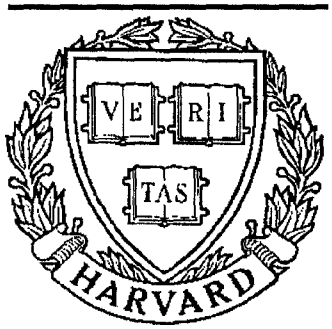
THESIS REPORT
Ph.D.

SYSTEMS
RESEARCH
CENTER



Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), the University of Maryland, Harvard University, and Industry

A Three-Component Image Model Based on Human Visual Perception and Its Applications in Image Coding and Processing

by X. Ran
Advisor: N. Farvardin

Ph.D. 92-15

ABSTRACT

Title of Dissertation: A THREE-COMPONENT IMAGE MODEL BASED ON HUMAN VISUAL PERCEPTION AND ITS APPLICATIONS IN IMAGE CODING AND PROCESSING

Xiaonong Ran, Doctor of Philosophy, 1992

Dissertation directed by: Associate Professor Nariman Farvardin
Department of Electrical Engineering

In this work, results of psychovisual studies of the human visual system are discussed and interpreted in a mathematical framework. The formation of the perception is described by appropriate minimization problems and the edge information is found to be of primary importance in the visual perception. Having introduced the concept of edge strength, it is demonstrated that strong edges are of higher perceptual importance than weaker edges (textures). We have also found that smooth areas of an image influence the human visual perception together with the edge information, and that this influence can be mathematically described via a minimization problem. Based on this study, we have proposed to decompose the image into three components: (i) primary, (ii) smooth and (iii) texture, which contain, respectively, the strong edges, the background and

the textures. An algorithm is developed to generate the three-component image model.

Then, the use of this perceptually-motivated image model in the context of image compression is investigated. The primary component is encoded separately by encoding the intensity and geometric information of the strong edge brim contours. Two alternatives for coding the smooth and texture components are studied: Entropy-coded adaptive DCT and entropy-coded subband coding. It is shown via extensive simulations that the proposed schemes, which can be thought of as a hybrid of waveform coding and feature-based coding techniques, result in both subjective and objective performance improvements over several other image coding schemes and, in particular, over the JPEG continuous-tone image compression standard. These improvements are especially noticeable at low bit rates. Furthermore, it is shown that a perceptual tuning based on the contrast-sensitivity of the human visual system can be used in the DCT-based scheme, which in conjunction with the three-component model, leads to additional subjective performance improvements.

Finally, a scheme for structurally representing planar curves is developed based on the ideas for the three-component image model. This scheme does not have the ambiguity problem associated with the scale-space-based schemes.

A THREE-COMPONENT IMAGE MODEL BASED ON HUMAN VISUAL PERCEPTION AND ITS APPLICATIONS IN IMAGE CODING AND PROCESSING

by

Xiaonong Ran

Dissertation submitted to the Faculty of the Graduate School
of The University of Maryland in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
1992

Advisory Committee:

Associate Professor Nariman Farvardin, Chairman/Advisor
Professor Lee D. Davisson
Assistant Professor Thomas E. Fuja
Assistant Professor K. J. Ray Liu
Professor Seymour Goldberg

Dedication

To Minmin and Mengchen

Acknowledgements

I am very grateful to my advisor, Nariman Farvardin, for his teaching and ideas. I am also indebted to the professors who have taught me in the graduate courses at UMCP and who are Imre Csiszár, Thomas E. Fuja, Seymour Goldberg, Armand M. Makowski, Robert W. Newcomb, Frank W. Olver, Adrian Papamarcou, Ryszard Syski, Andre L. Tits and Peter Walters.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Approach

The growing interest and need to store and/or transmit digital imagery and video and the practical limitations on the storage and transmission capacity, have led to a significant amount of research activity in image compression (also called image coding) over the past two decades. In any image compression system the goal is that of producing a good replica of the original image with as small a number of bits as possible.

