{aligned} 0 &= \langle x - T_{cc}(x) \mid z - x\rangle \\ &= \langle x - \sum_{i\in I} w_i T_i(x) \mid z - x\rangle \\ &= \sum_{i\in I} w_i \langle x - T_i(x) \mid z - x\rangle \\ &\leq -\sum_{i\in I} w_i \|x - T_i(x)\|^2 \\ &\leq 0. \end{aligned} \tag{3.39}$$

Therefore, $\sum_{i\in I} w_i \|x - T_i(x)\|^2 = 0$, i.e., $(\forall i \in I)\ x = T_i(x)$. We conclude that $x \in \bigcap_{i\in I} S_i$ and thus $\operatorname{Fix} T_{cc} \subset S$. Finally, we show $\operatorname{Fix} T_{com} \subset S$. Fix $z \in S$ and $x \in \operatorname{Fix} T_{com}$. For the sake of simplicity, we only show the case when T_{com} is the composition of three operators, i.e., $T_{com} = T_3 \circ T_2 \circ T_1$. The proof is similar for the general case. Applying (3.21), we obtain

$$\|T_1(x) - z\|^2 \leq \|x - z\|^2 - \|x - T_1(x)\|^2, \tag{3.40}$$

and

$$\|T_2 \circ T_1(x) - z\|^2 \le \|T_1(x) - z\|^2 - \|x - T_2 \circ T_1(x)\|^2, \tag{3.41}$$

and

$$\|T_3 \circ T_2 \circ T_1(x) - z\|^2 \le \|T_2 \circ T_1(x) - z\|^2 - \|x - T_3 \circ T_2 \circ T_1(x)\|^2. \tag{3.42}$$

Inserting (3.40) into (3.41), and (3.41) into (3.42), and noting that $x = T_3 \circ T_2 \circ T_1(x)$, we get

$$\|x - z\|^2 \le \|x - z\|^2 - \|x - T_1(x)\|^2 - \|T_1(x) - T_2 \circ T_1(x)\|^2 - \|T_2 \circ T_1(x) - T_3 \circ T_2 \circ T_1(x)\|^2. \tag{3.43}$$

Thus,

$$\|x - T_1(x)\|^2 + \|T_1(x) - T_2 \circ T_1(x)\|^2 + \|T_2 \circ T_1(x) - T_3 \circ T_2 \circ T_1(x)\|^2 \le 0. \tag{3.44}$$

It must be true that $x = T_1(x)$, which implies $x \in \operatorname{Fix} T_1$. We must also have $T_1(x) = T_2 \circ T_1(x)$, which implies $x = T_1(x) \in \operatorname{Fix} T_2$. Finally we must have $T_2 \circ T_1(x) = T_3 \circ T_2 \circ T_1(x)$, which implies $x = T_2(x) = T_2 \circ T_1(x) \in \operatorname{Fix} T_3$. Therefore, $x \in S$. □

Note that the composition T_{com} of several firmly nonexpansive operators is not necessarily firmly nonexpansive. An example in two dimensional space is $T = P_1 \circ P_2$, where P_1 is the projection onto line $x = y$ and P_2 is the projection onto line $x = 0$. Assume two points $a = (4, 0)$ and $b = (3, 1)$. Then $T(a) = (2, 2)$ and $T(b) = (1.5, 1.5)$. Hence, we have $\langle T(a) - T(b) \mid a - b \rangle = 0$, but $\|T(a) - T(b)\| = 1/\sqrt{2}$. However, for nonexpansive operators, we have a weaker result.

Theorem 3.2 *Let $(T_i)_{i\in I}$ be a finite family of m nonexpansive operators from Ξ into Ξ with fixed point sets $(S_i)_{i\in I}$. Let T_{com} denote their composition and T_{cc} their convex combination, i.e., $T_{cc} = \sum_{i\in I} w_i T_i$, where $(\forall i \in I)\ w_i > 0$ and $\sum_{i\in I} w_i = 1$. Then,*

i) T_{com} and T_{cc} are also nonexpansive.

ii) If $\bigcap_{i\in I} S_i = S \neq \emptyset$, then $\operatorname{Fix} T_{cc} = S$.

Proof. i): Since the T_is are nonexpansive, we have

$$(\forall i \in I)(\forall (x,y) \in \Xi^2)\ \|T_i(x) - T_i(y)\| \leq \|x - y\|. \tag{3.45}$$

Applying the above inequalities, for every (x, y) in Ξ^2, we get

$$\begin{aligned} \|T_{\mathrm{com}}(x) \;\; - &\;\; T_{\mathrm{com}}(y)\| \\ &= \|T_m \circ T_{m-1} \circ \cdots \circ T_1(x) - T_m \circ T_{m-1} \circ \cdots \circ T_1(y)\| \\ &\leq \|T_{m-1} \circ \cdots \circ T_1(x) - T_{m-1} \circ \cdots \circ T_1(y)\| \\ &\leq \cdots \\ &\leq \|T_1(x) - T_1(y)\| \\ &\leq \|x - y\|. \end{aligned} \tag{3.46}$$

To prove the nonexpansivity of T_{cc}, note that for every (x, y) in Ξ^2, we have

$$\begin{aligned} \|T_{\mathrm{cc}}(x) - T_{\mathrm{cc}}(y)\| &= \|\sum_{i\in I} w_i T_i(x) - \sum_{i\in I} w_i T_i(y)\| \\ &\leq \sum_{i\in I} w_i \|T_i(x) - T_i(y)\| \\ &\leq \sum_{i\in I} w_i \|x - y\| \\ &= \|x - y\|. \end{aligned} \tag{3.47}$$

ii): By Proposition 3.5, for every T_i, we have $T_i = 2F_i - \mathrm{Id}$, where the F_is are firmly nonexpansive operators. Moreover, $(\forall i \in I)\ \mathrm{Fix}\, T_i = \mathrm{Fix}\, F_i$. Then,

$$T_{cc} = \sum_{i \in I} w_i (2F_i - \mathrm{Id}) = 2 \sum_{i \in I} w_i F_i - \mathrm{Id}. \tag{3.48}$$

By Theorem 3.1, we have $\mathrm{Fix}\,(\sum_{i \in I} w_i F_i) = \bigcap_{i \in I} \mathrm{Fix}\, F_i = S$ and $\sum_{i \in I} w_i F_i$ is firmly nonexpansive. Observe (3.48): by Proposition 3.5 again, we have $\mathrm{Fix}\, T_{cc} = \mathrm{Fix}\,(\sum_{i \in I} w_i F_i)$. Therefore, $\mathrm{Fix}\, T_{cc} = S$. □

3.3 Iterative Construction of a Fixed Point

The basic numerical problem we will be concerned with is to find a fixed point of a nonexpansive operator and, more generally, a common fixed point of a family of nonexpansive operators. To obtain a fixed point of a single operator T, we start with Banach's 1922 classical theorem.

Theorem 3.3 *[52] Let $T : \Xi \to \Xi$ be contractive and let x be its fixed point. Then for any $x_0 \in \Xi$, the sequence constructed by*

$$(\forall n \in \mathbb{N})\ x_{n+1} = T(x_n) \tag{3.49}$$

converges to x.

The operator in the above theorem is required to be contractive. However, nonexpansive operators are more common in practice. Therefore, iterative schemes based on nonexpansive operators will be more useful. It is important to note that if T is a nonexpansive operator, Theorem 3.3 fails as the iterations (3.49) will not converge to a fixed point in general (consider a rotation T in the euclidean plane). However, convergence can be obtained via suitable underrelaxations, as shown next.

Proposition 3.9 *[22] Suppose that the operator $T : \Xi \to \Xi$ is nonexpansive with a fixed point and that $(\lambda_n)_{n\geq 0}$ is a real sequence in $]0,1[$ such that*

$$(\forall n \in \mathbb{N}) \quad \sum_{n\geq 0} \lambda_n(1-\lambda_n) = +\infty. \tag{3.50}$$

Then every sequence $(x_n)_{n\geq 0}$ generated by the iterative scheme

$$(\forall n \in \mathbb{N}) \quad x_{n+1} = x_n + \lambda_n(T(x_n) - x_n) \tag{3.51}$$

converges to a fixed point of T.

From this result, it is straightforward to derive an iterative scheme for finding a fixed point of a firmly nonexpansive operator.

Theorem 3.4 *Suppose that the operator $F : \Xi \to \Xi$ is firmly nonexpansive with a fixed point and that $(\lambda'_n)_{n\geq 0}$ is a real sequence in $]0,2[$ such that*

$$\sum_{n\geq 0} \lambda'_n(2-\lambda'_n) = +\infty. \tag{3.52}$$

Then every sequence $(x_n)_{n\geq 0}$ generated by the iterative scheme

$$(\forall n \in \mathbb{N}) \quad x_{n+1} = x_n + \lambda'_n(F(x_n) - x_n). \tag{3.53}$$

converges to a fixed point of F.

Proof. By Proposition 3.5, $T = 2F - \mathrm{Id}$ is nonexpansive and $\mathrm{Fix}\, T = \mathrm{Fix}\, F$. By inserting this into (3.51), we get

$$\begin{aligned}(\forall n \in \mathbb{N})\ x_{n+1} &= x_n + \lambda_n(2F(x_n) - x_n - x_n) \\ &= x_n + 2\lambda_n(F(x_n) - x_n) \\ &= x_n + \lambda_n'(F(x_n) - x_n),\end{aligned} \tag{3.54}$$

where $\lambda_n' = 2\lambda_n$. Therefore, $\lambda_n' \in]0,2[$ and $\sum_{n\geq 0} \lambda_n'(2-\lambda_n') = +\infty$. □

3.4 Iterative Construction of a Common Fixed Point

Theorem 3.3 and Theorem 3.4 provide simple ways to build a sequence converging to a fixed point of an operator. If T is the convex combination of a family of nonexpansive operators $(T_i)_{i\in I}$, this iterative scheme, then, suggests a way to build a sequence converging to a point in $\bigcap_{i\in I} \mathrm{Fix}\, T_i$. A common fixed point of a family of firmly nonexpansive operators can be obtained in this fashion through the construction of T, which can be either the composition or the convex combination of $(T_i)_{i\in I}$.

The above iterations allow sequential or fully parallel applications of the operators. More flexible schemes are also possible for firmly nonexpansive operators. The following theorems describe schemes with more general iterations.

Theorem 3.5 *Let $(T_i)_{i\in I}$ be a finite family of firmly nonexpansive operators from Ξ to Ξ such that*

$$S = \bigcap_{i\in I} \mathrm{Fix}\, T_i \neq \varnothing. \tag{3.55}$$

Fix $(\delta, \varepsilon) \in \,]0,1]^2$ *and* $M \subset \mathbb{N}^*$. *Then every sequence* $(x_n)_{n\geq 0}$ *generated by*

$$(\forall n \in \mathbb{N})\ x_{n+1} = x_n + \lambda_n\Big(\sum_{i\in I_n} w_{i,n}T_i(x_n) - x_n\Big), \tag{3.56}$$

converges to a point in S *if, for every integer* n,

a) $\emptyset \neq I_n \subset I$ *and* $(\forall i \in I)\ i \in \bigcup_{k=n}^{n+M-1} I_k$;

b) $\sum_{i\in I_n} w_{i,n} = 1$ *and* $(\forall i \in I_n)\ w_{i,n} \geq (1 - 1_{S_i}(x_n))\delta$;

c) $\varepsilon \leq \lambda_n \leq 2 - \varepsilon$.

The control sequence $(I_n)_{n\geq 0}$ dictates the sets to be activated at each iteration and condition a) ensures that every set is activated at least once every M consecutive iterations. The relaxation range is restricted to $]0,2[$ and it allows flexible use of the algorithm. In [15], we gave a projection operator version of this theorem. Following [12], we will state a more general convergence result. First, we need an intermediate result.

Proposition 3.10 *Let* $(T_i)_{i\in I}$ *be a finite family of firmly nonexpansive operators from* Ξ *to* Ξ *such that (3.55) holds. Fix* $(\delta,\varepsilon) \in \,]0,1[^2$. *Take any sequence* $(x_n)_{n\geq 0}$ *generated by (3.56) where, for every integer* n,

a) $\emptyset \neq I_n \subset I$ *and* $I \subset \bigcup_{k\geq 0} I_{n+k}$;

b) $\sum_{i\in I_n} w_{i,n} = 1$ *and* $(\forall i \in I_n)\ w_{i,n} \geq (1 - 1_{S_i}(x_n))\delta$;

c) $\varepsilon \leq \lambda_n \leq (2-\varepsilon)L_n$ *with*

$$L_n = \begin{cases} \dfrac{\sum_{i\in I_n} w_{i,n}\|T_i(x_n) - x_n\|^2}{\|\sum_{i\in I_n} w_{i,w}T_i(x_n) - x_n\|^2} & \textit{if } x_n \notin \bigcap_{i\in I_n} S_i \\ 1 & \textit{if } x_n \in \bigcap_{i\in I_n} S_i. \end{cases}$$

Then we have

$$(\forall n \in \mathbb{N})(\forall z \in \bigcap_{i\in I_n} S_i)(\forall j \in I_n) \;\; \|x_{n+1} - z\|^2 \leq \|x_n - z\|^2 - \delta\varepsilon^2 \|T_j(x_n) - x_n\|^2. \tag{3.57}$$

Proof: Fix $n \in \mathbb{N}$, $j \in I_n$, and $z \in \bigcap_{i\in I_n} S_i$. Using (3.56) We have

$$\begin{aligned}
\|x_{n+1}-z\|^2 &= \|x_n - z + \lambda_n(\sum_{i\in I_n} w_{i,n}T_i(x_n) - x_n)\|^2 \\
&= \|x_n - z\|^2 + 2\lambda_n \sum_{i\in I_n} w_{i,n}\langle x_n - z \mid T_i(x_n) - x_n\rangle \\
&\quad + \lambda_n^2 \| \sum_{i\in I_n} w_{i,n}(T_i(x_n) - x_n)\|^2 \\
&= \|x_n - z\|^2 + 2\lambda_n \sum_{i\in I_n} w_{i,n}\langle x_n - z \mid T_i(x_n) - x_n\rangle \\
&\quad + \frac{\lambda_n^2}{L_n} \sum_{i\in I_n} w_{i,n}\|(T_i(x_n) - x_n)\|^2 \\
&\leq \|x_n - z\|^2 - \lambda_n(2 - \lambda_n/L_n)w_{j,n}\|(T_j(x_n) - x_n)\|^2.
\end{aligned} \tag{3.58}$$

But, by virtue of b) and c), $\lambda_n(2 - \lambda_n/L_n) \geq \varepsilon^2$, which proves the inequality. □

This result gives much more flexibility to the control sequence $(I_n)_{n\geq 0}$. It is only required that every set be activated repeatedly, but in any order. The relaxation range is also extended to $]0, 2L_n[$. By the convexity of the function $\|\cdot\|^2$, it is easy to see that $L_n \geq 1$. Thus, the relaxation range is greatly extended, which in turn gives very large steps $(\|x_{n+1} - x_n\|)_{n\geq 0}$ and accelerated convergence.

We are now ready to state the main convergence result.

Theorem 3.6 *Every sequence $(x_n)_{n\geq 0}$ generated according to the iterative scheme described in Proposition 3.10 converges to a point in S.*

Proof. By Proposition 3.10, $(\|x_{n+1} - z\|)_{n\geq 0}$ is nonincreasing. Consequently, for a fixed z in S, $(x_n)_{n\geq 0}$ lies in the compact set $B(z, \|x_0 - z\|)$ and it therefore possesses a cluster point x. Hence, there exists a subsequence $(x_{n_k})_{k\geq 0}$ of $(x_n)_{n\geq 0}$ such that $(\|x_{n_k} - x\|)_{n\geq 0}$ converges to zero. If we can prove that $x \in S$, then we can conclude $x_n \to x$ because $(\|x_n - x\|)_{n\geq 0}$ is nonincreasing by (3.57). First note that

$$(\forall n \in \mathbb{N})\ \|x - z\| \leq \|x_n - z\|. \tag{3.59}$$

Now suppose that $x \notin S$ and define $I^+ = \{i \in I \mid x \in S_i\}$, $I^- = I \setminus I^+$, $\alpha = \min_{i\in I^-} \|T_i(x) - x\| \neq 0$ and $\beta = \delta\varepsilon^2$. We can choose a real number γ such that

$$0 < \gamma < \frac{\alpha^2\beta}{(1+4\beta)\alpha + 2\|x - z\|}. \tag{3.60}$$

Since $\gamma < \alpha$ and $\|T_i(x) - x\| \leq d(x, S_i)$ by (3.21), $B(x, \gamma)$ does not intersect with $\bigcup_{i\in I^-} S_i$, that is

$$B(x,\gamma) \bigcap \bigcup_{i\in I^-} S_i = \emptyset. \tag{3.61}$$

Since x is a cluster point of $(x_n)_{n\geq 0}$, there exists an integer p such that $x_p \in B(x, \gamma)$. Now fix an arbitrary $j \in I^-$. Let us show by contradiction that $j \notin I_p$. Suppose $j \in I_p$. Then Proposition 3.10 yields

$$\|x_{p+1} - z\|^2 \leq \|x_p - z\|^2 - \beta\|T_j(x_p) - x_p\|^2. \tag{3.62}$$

But we also have

$$\|x_p - z\| \leq \|x_p - x\| + \|x - z\| \leq \gamma + \|x - z\|. \tag{3.63}$$

On the other hand, since $x_p \in B(x, \gamma)$ and $j \in I^-$, (3.61) yields

$$\|T_j(x_p) - x_p\| \geq \|x - T_j(x)\| - \|x - x_p\| - \|T_j(x) - T_j(x_p)\| \geq \alpha - 2\gamma. \tag{3.64}$$

From (3.60), it is easy to verify that $\alpha > 2\gamma$. Therefore, it follows from (3.60) and (3.62) that

$$\begin{aligned} \|x_{p+1} - z\|^2 - \|x - z\|^2 &\leq (\gamma + \|x - z\|)^2 - \beta(\alpha - 2\gamma)^2 - \|x - z\|^2 \\ &= (\gamma + 2\|x - z\| + 4\alpha\beta)\gamma - 4\beta\gamma^2 - \alpha^2\beta \\ &< (\gamma + 2\|x - z\| + 4\alpha\beta)\gamma - \alpha^2\beta \\ &\leq (\alpha + 2\|x - z\| + 4\alpha\beta)\gamma - \alpha^2\beta \\ &< 0. \end{aligned} \tag{3.65}$$

But the last inequality is incompatible with (3.59). Therefore, $j \notin I_p$. Since j is arbitrary, it follows that $I_p \bigcap I^- = \emptyset$ and $I_p \subset I^+$. Proposition 3.10 and (3.61) then yield $x \in \bigcap_{i \in I^+} S_i \Rightarrow x \in \bigcap_{i \in I_p} S_i \Rightarrow \|x_{p+1} - x\| \leq \|x_p - x\| \Rightarrow x_{p+1} \in B(x, \gamma) \Rightarrow x_{p+1} \notin S_j$. Then, by repeating the same process for index $p+1$ as for p, we obtain $j \notin I_{p+1}$, $x_{p+2} \in B(x, \gamma)$, and $x_{p+2} \notin S_j$. Thus, by induction, we obtain $(\forall k \in \mathbb{N})$ $x_{p+k} \notin S_j$ and $j \notin I_{p+k}$, which violates condition a). It follows that $x \in S$. □

Relaxations not only provide flexibility for the iterative scheme, but also give potential for achieving accelerated iteration. Theorem 3.5 generalizes certain finite dimensional results of [14], [15], [16], [19], [42]. Finally, let us note that Theorem 3.6 is a particular case of a result given in [12].

3.5 Approximate Fixed Points

A fixed point of an operator T is defined by the identity $x = T(x)$. It is often hard to find, for a given x, a T that satisfies this identity (this is the basic problem of fractal coding). It is then of interest to investigate the points achieving approximate fixed properties relative to an operator. In this section, we will address the problem of approximate fixed points.

Definition 3.10 *[52] Let T be an operator from Ξ into Ξ, with nonempty convex and closed fixed point set S. Then, for every $\varepsilon \in [0, +\infty]$, the set*

$$S_\varepsilon = \{x \in \Xi \mid \|T(x) - x\| \leq \varepsilon\} \tag{3.66}$$

is called the ε–fixed point set of T.

Definition 3.11 *If S_ε is closed and convex for every ε, we call T a regular operator.*

When $\varepsilon = 0$, $S_0 = \mathrm{Fix}\,T$. Thus, we call $S_0 = S$ the base set and S_ε its ε–extension.

T is not necessarily regular, even if it is nonexpansive. An example in the euclidean plane is $T = P_1 \circ P_2$, where P_1 is the projection onto the set

$$A_1 = \{x = (x^{(1)}, x^{(2)}) \in \mathbb{R}^2 \mid \ x^{(2)} \leq 0\}, \tag{3.67}$$

and P_2 the projection onto the set

$$A_2 = \{x = (x^{(1)}, x^{(2)}) \in \mathbb{R}^2 \mid \ |x^{(1)}| \geq x^{(2)}\}. \tag{3.68}$$

Then, for $a = (1, 1)$ and $b = (-1, 1)$, we have $\|T(a) - a\| = \|T(b) - b\| = \sqrt{2}/2$ while $\|T((a+b)/2) - (a+b)/2\| = 1$. Thus, set

$$C = \{x \in \mathbb{R}^2 \mid \|T(x) - x\| \leq \sqrt{2}/2\} \tag{3.69}$$

is not convex.

Proposition 3.11 *[47] Consider a function $f : \Xi \to \mathbb{R}$. If f is convex, then the set $\{x \in \Xi \mid f(x) \leq \alpha\}$ is closed and convex, for every $\alpha \in \mathbb{R}$.*

Theorem 3.7 *Consider an operator $T : \Xi \to \Xi$. If T is either an affine operator with $\mathrm{Fix}\, T \neq \emptyset$, or a projector onto a nonempty closed and convex set, then T is a regular operator.*

Proof. By Proposition 3.11, to show convexity and closedness of S_ε, it suffices to show that function f defined by

$$(\forall x \in \Xi)\ f(x) = \|T(x) - x\| \tag{3.70}$$

is convex. We first show the convexity of f. Invoking Definition 3.4, it is equivalent to show that for every (x, y) in Ξ^2 and every α in $[0,1]$, we have

$$\|T((1-\alpha)x + \alpha y) - ((1-\alpha)x + \alpha y)\| \leq (1-\alpha)\|T(x) - x\| + \alpha\|T(y) - y\|. \tag{3.71}$$

Now, assume that T is an affine operator. Then,

$$\begin{aligned} \|T((1-\alpha)x + \alpha y) &- ((1-\alpha)x + \alpha y)\| \\ &= \|(1-\alpha)T(x) + \alpha T(y) - (1-\alpha)x - \alpha y\| \\ &\leq \|(1-\alpha)(T(x) - x)\| + \|\alpha(T(y) - y)\| \\ &= (1-\alpha)\|T(x) - x\| + \alpha\|T(y) - y\|. \end{aligned} \tag{3.72}$$

Next, assume that T is the projection operator onto a nonempty closed and convex set A. Due to the convexity of A, we have

$$(1-\alpha)T(x) + \alpha T(y) \in A. \tag{3.73}$$

Note that in this case $T((1-\alpha)x+\alpha y)$ is the closest point in A to $(1-\alpha)x+\alpha y$. Therefore

$$\begin{aligned} \|T((1-\alpha)x+\alpha y) & - ((1-\alpha)x+\alpha y)\| \\ & \leq \|(1-\alpha)T(x)+\alpha T(y)-((1-\alpha)x+\alpha y)\| \\ & = \|(1-\alpha)(T(x)-x)+\alpha(T(y)-y)\| \\ & \leq (1-\alpha)\|T(x)-x\|+\alpha\|T(y)-y\|. \end{aligned} \quad (3.74)$$

□

Chapter 4

NONEXPANSIVE OPERATORS FOR IMAGE CODING

4.1 Introduction

The basic property of the (possibly nonlinear) operators used in operator theoretic coding is nonexpansivity, which is essential for the convergence results established in Chapter 3. In other terms, these are mappings which do not increase the distance between any two points of the space, that is

$$(\forall (x, y) \in \Xi^2) \ \ \|T(x) - T(y)\| \leq \|x - y\|. \tag{4.1}$$

In fact, our main convergence result – Theorem 3.6 – was stated for the special case of firmly nonexpansive operators, i.e., operators which satisfy the stronger inequality

$$(\forall (x, y) \in \Xi^2) \ \ \|F(x) - F(y)\|^2 \leq \langle F(x) - F(y) \mid x - y \rangle. \tag{4.2}$$

However, it was seen in Proposition 3.5 that a firmly nonexpansive operator F can be constructed from a nonexpansive one T via the formula

$$F = \frac{T + \mathrm{Id}}{2} \tag{4.3}$$

without affecting the fixed point set ($\mathrm{Fix}\,F = \mathrm{Fix}\,T$). We can therefore, without loss of generality, consider only nonexpansive operators; those operators which are not firmly nonexpansive will simply be transformed via (4.3).

We will use x to denote an $N \times N$ grayscale image to be processed. Therefore Ξ is an N^2-dimensional euclidean space. An image x will be regarded as a mapping from $D = \{0, 1, \cdots, N-1\} \times \{0, 1, \cdots, N-1\}$ into $\mathbb{R}_+$. In general, an operator T_i will act either on the whole image when associated with a global property or feature, or on a portion (mostly a block) of the image when associated with a local feature. All operators will be defined as mappings from Ξ into Ξ. Also, we will denote by x^Δ the subimage whose support is $\Delta \subset D$.

4.2 Projectors onto Convex Sets

In this section, T_i is the projector onto a closed and convex feature set

$$S_i = \mathrm{Fix}\,T_i = \{x \in \Xi \mid x \text{ satisfies } \Psi_i\}. \tag{4.4}$$

The projector onto S_i was defined in Definition 3.7 and seen to be firmly nonexpansive in Proposition 3.1. A great deal of image properties or features can be expressed directly in the form of such sets and this framework has been used extensively in image recovery [13], [53], [58], [62]. The implicit assumption here is that projections onto S_i are easily computable.

We will call $\hat{x} = \mathcal{T}(x)$ the discrete Fourier transform of x. In some situations, it will also be possible to replace $\mathcal{T}$ by some other linear transforms commonly used in image processing, e.g., DCT, Hartley, Hadamard, Haar, slant [26], [44] or wavelet transform [59].