In this dissertation, we limit ourselves to the compression of continuous-tone still images.[1] The major thrust in still image compression can be divided into the following two categories: (i) *waveform coding* techniques and (ii) *feature-based* or *second-generation coding* techniques.

Waveform coding techniques revolve around information-theoretic principles

[1]Continuous-tone still image compression can be applied directly to the compression of components of color images, and to the compression of I-frames (Intrapictures) of motion video signals [1] - [3].

in which upon selecting a distortion criterion (in most cases squared-error) and certain probabilistic assumptions on the image data, the goal becomes that of minimizing the average distortion between the original image and its reconstruction under an average bit rate constraint. In these techniques, the human visual system (HVS) and its sensitivity to the coding error generally do not play a central role in system design. Waveform coding techniques, which by and large constitute the bulk of the research in image coding, have led to a plethora of different coding techniques. Among these, the most noteworthy are: (i) various adaptive forms of two-dimensional (2-D) discrete cosine transform (DCT) coding techniques, (ii) 2-D subband and wavelet image coding techniques and (iii) various forms of vector quantization of images. Several versions of transform-based methods have been proposed in the past two decades, [4] - [7]. The work in DCT coding has recently led to an international still image compression standard known by the acronym JPEG (Joint Photographic Experts Group) [2]. Subband image coding was first introduced by Woods and O'Neil [8] in 1986 and since then has been the subject of much research [9] - [12]. More recently, similar multi-resolution methods using wavelet transforms have received some attention in the context of image coding [13]. Since the pioneering work of Linde, Buzo and Gray [14], vector quantization has been widely studied for image coding both in the spatial domain and in the frequency domain in conjunction with transform or subband coding. A survey of vector quantization methods in image coding can be found in [15]. Additional details on these waveform coding techniques and various combinations thereof can be found in [16], [17].

Generally speaking, waveform coding techniques provide good-to-excellent quality results at compression ratios of 20:1 - 10:1 (bit rates of 0.4 - 0.8 bits/pixel,

assuming 8 bits/pixel for the original). With such compression ratios, the average distortion is typically small and hence the corresponding reconstructed image is of high perceptual quality. At lower bit rates however, waveform coding produces specific types of artifacts such as blockiness, ringing and blurring around the edges, etc. In contrast with waveform coding, feature-based image coding methods are closely tied to the HVS and its separate sensitivities to the strong edge and texture information. Instead of considering the image as a waveform, these methods attempt to describe the image by a collection of physically significant entities such as regions or contours; this leads to a more compact representation of the image and hence significantly higher compression ratios, albeit at the cost of a different type of distortion [18], [19].

Instead of abandoning the concepts of either one of the above two categories of image compression, in this dissertation, the ideas of the two categories are combined. The objective is to achieve high compression ratios (higher than $20:1$) with better objective and subjective performances than that of waveform coding techniques and without introducing the distortions of feature-based image coding methods.

The approach used in this work is to determine what special properties of the image signal are responsible for the formation of the perception in the HVS, to mathematically characterize these properties, to decompose the image signal in accordance with these properties and to develop separate coding schemes, based on information-theoretic principles, for the resultant image components to achieve performances as close to the optimums (in the rate-distortion sense) as possible. This approach can be summerized in Fig. 1.1.

With this approach, the properties of the HVS are incorporated into the

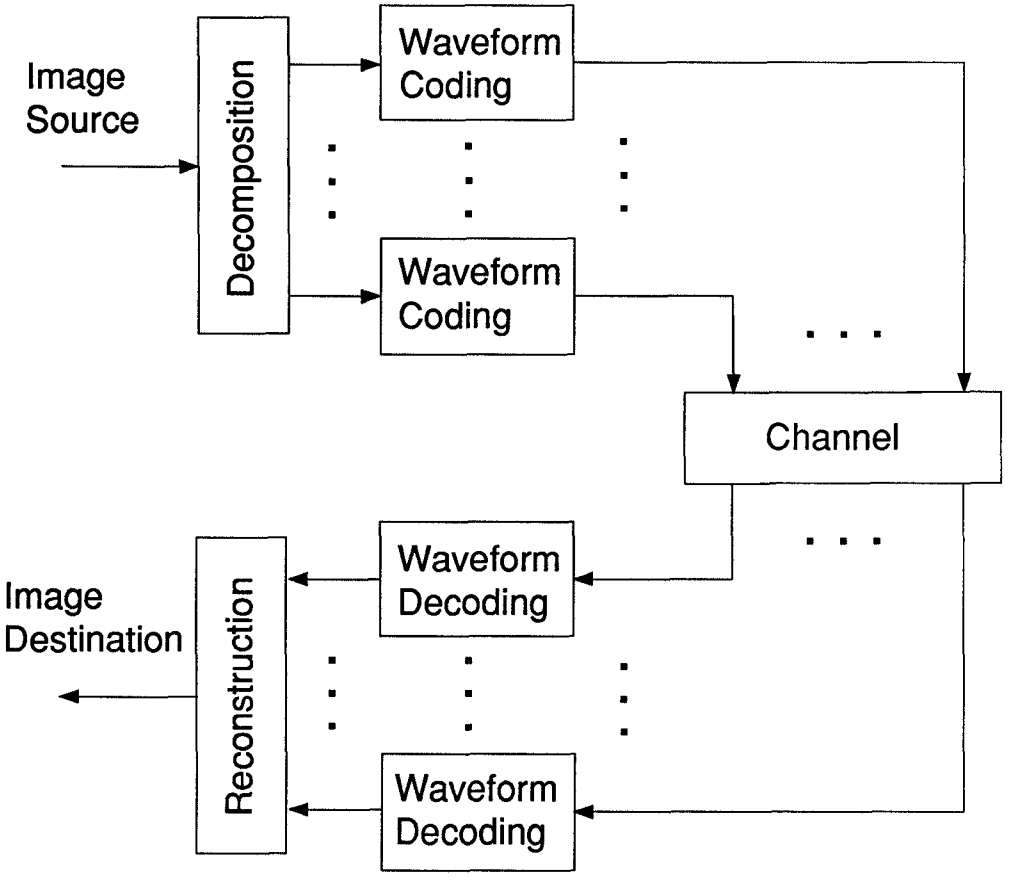


Figure 1.1: Block diagram of the image communication system based on the approach of this work.

design of the image coding system through the special decomposition of the image signal without introducing any distortion, while the quantization and coding of each components are performed based on information-theoretic principles to achieve the minimum coding rates.

1.2 Outline

This dissertation is organized into five chapters. Chapter 1 contains the motivation and approach for the work, this outline and a section on the main contributions of the work.

In Chapter 2, the special properties of the image signal which are significant to the perception of the HVS are investigated and mathematically interpreted. Based on these properties and their mathematical interpretation, a decomposition scheme of the image signal is developed. Experimental results and comparisons are provided.

In Chapter 3, the coding schemes for the image components are developed. Several coding systems are then constructed and compared with the international standard JPEG and with other state-of-the-art image coding systems.

In Chapter 4, a planar curve representation scheme is developed. The idea of this scheme is motivated by the concepts developed in Chapter 2. Simulations and a comparison are provided.

Chapter 5 contains a summary of this dissertation and a discussion on the possible directions for the continuation of this work.

1.3 Contributions

The major contribution of this work is the development of a component-based image coding scheme where the decomposition is based on the perceptual properties of the HVS, and the coding of the image components are performed based on information-theoretic principles. Various implementations of this scheme offer significant improvements both in subjective and objective performances over

other state-of-the-art image coding systems and, in particular, over the JPEG continuous-tone still image compression standard.