4.2.1 Spatial Properties

First of all, we define the set of images with prescribed support $\complement\Delta$ (the complement of $\Delta \subset D$ relative to D), which is

$$S_1 = \{x \in \Xi \mid x^{\Delta} \equiv 0\}. \tag{4.5}$$

The projector onto S_1 is given by

$$(\forall x \in \Xi)(\forall (m,n) \in D) \quad (T_1(x))(m,n) = \begin{cases} 0 & \text{if } (m,n) \in \Delta \\ x(m,n) & \text{else.} \end{cases} \tag{4.6}$$

A property set defined by a more general form of constraint is

$$S_2 = \{x \in \Xi \mid (\forall (m,n) \in \Delta)\ a \leq x(m,n) \leq b\}, \tag{4.7}$$

where $(a,b) \in \mathbb{R}^2$. Note that in blockwise image coding x^{Δ} is a block of the image x. Thus, set S_2 specifies the dynamic range of x within the given block. Usually, a constraint such as $a = b$ can be used to encode a shade area x^{Δ} of the image. The projection of x onto S_1 is given by $(\forall (m,n) \in D)$

$$(T_2(x))(m,n) = \begin{cases} a & \text{if } (m,n) \in \Delta \text{ and } x(m,n) < a \\ b & \text{if } (m,n) \in \Delta \text{ and } x(m,n) > b \\ x(m,n) & \text{else.} \end{cases} \tag{4.8}$$

Another way to generalize (4.5) is

$$S_3 = \{x \in \Xi \mid (\forall (m,n) \in \Delta)\ x(m,n) = g(m,n)\}, \tag{4.9}$$

where g is a known function. The projector onto S_3 is obtained as

$$(\forall x \in \Xi)(\forall (m,n) \in D)\ (T_3(x))(m,n) = \begin{cases} g(m,n) & \text{if } (m,n) \in \Delta \\ x(m,n) & \text{else.} \end{cases} \tag{4.10}$$

The operator T_2 is useful to encode a certain image area which does not display obvious edge features and where there are only small variations in pixel values. This is typically the case for textured areas. On the other hand, T_3 is used to preserve exactly an image feature (especially edge features) specified by g, which will often be a quantized version of the image to be coded in area Δ.

4.2.2 Spectral Properties

The constraints defining the above sets can also be applied in the discrete Fourier transform (DFT) domain.[1] The set corresponding to S_2 in the DFT domain is defined by

$$S_4 = \{x \in \Xi \mid (\forall (k,l) \in \Lambda)\ a' \leq \Re(\widehat{x^{\Delta}}(k,l)) \leq b'\}, \tag{4.11}$$

where x^{Δ} is a subimage and Λ the frequency range of its DFT. The projector onto S_4 is

$$(\forall x \in \Xi)\ T_4(x) = \begin{cases} \mathcal{T}^{-1}(\dot{X}) & \text{on } \Delta \\ x & \text{else,} \end{cases} \tag{4.12}$$

where

$$(\forall (k,l) \in \Lambda)\ \Re(\dot{X}(k,l)) = \begin{cases} a' & \text{if } \Re(\widehat{x^{\Delta}}(k,l)) < a' \\ b' & \text{if } \Re(\widehat{x^{\Delta}}(k,l)) > b' \\ \Re(\widehat{x^{\Delta}}(k,l)) & \text{else.} \end{cases} \tag{4.13}$$

[1] Actually, most of those properties can also be applied to other unitary transform, such as the discrete cosine/sine transform, etc.

Imposing the constraint on the imaginary part of the transform gives rise to

$$S_4' = \{x \in \Xi \mid (\forall (k,l) \in \Lambda)\ a' \leq \Im(\widehat{x^{\Delta}}(k,l)) \leq b'\}. \tag{4.14}$$

A set corresponding to S_3 in the transform domain is defined by

$$S_5 = \{x \in \Xi \mid (\forall (k,l) \in \Lambda)\ \widehat{x^{\Delta}}(k,l) = G(k,l)\}, \tag{4.15}$$

where $G: \Lambda \to \mathbb{C}$ is a known function. The corresponding projector is given by

$$(\forall x \in \Xi)\ T_5(x) = \begin{cases} \mathcal{T}^{-1}(\dot{X}) & \text{on } \Delta \\ x & \text{else,} \end{cases} \tag{4.16}$$

where

$$(\forall (k,l) \in \Lambda)\ \dot{X}(k,l) = G(k,l). \tag{4.17}$$

In transform coding, $G \equiv 0$ over the high frequency range. A set can also be defined through the knowledge of the spectral density of the image over Λ, namely

$$S_6 = \{x \in \Xi \mid (\forall (k,l) \in \Lambda)\ |\widehat{x^{\Delta}}(k,l)| = G(k,l)\}, \tag{4.18}$$

where $G: \Lambda \to \mathbb{R}_+$ is known. This set has been used in the image phase retrieval problem [53]. Unfortunately, S_6 is not convex and it must be replaced by its convex hull

$$S_7 = \{x \in \Xi \mid (\forall (k,l) \in \Lambda)\ |\widehat{x^{\Delta}}(k,l)| \leq G(k,l)\}. \tag{4.19}$$

The associated projector reads

$$(\forall x \in \Xi)\ T_7(x) = \begin{cases} \mathcal{T}^{-1}(\dot{X}) & \text{on } \Delta \\ x & \text{else,} \end{cases} \tag{4.20}$$

where

$$(\forall(k,l)\in\Lambda)\;\; \dot{X}(k,l)=\begin{cases} G(k,l)e^{i\angle\widehat{x^{\Delta}}(k,l)} & \text{if } |\widehat{x^{\Delta}}(k,l)|>G(k,l)\\ \widehat{x^{\Delta}}(k,l) & \text{else.}\end{cases} \tag{4.21}$$

It is also possible to impose constraints on the phase of the DFT coefficients. The set of images whose block x^{Δ} has a prescribed phase φ is defined by

$$S_8=\{x\in\Xi \mid (\forall(k,l)\in\Lambda)\;\angle\widehat{x^{\Delta}}(k,l)=\varphi(k,l)\}. \tag{4.22}$$

The associated projector is defined by

$$(\forall x\in\Xi)\quad T_8(x)=\begin{cases} \mathcal{T}^{-1}(\dot{X}) & \text{on } \Delta\\ x & \text{else,}\end{cases} \tag{4.23}$$

where $\forall(k,l)\in\Lambda$, we define[2]

$$\dot{X}(k,l)=e^{i\varphi(k,l)}|\widehat{x^{\Delta}}(k,l)|\cos^{+}(\angle\widehat{x^{\Delta}}(k,l)-\varphi(k,l)). \tag{4.24}$$

4.2.3 Pattern Properties

To make use of pattern images in coding, it is useful to define a set of images, each of which has a certain block x^{Δ} whose distance to a given reference block r^{Δ} is within a certain value. By doing so, we obtain a closed ball

$$S_9=\{x\in\Xi \mid \|x^{\Delta}-r^{\Delta}\|\leq\rho\}. \tag{4.25}$$

[2]We use the notation

$$\cos^{+}(\phi)=\begin{cases} \cos(\phi) & \text{if } \cos(\phi)\geq 0\\ 0 & \text{else.}\end{cases}$$

In particular, the set containing images with block x^{Δ} of bounded energy is defined by

$$S_9' = \{x \in \Xi \mid \|x^{\Delta}\| \leq \rho\}. \tag{4.26}$$

The projector onto S_9 is defined by

$$(\forall x \in \Xi)\ T_9(x)(m,n) = \begin{cases} r^{\Delta} + \dfrac{\rho}{\|x^{\Delta} - r^{\Delta}\|}(x^{\Delta} - r^{\Delta}) & \text{if } \begin{cases} \|x^{\Delta} - r^{\Delta}\| > \rho \\ (m,n) \in \Delta \end{cases} . \\ x(m,n) & \text{else.} \end{cases} \tag{4.27}$$

A generalization of (4.25) used in [58] and [61] is

$$S_9'' = \{x \in \Xi \mid \|Lx^{\Delta} - r^{\Delta}\| \leq \rho\}, \tag{4.28}$$

where L is a linear transformation. In [61], such an operator was applied in the post-processing of transform coded images to remove blocky effects due to block-wise partitioning and it can therefore be called a smoothness operator. Finally, we consider the set of images whose projection onto a nonempty closed and convex set S_j is $g \in S_j$. We obtain the closed and convex cone

$$S_{10} = \{x \in \Xi \mid P_j(x) = g\}, \tag{4.29}$$

where P_j is the projector onto S_j.

4.2.4 Statistical Properties

Some projectors can be associated with statistical properties of the image to be encoded. Thus, the set of all images whose pixel average over a region Δ is within given bounds is the hyperslab

$$S_{11} = \{x \in \Xi \mid a \leq \sum_{(m,n)\in\Delta} x(m,n) \leq b\}. \tag{4.30}$$

Let $\rho = \sum_{(m,n)\in\Delta} x(m,n)$. Then the projector reads

$$(\forall x \in \Xi)\ T_{11}(x) = \begin{cases} x + \dfrac{b-\rho}{\eta} u_\Delta & \text{if } \rho > b \\ x + \dfrac{a-\rho}{\eta} u_\Delta & \text{if } \rho < a \\ x & \text{else,} \end{cases} \tag{4.31}$$

where u_Δ is the image whose pixel values are 1 if $(m,n) \in \Delta$ and 0 elsewhere, and η the number of pixels within region Δ. The corresponding set defined in the transform domain is given by

$$S_{12} = \{x \in \Xi \mid a' \leq \sum_{(m,n)\in\Lambda} \Re(\widehat{x^\Delta}(k,l)) \leq b'\}. \tag{4.32}$$

The set of images x whose block x^Δ has pixel average μ and variance bounded by B is

$$S_{13} = \{x \in \Xi \mid \sum_{(m,n)\in\Delta} |x(m,n) - \mu|^2 \leq B\}. \tag{4.33}$$

This set is actually a particular case of (4.25) where r is the constant image with pixel value μ.

4.3 Isometries

We discuss here operators related to perceptual features of a block x^Δ, such as symmetry, periodicity, and so on. These operators are isometries, which are affine operators characterized by (3.18). Some of them have proven very useful in fractal coding [4], [18], [23].

We assume that $\Delta = \{\ell_m, \cdots, \ell_m + M - 1\} \times \{\ell_n, \cdots, \ell_n + M - 1\}$, where ℓ_m and ℓ_n are the coordinates of the upper left corner of Δ. First, we consider the property of symmetry of x^Δ about its mid-horizontal axis. The operator which leaves fixed the images with this property is the reflection operator

$$(\forall x \in \Xi)\ T_{14}(x) = \dot{x}, \tag{4.34}$$

where

$$(\forall (m,n) \in D)\ \dot{x}(m,n) = \begin{cases} x(m, M-1-n+2\ell_n) & \text{if } (m,n) \in \Delta \\ x(m,n) & \text{else.} \end{cases} \tag{4.35}$$

The corresponding fixed point set is

$$S_{14} = \{x \in \Xi \mid (\forall (m,n) \in \Delta)\ x(m,n) = x(m, M-1-n+2\ell_n)\}. \tag{4.36}$$

Likewise, symmetry of x^Δ about its mid-vertical axis is associated with the operator

$$(\forall x \in \Xi)\ T_{15}(x) = \dot{x}, \tag{4.37}$$

where

$$(\forall (m,n) \in D)\ \dot{x}(m,n) = \begin{cases} x(M-1-m+2\ell_m, n) & \text{if } (m,n) \in \Delta \\ x(m,n) & \text{else,} \end{cases} \tag{4.38}$$

whose fixed point set is

$$S_{15} = \{x \in \Xi \mid (\forall (m,n) \in \Delta)\ x(m,n) = x(M-1-m+2\ell_m, n)\}. \tag{4.39}$$

Symmetry of x^Δ about its first diagonal is associated with the operator

$$(\forall x \in \Xi)\ T_{16}(x) = \dot{x}, \tag{4.40}$$

where

$$(\forall (m,n) \in D)\ \dot{x}(m,n) = \begin{cases} x(n-\ell_n+\ell_m, m-\ell_m+\ell_n) & \text{if } (m,n) \in \Delta \\ x(m,n) & \text{else.} \end{cases} \tag{4.41}$$

The fixed point set is

$$S_{16} = \{x \in \Xi \mid (\forall (m,n) \in \Delta)\ x(m,n) = x(n - \ell_n + \ell_m, m - \ell_m + \ell_n)\}. \tag{4.42}$$

Likewise, symmetry of x^{Δ} about its second diagonal yields

$$(\forall x \in \Xi)\ T_{17}(x) = \dot{x}, \tag{4.43}$$

where

$$(\forall (m,n) \in D)\ \dot{x}(m,n) = \begin{cases} x(M-1-n+\ell_n+\ell_m, M-1-m+\ell_m+\ell_n) & \text{if } (m,n) \in \Delta \\ x(m,n) & \text{else,} \end{cases} \tag{4.44}$$

with fixed point set

$$S_{17} = \{x \in \Xi \mid (\forall (m,n) \in \Delta)\ x(m,n) = x(M-1-n+\ell_n+\ell_m, M-1-m+\ell_m+\ell_n)\}. \tag{4.45}$$

Invariance under a +90° rotation about the center of the block is described by the rotation operator

$$(\forall x \in \Xi)\ T_{18}(x) = \dot{x}, \tag{4.46}$$

where

$$(\forall (m,n) \in D)\ \dot{x}(m,n) = \begin{cases} x(n-\ell_n+\ell_m, M-1-m+\ell_m+\ell_n) & \text{if } (m,n) \in \Delta \\ x(m,n) & \text{else.} \end{cases} \tag{4.47}$$

The corresponding fixed point set is

$$S_{18} = \{x \in \Xi \mid (\forall (m,n) \in \Delta)\ x(m,n) = x(n-\ell_n+\ell_m, M-1-m+\ell_m+\ell_n)\}. \tag{4.48}$$

For invariance under a $+180°$ rotation about the center of the block, the operator becomes

$$(\forall x \in \Xi)\ T_{19}(x) = \dot{x}, \tag{4.49}$$

where

$$(\forall (m,n) \in D)\ \dot{x}(m,n) = \begin{cases} x(M-1-m+2\ell_m, M-1-n+2\ell_n) & \text{if } (m,n) \in \Delta \\ x(m,n) & \text{else.} \end{cases} \tag{4.50}$$

The fixed point set is

$$S_{19} = \{x \in \Xi \mid (\forall (m,n) \in \Delta)\ x(m,n) = x(M-1-m+2\ell_m, M-1-n+2\ell_n)\}. \tag{4.51}$$

If an image is periodic with period k in the vertical direction, it is invariant under the corresponding translation and we obtain

$$(\forall x \in \Xi)\ T_{20}(x) = \dot{x}, \tag{4.52}$$

where

$$(\forall (m,n) \in D)\ \dot{x}(m,n) = \begin{cases} x(m, ((n+k-\ell_n) \text{ modulo } M) + \ell_n)) & \text{if } (m,n) \in \Delta \\ x(m,n) & \text{else.} \end{cases} \tag{4.53}$$

The resulting fixed points are

$$S_{20} = \{x \in \Xi \mid (\forall (m,n) \in \Delta)\ x(m,n) = x(m, ((n+k-\ell_n) \text{ modulo } M) + \ell_n\}. \tag{4.54}$$

This operator can be generalized to model periodicities in other directions.