More specifically, the binocular nature of the HVS is used in the discovery of a constrained variation energy minimization problem which provides an insight to the mechanism for the formation of the visual perception. The effects of strong edges and their interactions with textures or smooth areas on the visual perception are described in terms of constrained minimization problems. The constraining condition, or the strong edge information, of these constrained minimization problems are characterized; the characterization leads to the development of a scheme for the extraction of the strong edge information. The extraction is achieved in two stages: In the first stage, the strong edge information and the information of the background slow-intensity variations are extracted; in the second stage, the strong edge information is extracted from the result of the first stage. Having expressed the strong edge information in an image form by solving a constrained energy minimization problem, this extraction scheme gives rise to a three-way decomposition of the image signal, namely, the primary, smooth, and texture components. The edge extraction part of the three-component image model provides superior representations of the strong edge information as compared with the conventional Laplacian-Gaussian operator scheme.

To encode the primary component, special quantization rules are used for the intensity values and the geometrical locations of the edge brim contours. Various criteria on the intensity values for the formation of the contours are considered in order to obtain the most efficient coding results. Predictive encoding followed by arithmetic coding is used to code the geometric information of the contours.

The smooth and the texture components are encoded using two types of

image coding techniques, adaptive 2D DCT coding and 2D subband coding. The block classification scheme developed uses the frequency properties of the smooth and texture components and results in PSNR improvements. A study is conducted to determine the approximate distributions of the 2D DCT coefficients of the different classes among the generalized Gaussian distributions. Bit allocation is made efficiently based on the rate-distortion performances of the uniform-threshold quantizer and Huffman code pairs. An estimation scheme for the variances of the transform coefficients is developed using the Shannon lower bounds of rate-distortion functions. This variance estimation scheme gives rise to significant savings in the resulting encoding rates. Arithmetic coding is also incorporated in the systems and results in reduced encoding rates.

Various image coding systems are designed based on the developed coding schemes for the image components. Extensive simulations are conducted for these systems on different real-world images. The adaptive 2D DCT coding with the three-component image model offers the best performances both objectively and subjectively; the contributions of the constituent functional factors in this system to the overall performance are investigated through different combinations of these factors. The analytical PSNR performances of the coding systems are also studied.

A frequency-weighted distortion criterion is developed; this criterion uses the concept of contrast sensitivity of the HVS. In conjunction with the three-component model, this perceptual tuning provides further subjective performance improvements.

The concepts in the three-component image model are also useful for the problems in the areas of image processing and computer vision. One of such

problems is that of representing planar curves, which is important since for many applications in image processing and computer vision the information to be processed and/or extracted is expressed in the form of curves. An efficient way to represent a planar curve is to locate the so-called feature points along the curve and to use these feature points as the bases for a structural representation of the curve. Most of the existing schemes for locating the feature points transform the problem into the orientation domain and apply the Laplacian-Gaussian operator to locate the significant changes in orientation along the curve. The performance of Laplacian-Gaussian operator is limited due to the difficulty associated with the uncertainty principle of the operator. To obtain better performance, people have used scale-space approach in which multiple Laplacian-Gaussian operators are applied. However, the scale-space approach leads to an ambiguity problem in the process of coarse-to-fine tracking across the scale-space. This ambiguity is annihilated to an extent by a primitive-based scheme at the cost of some additional complexity. However, since the coarse-to-fine tracking in this primitive-based scheme is guided by the behavior of several idealized instances of the primitives of curve segments, the types of the feature points to be detected are limited a priori. In this dissertation, a new scheme for locating curve feature points is developed; this scheme is not based on Laplacian-Gaussian operator, thus does not have the difficulties related to the uncertainty principle of the operator. The new scheme is basically a complex-valued 1-D variant of the strong edge extracting scheme for the three-component model. Experimental results show that the new scheme offers efficient extraction of the curve feature points at a reduced complexity as compared to the primitive-based scheme.

Chapter 2

A Perceptually Motivated Three-Component Image Model

2.1 Introduction

In this chapter, we develop a model for real-world imagery based on those features of the image signal that are of distinct significance to human perception. Specifically, we formalize some previous psychovisual studies [20] [21] to characterize the properties of the image signal that are responsible for the formation of the perception in the human visual system. In particular, some interesting evidences on the binocular nature of the human vision in [20] are used to formulate the formation of the perception as a minimization of the intensity variation energy; this has led to the notion of "strong edges" which apparently plays a significant role in the perception. Additionally, we use the psychovisual observations in [21] to mathematically formulate the interaction between the strong edges and the areas of smooth intensity variations. To characterize the strong

edges, we introduce the concept of the "stressed image" and define the strong edges as the high curvature energy pixels of the stressed image. This stressed image, which has an interesting analogy to the stable configuration of a mechanical structure, is generated by a space-variant low-pass filtering of the original image. The above formalism has led to a three-component image model consisting of (i) the strong edge component, (ii) the smooth intensity variation component and (iii) the texture component.

The three-component model developed here is quite general and might prove useful in various image processing algorithms (ref. Chapter 4). However, it was primarily developed for an image coding situation where the different features of the image signal can be classified (according to the role they play in the formation of human perception), extracted and treated separately for subsequent encoding. This idea is similar to the sketch-based image coding scheme of [22] [23] which consists of two components: one is similar to the strong edge component of the three-component model developed here, and the other is simply the residual between the original image and an image obtained merely from the strong edge information. In the sketch-based scheme, the strong edges are extracted by the Laplacian-Gaussian Operator (LGO) followed by a gradient operator. This edge extraction scheme suffers from certain drawbacks which will be discussed in detail. In addition, a careful comparison between the LGO-based method and the edge extraction scheme associated with the three-component model will be presented. The applications of the three-component image model in defining a perceptual distortion criterion for specific image coding schemes are presented in the next chapter.

This chapter is organized as follows. In Section 2.2, certain psychovisual

evidences of the human perception are presented and discussed. The evidences lead to the characterization of the strong edge information in Section 2.3, followed by an algorithm for the extraction of strong edges along with the development of the three-component image model in Section 2.4. In Section 2.5, an example of the three-component image model is provided and certain comparisons against the LGO-based edge extraction scheme are presented. Summary and conclusions are provided in Section 2.6.

2.2 Observations on Psychovisual Aspects of Human Visual System

In the next three subsections, we describe some observations on psychovisual aspects of the HVS. The objective is to extract and discriminate different properties of image signals which are of significance to the human visual perception. Interpretations in the context of image coding are also provided, and will form the motivation and basis for the work described later. In the first two subsections, the binocular nature of the HVS is used to explain the relationship between the image signals and the human visual perception.

2.2.1 Edge Information of Image Signals

As described in [20], in natural binocular vision, when two views are presented with two forms which are different (in the sense that they do not admit of being combined into the image of a single object), the images of both forms will generally be seen at the same time superposed on one another in the field of view. Usually, in some locations of the field of view, one image dominates the

other, and vice versa in other parts of the field. For the situation where broad black-and-white figures are displayed to both views, the general rule is that the dominating image along an edge and in its vicinity will be the one which owns the edge. As an example, let us consider Fig. 2.1, in which we have two bars, one vertical and one horizontal. When the vertical bar is seen by the left eye and the horizontal bar by the right eye, without devoting exclusive attention to any of the two, the total effect will be an image similar to Fig. 2.2. As shown in Fig. 2.2, the perception will be a cross which is black over the center square; the background appears white. The four arms of the cross are perfectly black at their ends and almost entirely white near the center square, with transitions in between.

Based on this phenomenon, we may conclude that the collection of the individual pixel intensity values without any interaction between them (the most primitive property of the image) is not what produces the visual perception. For otherwise, the image in the field of view should be formed at each pixel with an intensity produced only by the two corresponding pixel intensities of the two images according to a certain law, and thus the perception should be a uniform combination of the two pictures. Obviously, the image shown in Fig. 2.2 cannot be constructed by a *uniform* combination of the two images in Fig. 2.1.