4.4 Approximate Fixed Points

In practice, it is unusual to encounters images (or blocks) possessing exactly any of the exact geometrical features described in Section 4.3. Therefore, the degree of approximation to such features should be quantified and the corresponding feature sets should be defined. The sets of Section 4.3 are of the form

$$S_i = \{x \in \Xi \mid T_i(x) = x\} = \mathrm{Fix}\, T_i, \tag{4.55}$$

where T_i is an isometry. If a processed image h does not display a feature Ψ_i exactly, its degree of approximation to this feature is evaluated by $\varepsilon_i = \|T_i(h) - h\|$. Consequently, the set of images which possess at least this degree of approximation to Ψ_i is

$$S_{i,\varepsilon_i} = \{x \in \Xi \mid \|T_i(x) - x\| \leq \varepsilon_i\} = \{x \in \Xi \mid \|(T_i - \mathrm{Id})(x)\| \leq \varepsilon_i\}. \tag{4.56}$$

By Theorem 3.7, we know that S_{i,ε_i} is closed and convex. The projections onto this set can be computed iteratively as in [58] or linearized at every step of the algorithm as in [13].

4.5 Other Operators

A critical property of the operators used in operator theoretic coding is firm nonexpansivity. As shown in Chapter 3, projectors and their underrelaxed versions are firmly nonexpansive. Others operators of interest are approximate projections, such as subgradient projections and surrogate projections [8], [9], [13], [30]. They basically amount to replacing the computation of an exact (and potentially involved) projection by a projection onto a simpler superset such as a half-space. Although such operators are not nonexpansive on the whole space they do not affect the convergence of common fixed point algorithms under mild conditions [13], [30].

In summary, useful operators can be constructed as:

1. Projectors onto closed and convex sets.
2. Relaxed projectors.
3. Approximate projectors.
4. Nonexpansive affine mappings (in particular, isometries).
5. Underrelaxed nonexpansive mappings.
6. Any composition of nonexpansive mappings, in particular of the operators in $1 \sim 4$.
7. Any convex combination of nonexpansive mappings, in particular of the operators in $1 \sim 4$.

If, in the last two cases, the operators are all firmly nonexpansive, then the fixed point set of the composition operators and that of the convex combination operator are the intersection of those of the individual operators by Theorem 3.1.

Chapter 5

OPERATOR THEORETIC CODING

5.1 Introduction

Conventional coding techniques are vector quantization, transform coding, predictive coding, and their combinations. These are noniterative techniques. With the requirement to produce high quality coding techniques achieving high compression ratios, the mathematical theories and algorithms involved are getting more and more sophisticated. Noniterative techniques often lead to tedious computational procedures or rigid designs, which hampers the efficient implementation of coding algorithms. On the other hand, iterative techniques give computationally efficient, schematically simple, and flexible coding.

The initial applications of iterative mapping techniques in image coding have been convex set theoretic coding [38], [50], [54], [56], [61] and fractal image coding [18], [23], [24]. Generally speaking, iterative techniques – even when only partially used and combined with classical techniques – have been shown to improve the coding quality and/or to simplify the computations involved. For example, [61] applied the classical successive projection algorithm POCS [7], [62] in JPEG-related techniques to replace the inversion of the DCT and the subsequent filtering operation in order to more effectively remove artifacts. [38] used POCS in the post-processing of a small codebook VQ in order to get rid of the "staircase" effect in edge blocks. Likewise, [54] used POCS during the decoding process to conceal partially damaged codes.

Overall, the great amount of interest iterative methods have aroused suggests that it is worthwhile to pursue more general schemes. This is precisely the objective of this chapter, in which we systematically develop the notion of operator theoretic coding (OTIC). As discussed in Section 1.3.2, an OTIC procedure involves three basic steps

- Codebook design.
- Encoding.
- Decoding.

We will now discuss these three steps in detail.

5.2 Codebook Design

The codebook is actually the collection of all relevant nonexpansive operators defined on the image space Ξ, such as those described in Chapter 4. They can be roughly classified into:

- Operators related to spatial features.
- Operators related to spectral features.

- Operators related to statistical features.
- Operators related to perceptual features.

Most operators are associated with one or more parameters to be determined during the encoding process. Therefore, an operator is parametrized by its index along with a few parameters. It is then used to represent a basic image property. The images possessing this property form the fixed point set of the operator.

5.3 Encoding

The encoding consists of two basic steps:

1. Determination of the parameters of a family of operator types (projector onto a given type of set, rotation operator, reflection operator, etc.) in the codebook based on the input.
2. Selection of a family of operators $(T_i)_{i \in I}$ from the parametrized operator types based on certain criteria.

5.3.1 Parametrization

An operator in an operator theoretic code must be specified by indexing and parametrization in order to be recognized by the decoder. For instance, the index will identify a projector onto a ball (operator type) while the parameters will specify the center and the radius of the ball. For most operators, the companion parameters must be determined in terms of a specific input image or block. Let us now detail this process for the types of operators discussed in Chapter 4.

1. Projector onto S_1: Region of Support

   Only the domain Δ needs to be specified.

2. Projector onto S_2: Dynamic Range of Pixels

   We need to specify the dynamic range over a given area Δ. The parameters a and b represent the lower bound and the upper bound respectively. In block-wise image coding, Δ may stand for the processing block. For a shade (or plain) area, we take a and b both equal to the average of the pixel values; for a midrange and edge area, however, a and b must be determined individually.

3. Projector onto S_3: Feature Match

   The parametrization of T_3 may require a function and a domain. The parametrization of this operator requires a lot of bits. To reduce the number of bits, a quantized representation is required. We refer to [48] for the problem of modeling image features through parametrized functions.

4. Projector onto S_4: Dynamic Range of Transform Coefficients; Band Limitation

   This operator is used to extract features in the transform domain. It is especially useful for transformations which produce real transform coefficients, e.g., discrete cosine transform, discrete sine transform, and Hartley transform. As for T_2, the parametrization of T_4 is done by specifying a, b, and Δ. This provides a dynamic range for the transform coefficients over a certain frequency band.

5. Projector onto S_5: Spectral Match

   Same as for T_3.

6. Projector onto S_7: Spectrum Constraint

   The parametrization here pertains to the function G. G could be approximated by simple functions or constants within several different frequency ranges in order to save bits.

7. Projector onto S_8: Phase Constraint

   The phase property could be specified by functions in either numerical or analytical form. This is in general expensive in terms of bits. In some cases, e.g., linear phase, it can be efficiently parametrized and quantized.

8. Projector onto S_9: Similarity to Reference Patterns

   Here, an input image – or block thereof – is similar to a pre-stored reference pattern. The measure of similarity can parametrized to describe a property set in which all image have at least the same degree of similarity to the pattern p. The similarity is parametrized as $\rho = \|h - r\|$. These operators can be utilized to generalize vector quantization techniques, where image patterns are codeword vectors in the codebook. We thus obtain a set theoretic codebook made up of balls. This approach is discussed in Section 5.5.1.

9. Projector onto $S_{9'}$: Energy Boundedness

   The parameter to be determined is the energy of the input h. S_9'' is a more general set which can represent many other features. For instance, in block-wise image coding, L can be constructed to constrain the energy of blocks representing borders so as to remove block-artifacts.

10. Projectors onto S_{11}, S_{12}, and S_{13}: Statistical Properties

    They are used to control the statistical properties of decoded images. Especially, S_{13} can also be used to control the smoothness of the decoded image.

11. Isometries: Perceptual Feature Match

    Since in image coding the indices of isometries are used to specify each of them, the isometries do not need to be parametrized except periodicity. If there is no exact match between a feature of the input image h and the operator, one has to employ the projector onto (4.56) and, therefore, to specify the parameter ε_i. For any given isometry T_i, $\varepsilon_i = \|T_i(h) - h\|$ is viewed as a measure of similarity between the feature described by the isometry and that borne by h.

5.3.2 Selecting Operators

After parametrization, one needs to select the useful operators and drop those contributing only marginally to the description of the features. For example, in the case of 256 gray level images, for T_2, $a = 0$ and $b = 255$ in (4.7). As another example, if for an isometry T_i only approximate features are extracted and the distortion measure ε_i is too large in (4.56), then T_i is not a good pattern to describe the input image. In such cases, those operators will not be included to the OTIC code.

5.3.3 Code Representation

The OTIC code consists of two components: the operator types and the scalar parameters associated with them. The operators are referred to according to their indexes in the codebook along with the specified parameters in a quantized form.

5.3.4 Entropy Coding

Image coding is usually completed by a last compression step – entropy coding. Entropy coding is a reversible process whose purpose is to effectively use bits for communications or storage. The idea is to find a reversible mapping to a new set of parameters such that the average number of bits per symbol is minimized. Commonly used techniques are Huffman coding, adaptive entropy coding, and run-length coding [6], [20]. While entropy coding is a very important step that further improves compression rates, it is well understood and will not be discussed further.

5.4 Decoding

5.4.1 Generalities

The objective of OTIC decoding is to find a common fixed point of the operators $(T_i)_{i\in I}$ representing the image, that is

$$\text{Find} \quad x \in \Xi \quad \text{such that} \quad (\forall i \in I) \;\; x = T_i(x) \tag{5.1}$$

or, equivalently,

$$\text{Find} \quad x \in \bigcap_{i \in I} \operatorname{Fix} T_i. \tag{5.2}$$

The mathematical foundations of OTIC decoding are the iterative mapping techniques described in Chapter 3. Starting from an arbitrary image x_0, one builds a sequence $(x_n)_{n \geq 0}$ converging to a solution of (5.1), i.e., to a common fixed point of $(T_i)_{i \in I}$. There are a variety of ways to realize the iterations: sequential applications of $(T_i)_{i \in I}$, fully parallel applications, or block-parallel applications [7], [8], [11], [12], [14], [13], [19], [30]. Here, we discuss three iterative decoding schemes derived directly from the theoretical results of Chapter 3.

5.4.2 Sequential Decoding

Let $I = \{1, \cdots, m\}$. In the case of composition, i.e.

$$T = T_1 \circ T_2 \circ \cdots \circ T_m, \tag{5.3}$$

T is also nonexpansive by Theorem 3.2. However, in general, we merely have $\operatorname{Fix} T \supset \bigcap_{i \in I} \operatorname{Fix} T_i$ and the limit of algorithm (3.50)-(3.51) in this case may not solve (5.1). For (5.6) to be satisfied, we need to replace the nonexpansive operators $(T_i)_{i \in I}$ by the averaged firmly nonexpansive operators $(F_i)_{i \in I}$ of (4.3). Hence, we obtain the sequential algorithm

$$(\forall n \in \mathbb{N}) \; x_{n+1} = x_n + \lambda_n \left(\left((\frac{\operatorname{Id} + T_1}{2}) \circ (\frac{\operatorname{Id} + T_2}{2}) \circ \cdots \circ (\frac{\operatorname{Id} + T_m}{2}) \right) (x_n) - x_n \right), \tag{5.4}$$

where (5.8) is satisfied. In this scheme, only one operator can be used at each step and therefore it is not well suited for parallel implementation. On the other hand, it has low memory requirements as only one vector needs to be stored at any step.

5.4.3 Fully Parallel Decoding

Let $(T_i)_{i\in I}$ be a finite family of nonexpansive operators selected by the encoder, say $I = \{1, \cdots, m\}$. By Theorem 3.2, we can construct an aggregated nonexpansive operator through convex combination, i.e.

$$T = \sum_{i\in I} w_i T_i \quad \text{where} \quad \sum_{i\in I} w_i = 1 \quad \text{and} \quad (\forall i \in I)\; w_i > 0. \tag{5.5}$$

We then have

$$\operatorname{Fix} T = \bigcap_{i\in I} \operatorname{Fix} T_i. \tag{5.6}$$

It follows from Proposition 3.9 that the iterative scheme

$$(\forall n \in \mathbb{N})\;\; x_{n+1} = x_n + \lambda_n\Big(\sum_{i\in I} w_i T_i(x_n) - x_n\Big), \tag{5.7}$$

where

$$(\forall n \in \mathbb{N})\;\; \lambda_n \in \,]0,1[\quad \text{and} \quad \sum_{n\geq 0} \lambda_n(1-\lambda_n) = +\infty \tag{5.8}$$

will converge to a point in $\operatorname{Fix} T$, i.e., to a solution of (5.1). Note that λ_n is the relaxation parameter of the operator T at iteration n. It provides the flexibility of applying variable underrelaxations over the iterations. The weights $(w_i)_{i\in I}$ may have an influence on the speed of convergence of the algorithm. In general they will be taken equal, i.e.,

$$(\forall i \in I)\;\; w_i = 1/m. \tag{5.9}$$

Of course, this algorithm can easily be implemented on a parallel processor architecture provided that the total number of processors is at least equal to the number m of operators.

5.4.4 Block-Iterative Decoding

We now present the general iterative method that will be used in OTIC, which is based on Theorem 3.6. We assume that the operators $(T_i)_{i\in I}$ are firmly nonexpansive (if not, then the averaging operation (4.3) should be performed first) and call $(S_i)_{i\in I}$ their fixed point sets.