Based on this observation, we conjecture that here it is the edge property (or information), to be defined next, of the image signal that arouses our visual perception. For the images in Fig. 2.1, the edge information is described by (i) the *locations* of variations of intensity values and (ii) the related *intensity values* at these locations. For example, in Fig. 2.1 (a), the locations of the intensity variations are along the border of the vertical black bar, and the related

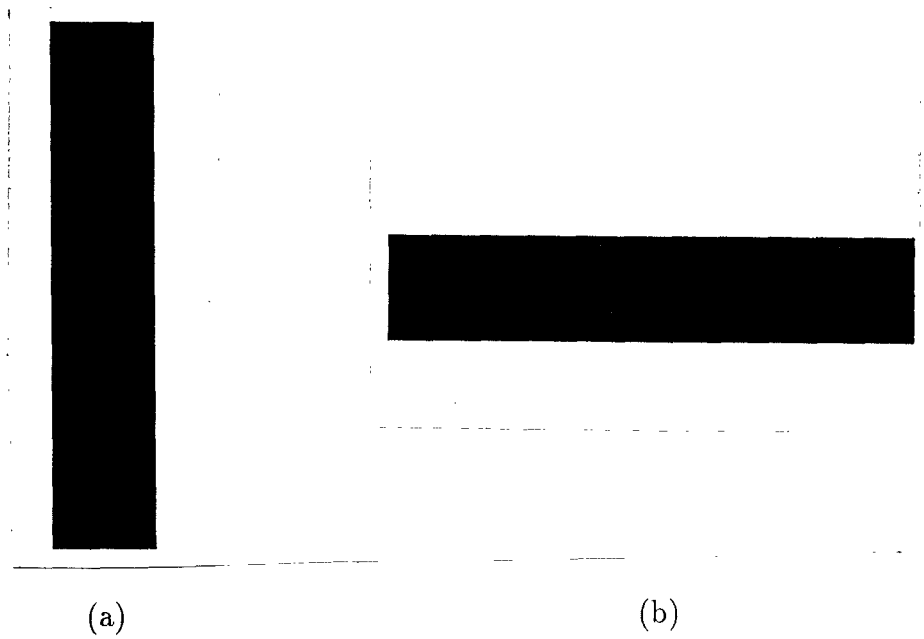


Figure 2.1: (a) Vertical bar for the left eye; (b) horizontal bar for the right eye.

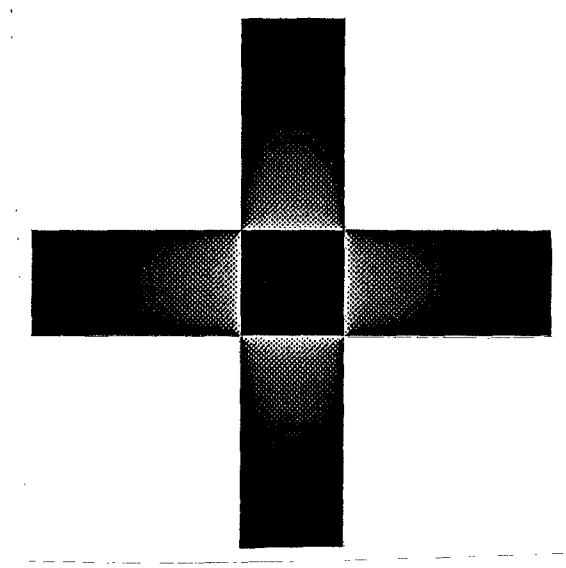


Figure 2.2: Binocular perception of the two images in Fig. 2.1.

intensity values are those intensity values immediately inside and outside the border. The above actually describes what we generally refer to as an *edge*. This edge basically has two related intensity values, zero (255) corresponding to black (white), representing the intensity variation. We call the contour formed by the lower intensity value the *lower brim* of the edge (or lower edge brim), and the contour formed by the higher intensity value the *upper brim* of the edge (or upper edge brim). For example, in Fig. 2.1 (a), the rectangular contour of the lower edge brim is just inside the border, while the one for the upper edge brim is just outside. The locations and intensity values of the lower and upper edge brims completely define the corresponding edge, and, therefore, we may define an edge through the concept of edge brims which will be characterized mathematically later in this chapter.