Recall from Proposition 3.10 that the basic iteration from step n to step $n+1$ is

$$x_{n+1} = x_n + \lambda_n(\sum_{i\in I_n} w_{i,n}T_i(x_n) - x_n), \tag{5.10}$$

where

a) $\emptyset \neq I_n \subset I$ and $I \subset \bigcup_{k\geq 0} I_{n+k}$;

b) $\sum_{i\in I_n} w_{i,n} = 1$ and $(\forall i \in I_n)\ w_{i,n} \geq (1 - 1_{S_i}(x_n))\delta$;

c) $0 < \varepsilon \leq \lambda_n \leq (2-\varepsilon)L_n$ with

$$L_n = \begin{cases} \dfrac{\sum_{i\in I_n} w_{i,n}\|T_i(x_n) - x_n\|^2}{\|\sum_{i\in I_n} w_{i,w}T_i(x_n) - x_n\|^2} & \text{if } x_n \notin \bigcap_{i\in I_n} S_i \\ 1 & \text{if } x_n \in \bigcap_{i\in I_n} S_i. \end{cases} \tag{5.11}$$

This scheme allows partially parallel applications of the operators. At iteration n, one selects a block of operators $(T_i)_{i\in I_n}$. If I_n is a singleton, one obtains a sequential scheme, if $I_n = I$, one obtains a fully parallel scheme. This gives much flexibility to take advantage of a parallel architecture when the number of processors is less than that of the operators. The only restriction is quite mild: each operator must be repeatedly used (see a)). The extended relaxation range c) goes well beyond $]0,2[$ and allows very fast convergence. This point had already been observed in the case of projection methods [13], [14], [42]. In practice, L_n can be very large, reaching up to hundreds or even thousands, which results in huge iteration steps.

5.5 Relation to Existing Theories and Techniques

OTIC generalizes several coding schemes. Here are some examples.

5.5.1 Vector Quantization

Vector quantization, can be regarded as a particular case of set theoretic coding in the sense that the codebook can be made up of sets as opposed to points as in standard vector quantization.

Assuming that a small codebook is used, codevectors are then sparsely scattered in a high dimensional image space. Consequently, it will be difficult for only one of those codevectors to faithfully represent a somewhat arbitrary input. Our solution here, is to generalize the conventional framework to a set theoretic vector quantization scheme as follows [45]. One chooses appropriate codevectors to define hyperballs $(S_i)_{i\in I}$, which include the input h of the decoder to represent h. The input is thus encoded into the intersection S of those simple sets. The decoding operation consists of finding an image in S.

It is also possible to incorporate some other *a priori* knowledge about the input such as information about the pixels' range, region of support, smoothness, or spectral information. One simply has to add more convex sets, each of which contains all the signals bearing the property in question. Those sets are then combined with $(S_i)_{i\in I}$ in the iterative decoding scheme, which will yield a smaller intersection S and, thereby, an even higher coding quality.

5.5.2 Fractal Coding

Here the family of nonexpansive operators $(T_i)_{i \in I}$ reduces to a single contraction and the iterative scheme (5.10) reduces to the successive approximations scheme (2.3).

5.5.3 Set Theoretic Coding

Here the family of nonexpansive operators $(T_i)_{i \in I}$ reduces to projectors onto convex sets and, at least in the majority of applications presented so far in the literature, the iterative scheme (5.10) has been used in the form of the POCS algorithm (2.8).

Chapter 6

OTIC CODING: SIMULATION RESULTS

6.1 Scope

We proceed in this chapter with a numerical implementation of the proposed OTIC coding scheme in which a mixture of nonexpansive operators is used. The design of an automated encoder is elaborated. The OTIC coder is very flexible in many respects and the implementation shown here represents just one of many possible options. These simulations are representative of the performance of OTIC to the extent that no attempt has been made to adapt the various parameters available to optimize performance relative to the type of images processed. Our main purpose is merely to demonstrate the generality and flexibility of OTIC.

Since OTIC is quite different from conventional coding techniques, it is not easy to compare it with them in terms of coding quality and complexity. Therefore, we will compare it only with fractal coding, which shares similar features.

6.2 Construction of an OTIC Code

6.2.1 Codebook Design

The codebook is a collection of nonexpansive operators available for representing local and global image features or properties. The more operators there are, the more bits are required for indexing purposes. In some cases, different classes of image blocks are coded based on different subsets of the codebook. In general, blocks with more complex features require more operators in order to effectively capture details, while blocks with simple perceptual features are less demanding in that respect.

6.2.2 Encoding

Encoding is the process of choosing a family of operator types from the codebook and of specifying their parameters to obtain a selective family of operators $(T_i)_{i\in I}$. A basic consideration here is the quantization of the parameters. The coarser the quantization, the lower the bit rates and the higher the coding distortion. Another consideration is to achieve a small feasibility set $\bigcap_{i\in I} \operatorname{Fix} T_i$ to minimize the reconstruction error.

6.2.3 Decoding

As seen in Section 5.4.4, the block-iterative decoding scheme offers great flexibility in terms of choosing at each iteration:

- The block of operators.
- The relaxation parameters.

- The weights on the operators.

We will therefore build our decoder around it.

6.3 Implementation

6.3.1 Preliminaries

We use the $N \times N$ ($N = 512$), 8 bits/pixel images "lena" and "peppers" shown in Figs. 6.1 and 6.2 on pages 77 and 78. The operator theoretic image codes are obtained through the encoding procedure described in Section 5.3. The decoding scheme consists of iterating the resulting operator code $(T_i)_{i \in I}$ on any initial image x_0, according to the powerful block-iterative parallel algorithm of Section 5.4.4. We will address in the specific context of this implementation the basic procedures an OTIC scheme involves, namely the preprocessing, the codebook design, the encoding, and finally the decoding. Here, the preprocessing and encoding processes are implemented in an automatic manner. In a specific implementation of OTIC, the preprocessing involves the selection of a distortion measure, the partitioning of the image, and the image block classification. We elaborate them as follows.

6.3.2 Quality Measure

The distortion between an original image h and another image x is measured via the peak signal-to-noise ratio (PSNR)

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{N^2 Dr(h)^2}{\|h - x\|^2} \right), \tag{6.1}$$

where $Dr(x)$ denotes the dynamic range of x, and $\|\cdot\|$ is the euclidean distance.

6.3.3 Image Partition

We use a square, block-wise coding approach (other shapes are also possible). The square support of the original image is partitioned into nonoverlapping square blocks. The selection of sizes for these blocks depends on several factors. Small image blocks are easy to analyze and encode accurately. On the other hand, large blocks allow a better exploitation of the redundancies in smooth image areas and lead to high compression ratios.

Although each operator is theoretically defined on the whole image space Ξ, some of them will act only on subblocks. Indeed, block-wise processing tends to more effectively exploit the local features without imposing heavy duty computations.

6.3.4 Block Classification

As in many image coding techniques, the specific coding procedure varies according to the class the processed block belongs to. Usually, blocks are classified based on the degree of complexity of their features. For instance, fractal coding classifies image blocks into three groups: edge, midrange, and shade. A shade block is "smooth" with no significant gradient. An edge block presents a strong change of intensity across a curve – often a portion of object boundary which runs through the block.

The encoding method applied to an image block depends on its features. Using the thorough study of image block classification done in [46], we consider two types of blocks: shade blocks and edge blocks.

We use standard edge detection tools to classify the image blocks. Specifically, we apply the so-called Sobel operators, namely

$$\frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{6.2}$$

to check the possible row gradients on processed blocks and

$$\frac{1}{4}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{6.3}$$

to check the possible column gradients [44].

6.3.5 Codebook Design

The global image codebook consists of the union of the block codebooks applied to each of the image blocks in the coding. Again, for the sake of convenience, we need only to focus on blockwise coding and assume that the operators are defined on a block only. Theoretically, they can then trivially be extended to nonexpansive operators on the whole space Ξ.

The following codebook will be used for edge block coding. Coding for shade blocks is very simple and will be described in Section 6.3.6. In our codebook, we include the seven operators defined in Section 4.2, which can be indexed with 3 bits. For each edge block input, all of the seven codewords will be selected and then the corresponding parameters will be determined.

- The projection operator P_1 is defined as T_2 in (4.8) to model the dynamic range of the image blocks in the pixel domain. Two parameters need to be determined, the upper bound b and the lower bound a.

- There are five operators used to define the dynamic range of the DCT in this specific implementation. P_2 is the projection operation defined as T_4 in (4.12) to define the dynamic range of the DCT of the processed block. The parameters a' and b' will be determined and quantized to 128 levels. Moreover, we impose a specific constraint on the dc coefficient for each block in order to control the average pixel value, which is the visual brightness of the block. The corresponding parameter dc will be quantized to 64 levels. The other four operators $P_3 \sim P_6$ will be used to capture local details. We further subdivide an edge block into 4 equal size subblocks. Each of the operators P_3, P_4, P_5, P_6 reflects the same kind of constraint as P_2 does, but on one the subblocks.
- The last projection operator P_7 defined as T_9' in (4.27) which imposes energy boundedness. The energy ρ^2 will be determined and then quantized to 128 levels.

6.3.6 Encoding

We partition the whole image into 8×8 processing blocks. For shade blocks, the coding is simple. The whole block is encoded into the average pixel intensity q. The corresponding decoding block is recovered by assigning each pixel the value q. In our simulations, q is quantized to 64 levels, i.e., it requires 5 bits for storage. For edge blocks, we apply the codebook described in Section 6.3.5. One then has to determine the parameters associated with each operator.

6.3.7 Decoding

We apply the block-iterative parallel scheme described in Section 5.4.4. The weights are set equal, I_n is chosen so that only active constraints are selected, i.e., $w_{i,n} = 1/\text{card } I_n$, where $I_n = \{i \in I \mid a_n \notin S_i\}$. The relaxation scheme is $(\forall n \in \mathbb{N})\ \lambda_n = 1.8L_n$.

6.3.8 Calculation of Bit Rates

The calculation of bit rates for the OTIC scheme implemented here is as follows. As before, let $N \times N$ be the image size. An image is partitioned into processing blocks. Assume N_s of them are classified as shade and N_e as edge. Shade blocks are quantized to L_s scalar levels. For edge block l, there are n_l operators selected from the codebook. Operator i_l takes m_{i_l} bits. For operator T_{i_l}, m_{i_l} bits are used to store the associated parameters. Accordingly, the bit rate is

$$Br = \frac{N_s(\log_2 L_s + 1) + \sum_{l=1}^{N_e}(\sum_{i=1}^{n_l} m_{i_l})}{N^2}. \tag{6.4}$$

6.3.9 Complements on Fractal Image Coding

A overview of fractal image coding was given in Section 2.5. Basically, fractal image coding is a single contractive operator based technique whose implementation will differ according to the selected encoding method. Here we follow Jacquin's implementation presented in [23] and [24]. The decoding is actually very simple and involves only the repeated applications of the single operator code according to (2.3). Fractal encoding proceeds in a blockwise manner and for each block it consists of the determination of a contractive operator and the specification of a domain block. The contractive operator is an affine transformation which is the composition of spatial contraction, luminance shift, scaling, and an isometry, while the domain block is a block in the original image with a size bigger than the processing block. Suppose that, for input image block h, $(T_i)_{i\in I}$ are all the possible contractive operators constructed based on the above basic operations, and $(b_j)_{j\in J}$ are all the available domain blocks. The encoder has to compare among all the operators and domain blocks to get the best $T_{\rm i}$ and $b_{\rm j}$ such that

$$\|h - T_{\rm i}(b_{\rm j})\| = \min_{i\in I, j\in J}\{\|h - T_i(b_j)\|\}. \tag{6.5}$$

Then $b_{\rm j}$ is considered to have the best similarity to h and $(T_{\rm i}, b_{\rm j})$ will constitute the fractal code for block h.

6.4 Results and Analysis

As can be seen from the images in Figs. 6.4, 6.6, and 6.8, the images recovered from OTIC provide a good representation of the original ones. In particular, textures are well preserved. Some "blockiness" artifacts are visible, as should be expected from the type of memoryless block coding method used here. This is due to the use of square image blocks. However, the reconstruction is almost free of edge degradations in the form of "staircase effects". Comparing Fig. 6.7 with Fig. 6.8, it can be seen that the OTIC coder gives a smoother image than the fractal coder. The PSNR measure (6.1) confirms that the OTIC coder gives decoded images numerically closer to the input image than the fractal coder.

By comparing Figs. 6.3 and 6.5, one can see that the coding quality varies as the decoding proceeds from different points although, theoretically, the fixed point is unique. This is due to the fact that the iterations are truncated after a finite number of steps, and therefore the fixed point is not attained. Thus, the quality of a fractal coder is dependent on the initial point in practice. Notice that the large block artifacts in Fig. 6.5 are due to the fact that the image is initially broken down into four pieces and then encoded.

In terms of complexity, OTIC encoding is simpler than fractal encoding since the latter requires the determination of the parameters of the selected operators. On the other hand, fractal encoding requires an exhaustive search and comparisons of various so-called domain blocks and contractive affine operators [23]. On the average OTIC decoding usually involves more iterations but an exact comparison of the numerical load of a single iteration is complicated as the OTIC decoder is trivially parallelizable while the fractal decoder is not.

As in fractal image coding, the quality of an OTIC scheme relies heavily on block classification. Many of the artifacts visible in the decoded images are partially due to wrong block classification. More detailed classifications and coding methods may be necessary to improve the overall coding quality. Quality could also be further improved by including global operators operating on the whole image as an enhancement device.

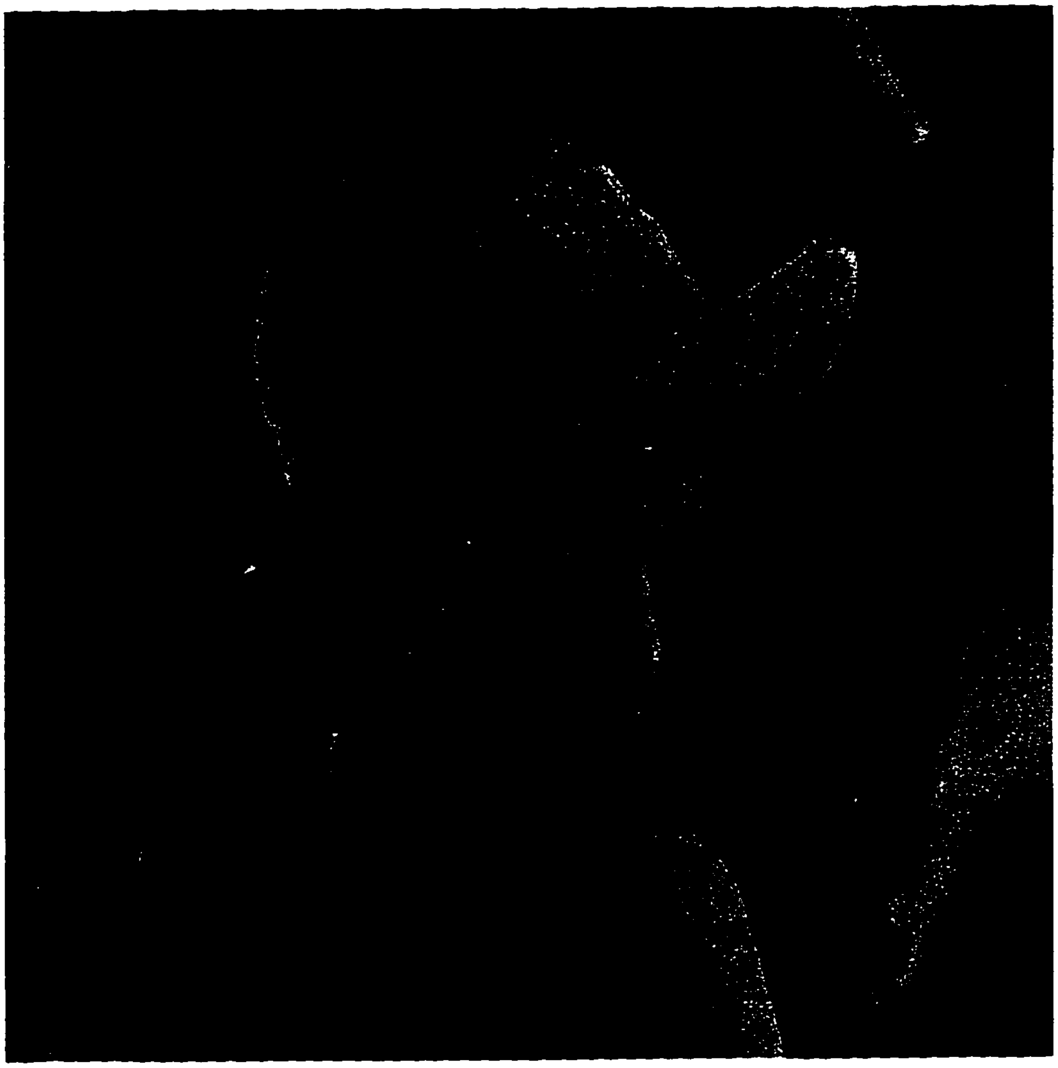

Figure 6.1: Original "lena" image.

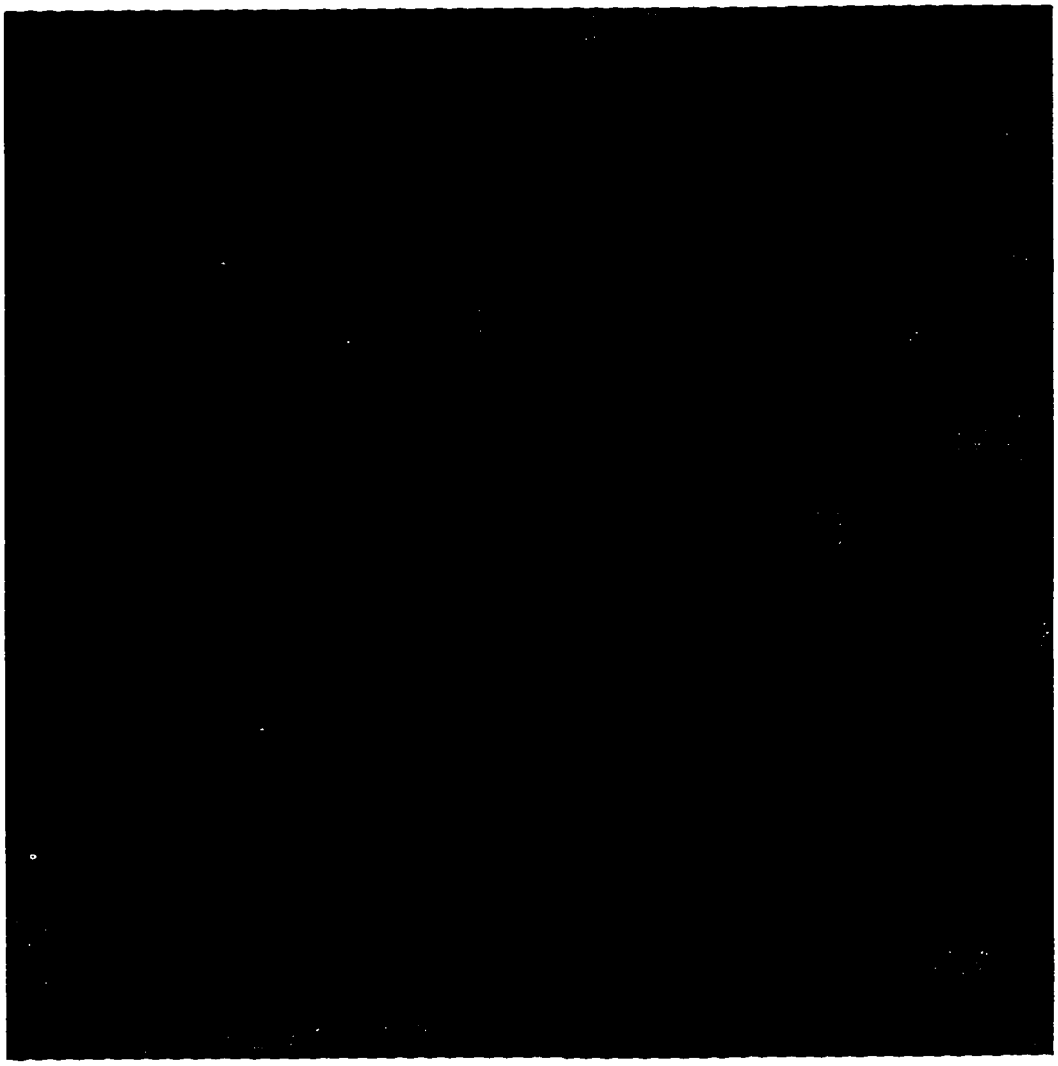

Figure 6.2: Original "peppers" image.

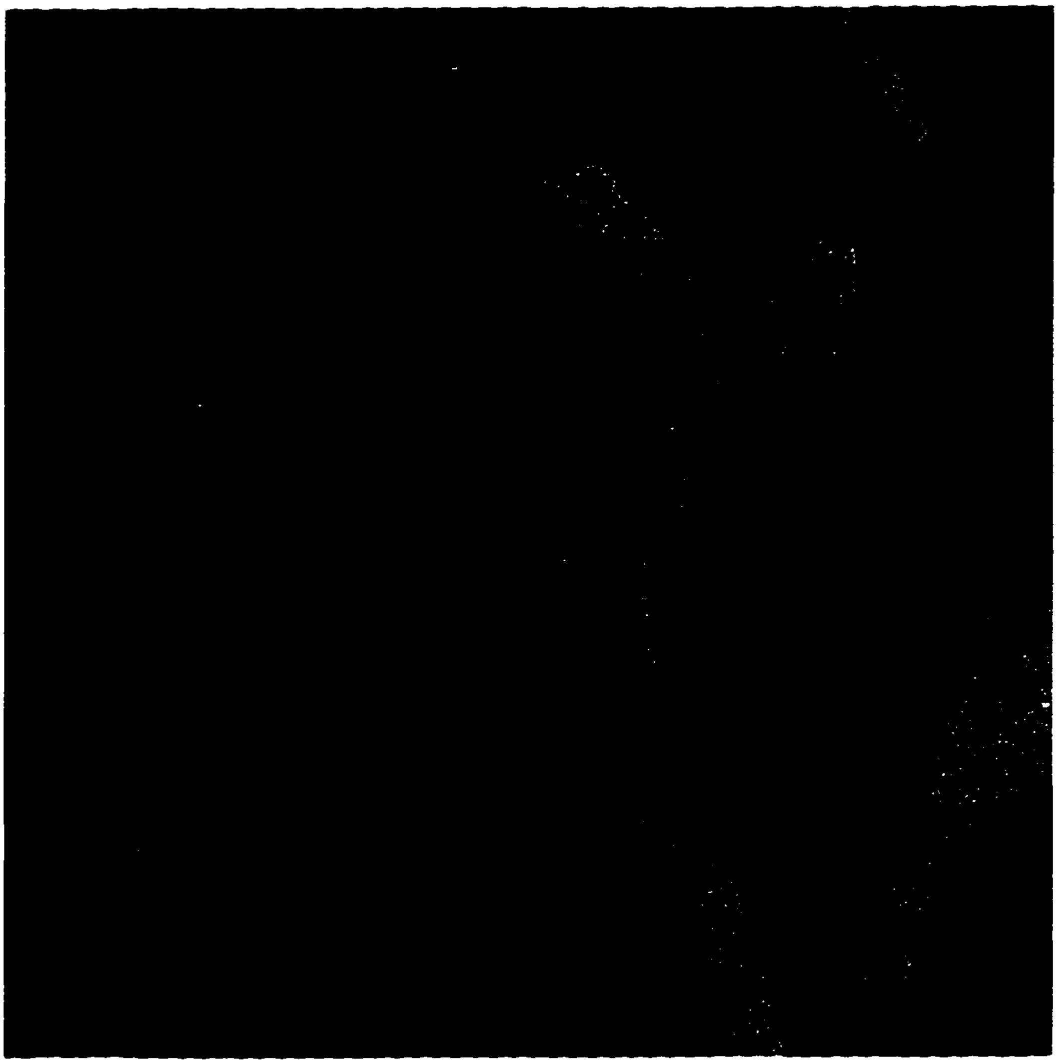

Figure 6.3: Fractal coding of "lena". Image decoded after 60 iterations, initialized at the "black" image. Block size 8 $\times$ 8. PSNR= 29.4dB, 0.6 bpp.

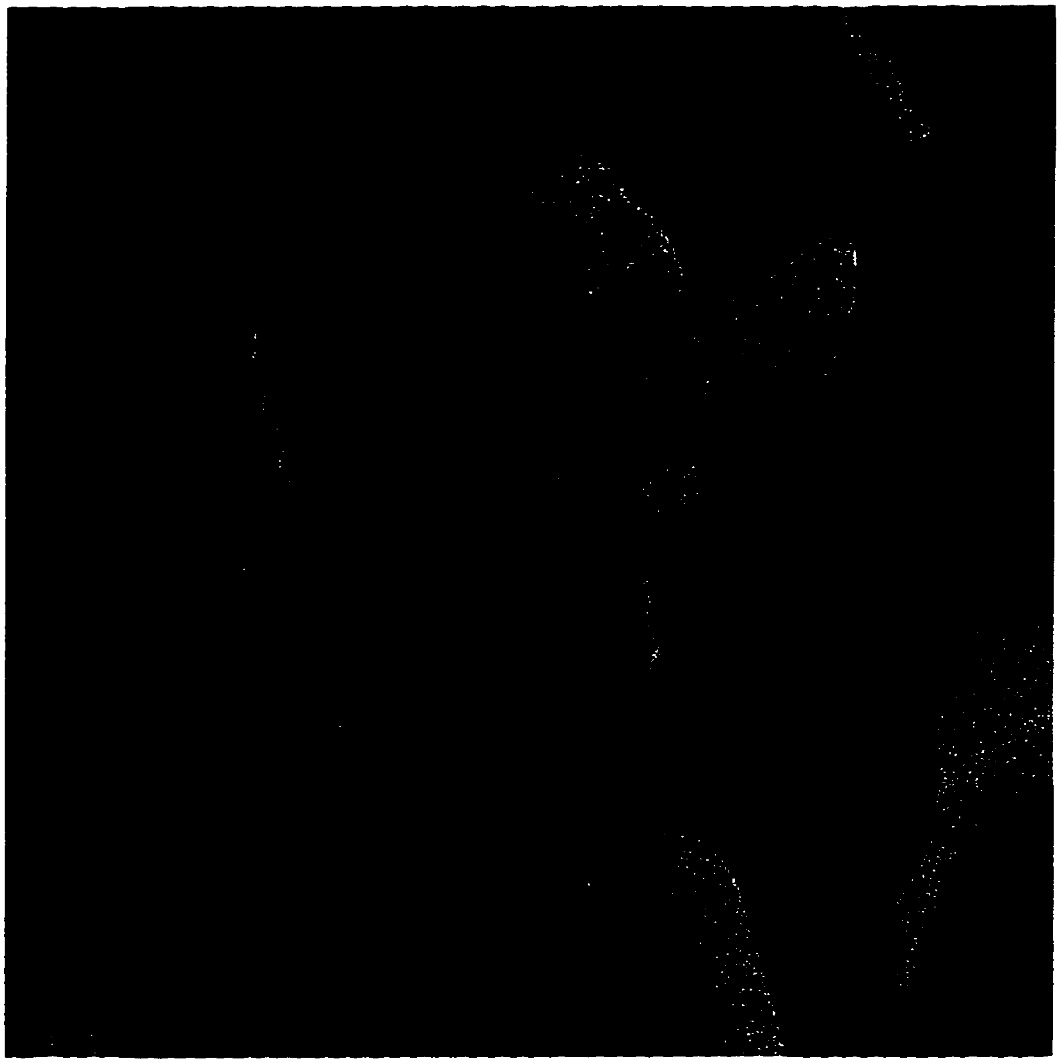

Figure 6.4: OTIC coding of "lena". Image decoded after 60 iterations, initialized at the "black" image. Bock size 8 × 8. PSNR= 30.6dB, 0.85 bpp.

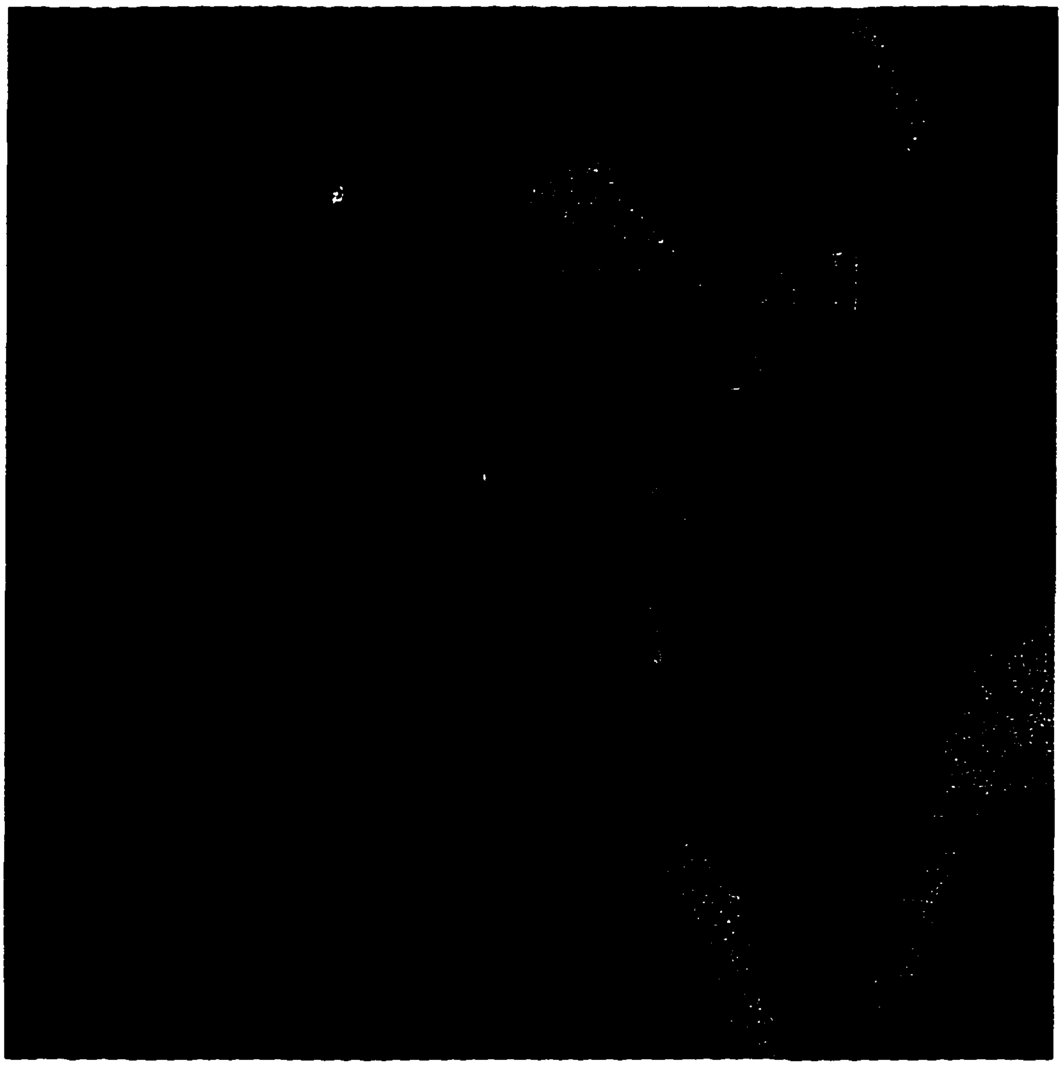

Figure 6.5: Fractal coding of "lena". Image decoded after 60 iterations, initialized at the "peppers" image. Block size 8×8. PSNR= 26.2dB, 0.6 bpp.

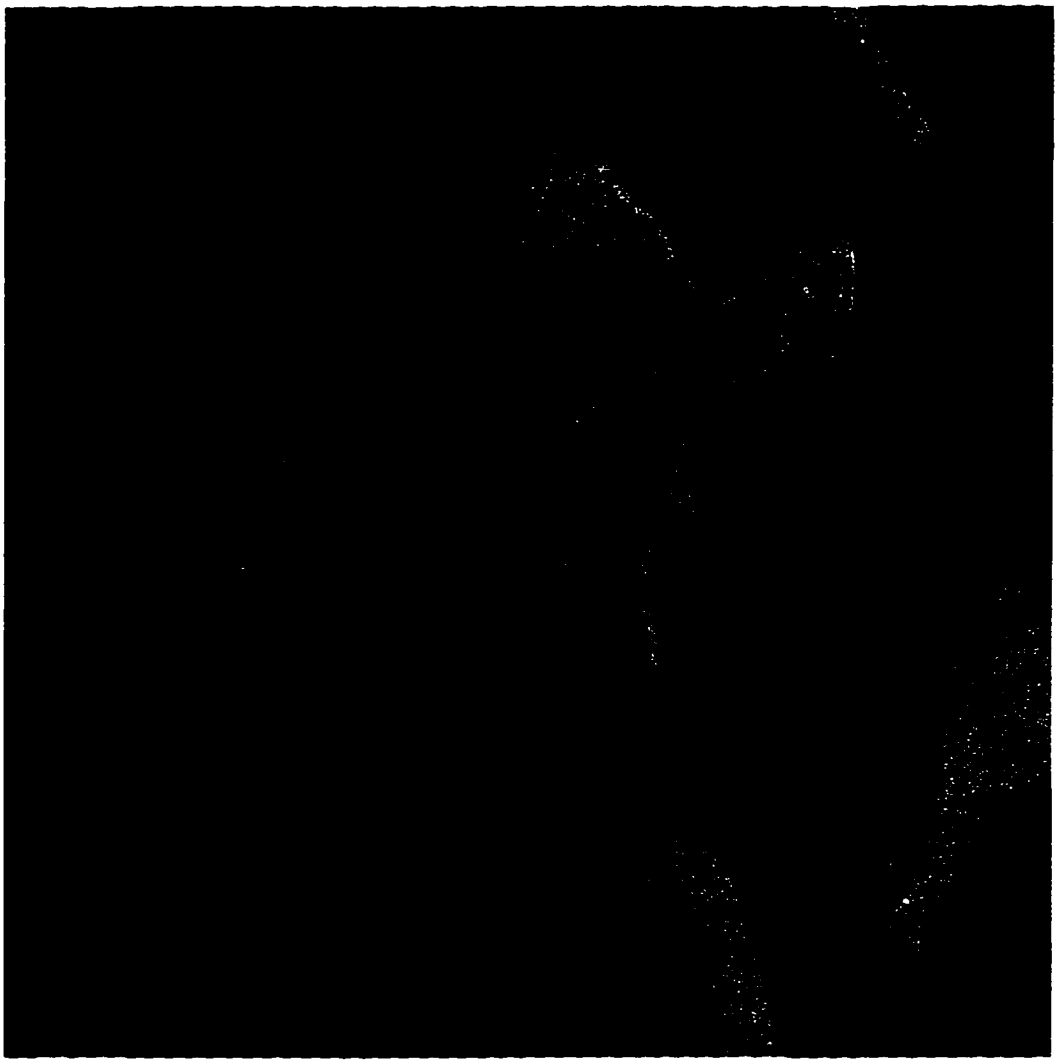

Figure 6.6: OTIC coding of "lena". Image decoded after 60 iterations, initialized at the "peppers" image. Bock size 8 $\times$ 8. PSNR= 28.3dB, 0.85 bpp.

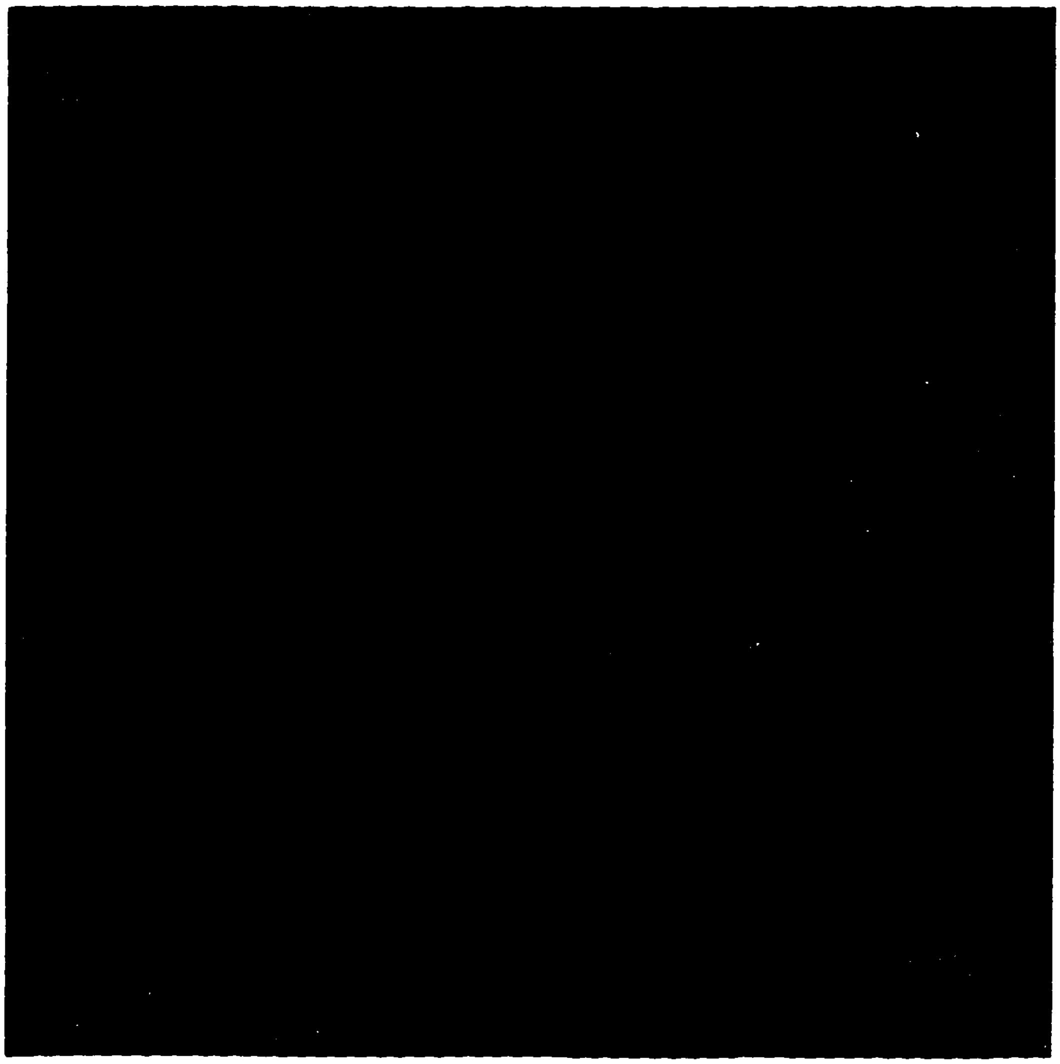

Figure 6.7: Fractal coding of "peppers". Image decoded after 60 iterations, initialized at the "black" image. Block size 8×8. PSNR= 31.5dB, 0.43 bpp.

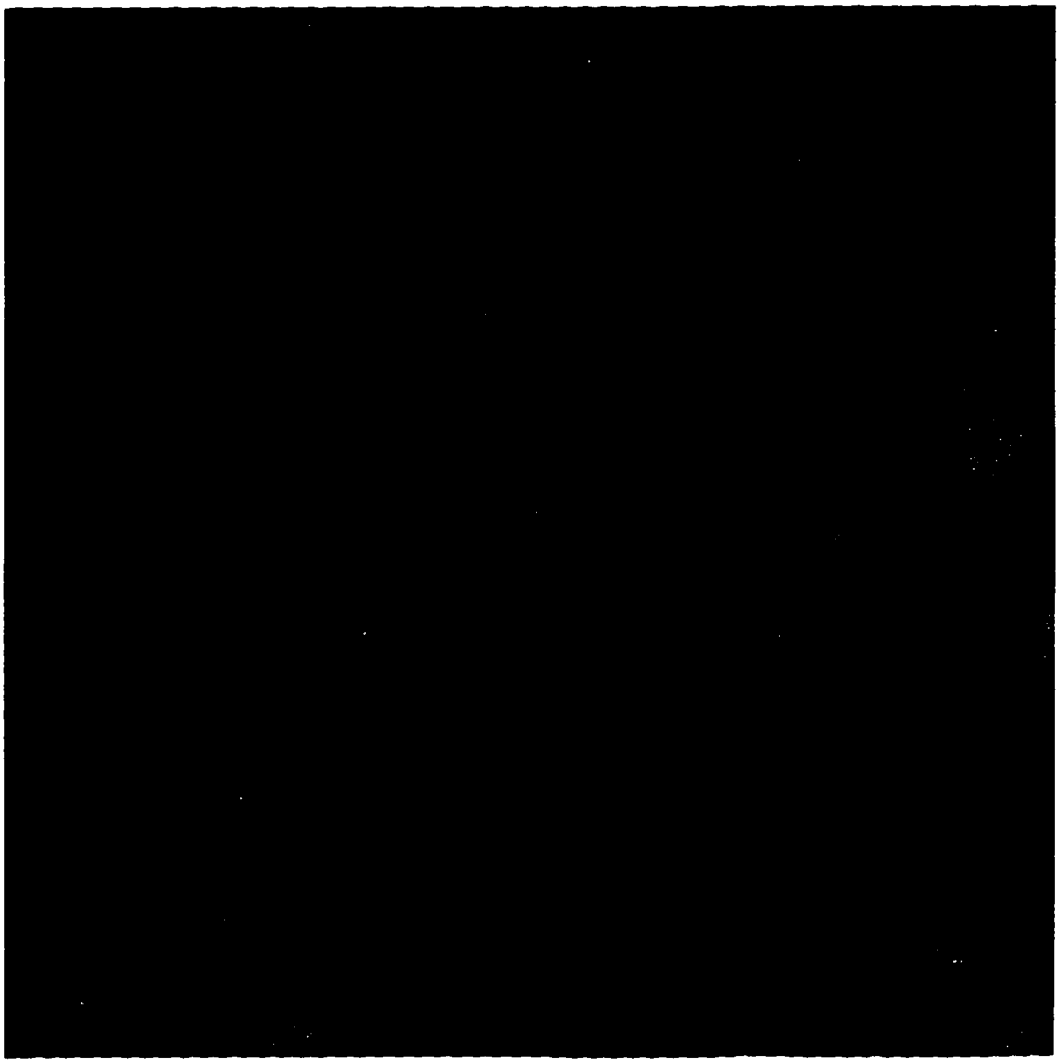

Figure 6.8: OTIC coding of "peppers". Image decoded after 60 iterations, initialized at the "black" image. Block size 8 × 8. PSNR= 33.7dB, 0.66 bpp.

Chapter 7

CONCLUSIONS

7.1 Summary of Contributions

In this dissertation, we have developed – under the name operator theoretic image coding (OTIC) – a new, generalized framework for iterative image coding. Its mathematical foundations has been built on fixed point theory and nonlinear functional analysis. The novelty of the proposed approach resides in that the features of an image are individually encoded into nonexpansive operators. OTIC contains as particular cases most iterative coding schemes, in particular fractal coding and set theoretic coding. In fractal coding the image is encoded into a single contraction and decoded as its fixed point. In set theoretic coding, the image is encoded into a family of projectors onto convex sets and decoded as a common point of these sets. In OTIC, the image is encoded into a family of arbitrary nonexpansive operators and decoded as one of their common fixed points. As a by-product of this work, we thus obtain a general and consistent definition of a set theoretic codec.

The great versatility of OTIC lies in the fact that there is no theoretical limitation on the number of operators that can be involved in the description of an image. Hence, a wide variety of features can be easily encoded and transmitted to the decoder. The decoder processes the operators so as to find one of their common fixed points. The numerical viability of OTIC lies in the availability of powerful parallel methods for carrying out the decoding process. Such methods have been developed in the context of this work. The flexibility and quality of OTIC has been illustrated in numerical simulations.

Although the implementations of OTIC shown here have been demonstrated only for still images, its extension to coding of digital image sequence is also possible. Study related to operator theoretic video coding will be an important part of our future work.

It is also necessary to mention that although OTIC provides a very general and flexible platform for iterative signal coding, substantial improvements in coding quality and compression ratios heavily rely, as in any coding technique, on the implementation. In the simulations presented in this dissertation, the implementation is quite rudimentary and the results relatively modest. It can be expected that more sophisticated implementations will bring much better coding results.

7.2 Directions for Future Research

There are several points which may be investigated in order to extend and improve the general OTIC framework presented in this dissertation.

- One asset of the proposed OTIC scheme is to rely on fairly general operators, namely nonexpansive operators. The flexibility of the method could be further extended by considering more general operators, for instance quasi-nonexpansive mappings or asymptotically nonexpansive mappings. The convergence of algorithms based on such operators has been the focus of little investigations in the literature and is worth pursuing.

- The OTIC method is very flexible and the convergence of the decoder could potentially greatly be accelerated by studying the influence of the weights and of the control schemes.

- The automatic selection of optimal codes should be investigated. Ultimately, the goal is to select a small number of operators that will yield a small feasibility set S in (1.5). This, in turn, will minimize the reconstruction error and the bit rate.

- Meaningful quality measures should be developed to compare the performance of OTIC to existing coding schemes based on criteria such as encoding speed, bit-rate, decoding speed, reconstruction quality and memory requirements.

New York, February 1996.

Bibliography

[1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers,* vol. 23, no. 1, pp. 90-93, January 1974.

[2] K. Aizawa and T. S. Huang, "Model-based image coding: Advanced video coding techniques for very low bit-rate application," *Proceedings of the IEEE,* vol. 83, no. 2, pp. 259-271, February 1995.

[3] M. Barlaud, P. A. Chou, N. M. Nasrabadi, D. Neuhoff, M. J. T. Smith, and J. W. Woods, "Guest editorial: Introduction to the special issue on vector quantization," *IEEE Transactions on Image Processing,* vol. 5, no. 2, pp. 197-201, February 1996.

[4] M. F. Barnsley, *Fractals Everywhere.* San Diego: Academic Press, 1988.

[5] M. F. Barnsley and L. P. Hurd, *Fractal Image Compression.* Wellsley, MA: AK Peters, 1993.

[6] T. C. Bell, J. G. Cleary, and J. H. Witten, *Text Compression.* Englewood Cliff, NJ: Prentice-Hall, 1990.

[7] L. M. Brègman, "The method of successive projection for finding a common point of convex sets," *Soviet Mathematics - Doklady,* vol. 6, no. 3, pp. 688-692, May 1965.

[8] Y. Censor, "Iterative methods for the convex feasibility problem," *Annals of Discrete Mathematics,* vol. 20, pp. 83-91, 1984.

[9] Y. Censor and A. Lent, "Cyclic subgradient projections," *Mathematical Programming,* vol. 24, no. 2, pp. 233-235, 1982.

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

[11] P. L. Combettes, "The foundations of set theoretic estimation," *Proceedings of the IEEE,* vol. 81, no. 2, pp. 182-208, February 1993.

[12] P. L. Combettes, "Construction d'un point fixe commun à une famille de contractions fermes," *Comptes Rendus de l'Académie des Sciences de Paris,* Série I, vol. 320, no. 11, pp. 1385-1390, June 1995.

[13] P. L. Combettes, *The Convex Feasibility Problem in Image Recovery,* in *Advances in Imaging and Electron Physics* (P. Hawkes, Ed.), vol. 95, pp. 155-270. New York: Academic Press, 1996.

[14] P. L. Combettes and H. Puh, "A fast parallel projection algorithm for set theoretic image recovery," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing,* vol. 5, pp. 473-476. Adelaide, Australia, April 19-22, 1994.

[15] P. L. Combettes and H. Puh, "Iterations of parallel convex projections in Hilbert spaces," *Numerical Functional Analysis and Optimization,* vol. 15, no. 3&4, pp. 225-243, 1994.

[16] A. R. De Pierro nd A. N. Iusem, "A parallel projection method for finding a common point of a family of convex sets," *Pesquisa Operacional,* vol. 5, no. 1, pp. 1-21, 1985.

[17] I. Ekeland and R. Temam, *Analyse Convexe et Problèmes Variationnels.* Paris: Dunod, 1974. English translation: *Convex Analysis and Variational Problems,* Amsterdam: North-Holland Publication, 1976.

[18] Y. Fisher (Ed.), *Fractal Image Compression – Theory and Application.* New York: Springer-Verlag, 1995.

[19] S. D. Flåm, and J. Zowe, "Relaxed outer projections, weighted averages, and convex feasibility," *BIT,* vol. 30, no. 2, pp. 289-300, 1990.

[20] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Boston MA: Kluwer Academic Publishers, 1992.

[21] K. Goebel and W. A. Kirk, *Topics in Metric Fixed Point Theory.* Cambridge: Cambridge University Press, 1990.

[22] C. W. Groetsch, "A note on segmenting Mann iterates," *Journal of Mathematical Analysis and Applications,* vol. 40, no. 2, pp. 369-372, November 1972.

[23] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Transactions on Image Processing,* vol. 1, no. 1, pp 18-31, January 1992.

[24] A. E. Jacquin, "Fractal image coding: A review," *Proceedings of the IEEE,* vol. 81, no. 10, pp. 1451-1465, October 1993.

[25] A. K. Jain, "Image data compression: A review," *Proceedings of the IEEE,* vol. 69, no. 3, pp. 349-389, March 1981.

[26] A. K. Jain, *Fundamentals of Digital Image Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[27] N. Jayant, "Signal compression: Technology targets and research directions," *IEEE Journal on Selected Areas in Communications,* vol. 10, no. 5, pp. 796-817, June 1992.

[28] M. C. Joshi and R. K. Bose, *Some Topics in Nonlinear Functional Analysis.* New York: Halsted Press/John Wiley, 1985.

[29] I. Katsavounidis, C.-C. J. Kuo, and Z. Zhang, "A new initialization technique for generalized Lloyd iteration," *IEEE Signal Processing Letters,* vol. 1, no. 10, pp. 144-146, October 1994.

[30] K. C. Kiwiel, "Block-iterative surrogate projection methods for convex feasibility problems," *Linear Algebra and its Applications*, vol. 215, pp. 225-259, January 1995.

[31] F. Kossentini, M. J. T. Smith, and C. F. Barnes, "Image coding using entropy-constrained residual vector quantization," *IEEE Transactions on Image Processing*, vol. 4, no. 10, pp. 1349-1357, October 1995.

[32] D. LeGall, "MPEG: A video compression standard for multimedia applications," *Communications of the ACM*, vol. 34, no. 4, pp. 46-58, April 1991.

[33] H. Li, A. Lundmark, and R. Forchheimer, "Image sequence coding at very low bitrates: A review," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 589-609, September 1994.

[34] D. W. Lin, "Fractal image coding as generalized predictive coding," *Proceedings of the IEEE International Conference on Image Processing*, vol 3, pp. 117-121, Austin, Texas, November 13-16, 1994.

[35] P.-L. Lions, "Approximation de points fixes de contractions," *Comptes Rendus de l'Académie des Sciences de Paris*, vol. A284, no. 21, pp. 1357-1359, June 1977.

[36] S. P. Lloyd, "Least-square quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129-137, March 1982.

[37] G. F. McLean, "Vector quantization for texture classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 637-649, May/June 1993.

[38] A. Narayan and J. F. Doherty, "Convex projections based edge recovery in low bit rate VQ," *IEEE Signal Processing Letters*, vol. 3, no. 4, pp. 97-99, April 1996.

[39] N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: A review," *IEEE Transactions on Communications*, vol. 36, no. 8, pp. 957-971, August 1988.

[40] K. L. Oehler and R. M. Gray, "Combining image compression and classification using vector quantization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 461-473, May 1995.

[41] Z. Opial, "Weak convergence of the sequence of successive approximations for nonexpansive mappings," *Bulletin of the American Mathematical Society*, vol. 73, no. 4, pp. 591-597, July 1967.

[42] G. Pierra, "Decomposition through formalization in a product space," *Mathematical Programming*, vol. 28, no. 1, pp. 96-115, January 1984.

[43] R. P. Plant, "Common fixed points of noncommuting mappings," *Journal of Mathematical Analysis and Applications*, vol. 188, no. 2, pp. 436-440, December 1994.

[44] W. K. Pratt, *Digital Image Processing*, 2nd Edition. New York: Wiley Interscience, 1991.

[45] H. Puh and P. L. Combettes, "Set theoretic vector quantization," *Proceedings of the Ninth IEEE Workshop on Image and Multidimensional Signal Processing,* pp. 48-49. Belize City, Belize, March 3-6, 1996.

[46] B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Transactions on Communication,* vol. 34, no. 11, pp. 1105-1115, November 1986.

[47] R. T. Rockafellar, *Convex Analysis.* Princeton, NJ: Princeton University Press, 1970.

[48] A. Rosenfeld (Ed.), *Image Modeling.* New York: Academic Press, 1981.

[49] R. Rosenholtz and A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 2, no. 1, pp. 91-95, March 1992.

[50] P. Santago and S. A. Rajala, "Using convex set techniques for combined pixel and frequency domain coding of time-varying images," *IEEE Journal on Selected Areas in Communications,* vol. 5, no. 7, pp. 1127-1139, August 1987.

[51] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *IRE National Convention Record,* Part 4, pp. 142-163, 1959.

[52] D. R. Smart, *Fixed Point Theorems.* London: Cambridge University Press, 1974.

[53] H. Stark (Ed.), *Image Recovery: Theory and Application.* Orlando: Academic Press, 1987.

[54] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projections onto convex sets," *IEEE Transactions on Image Processing,* vol. 4, no. 4, pp. 470-477, April 1995.

[55] A. M. Tekalp, *Digital Video Processing.* Upper Saddle River, NJ: Prentice Hall, 1995.

[56] N. T. Thao, K. Asai, and M. Vetterli, "Set theoretic compression with an application to image coding," *Proceedings of the IEEE International Conference on Image Processing,* vol. 2, pp. 336-339. Austin, TX, November 13-16, 1994.

[57] P. Tseng, "On the convergence of products of firmly nonexpansive mappings," *SIAM Journal on Optimization,* vol. 2, no. 3, pp. 425-434, August 1992.

[58] H. J. Trussell and M. R. Civanlar, "The feasible solution in signal restoration," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 32, no. 2, pp. 201-212, April 1984.

[59] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding.* Englewood Cliffs, NJ: Prentice Hall, 1995.

[60] G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM,* vol. 34, no. 4, pp. 30-44, April 1991.

[61] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, "Projection-based spatially adaptive reconstruction of block-transform compressed images," *IEEE Transactions on Image Processing,* vol. 4, no. 7, pp. 896-908, July 1995.

[62] D. C. Youla and H. Webb, "Image restoration by the method of convex projections: Part 1 – Theory," *IEEE Transactions of Medical Imaging,* vol. 1, no. 2, pp. 81-94, October 1982.

[63] E. Zeidler, *Nonlinear Functional Analysis and its Application I.* New York: Springer-Verlag, 1986.