We now go back to our conjecture that the edge information is responsible for the formation of the visual perception. Remarkably, this conjecture is justified in this case by actually producing Fig. 2.2 *only* from the knowledge of lower and upper edge brims of the two images in Fig. 2.1 by minimizing the variation in intensity values as described below.

Let a generic digital image of size $M \times M$ be denoted by an array of real numbers $\{x_{i,j}\}$, $i,j = 0,1,\ldots,M-1$, where $x_{i,j}$ is the intensity value of the pixel (i,j) at the ith row and the jth column. Note that an image $\{x_{i,j}\}$, $i,j = 0,1,\ldots,M-1$, can be defined as a set $\mathcal{X}$ of triples: $\mathcal{X} \equiv \{(i,j,x_{i,j}),$ $i,j = 0,1,\ldots,M-1\} \subset \mathbb{R}^3$. Then the lower and upper edge brims of an image $\mathcal{X}$ are a subset of $\mathcal{X}$. Let the subsets of lower and upper edge brims of the images (a) and (b) in Fig. 2.1 be denoted by $\mathcal{B}_a$ and $\mathcal{B}_b$, respectively. We combine the information in $\mathcal{B}_a$ and $\mathcal{B}_b$ by forming a set $\mathcal{B}$ from $\mathcal{B}_a \cup \mathcal{B}_b$ in the

following way. We first define two projection functions, $f_1 : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ where $f_1((i,j,x)) = (i,j)$, and f_2: $\mathbb{R}^3 \rightarrow \mathbb{R}$ where $f_2((i,j,x)) = x$. We denote by $f_k(\mathcal{A})$ the image of a set $\mathcal{A}$ under f_k, $k = 1, 2$. For a set $\mathcal{A} \subset \mathbb{R}^3$, we also define $\mathcal{A}^{(i,j)} = \{s : s \in \mathcal{A} \text{ and } f_1(s) = (i,j)\}$. Then the set $\mathcal{B}$ is defined as,

$$\mathcal{B} = \{(i,j,\overline{x}_{i,j}) : \ (i,j) \in f_1(\mathcal{B}_a \cup \mathcal{B}_b)\}, \tag{2.1}$$

where $\overline{x}_{i,j}$ is the average value of the elements of the set $f_2\left((\mathcal{B}_a \cup \mathcal{B}_b)^{(i,j)}\right)$. In other words, the set $\mathcal{B}$ is a "linear" combination of $\mathcal{B}_a$ and $\mathcal{B}_b$ in the above sense.

Before describing how the image shown in Fig. 2.2, denoted by $\mathcal{X}_c$, can be obtained from the edge information contained in $\mathcal{B}$, we make the following definition. The *variation energy* of intensity values of an image $\mathcal{X}$ is defined as:

$$V_{\mathcal{X}} = \sum_{i=0}^{M-2} \sum_{j=0}^{M-2} [(x_{i,j} - x_{i,j+1})^2 + (x_{i,j} - x_{i+1,j})^2]. \tag{2.2}$$

As indicated above, our objective is to generate an image $\mathcal{X}_c$, as shown in Fig. 2.2, solely from the information contained in $\mathcal{B}$. In other words, we seek to find an image $\mathcal{X}_c$ having minimum information while containing the information in $\mathcal{B}$. We call this concept the *minimum information principle* [1]. We quantify the information content of an image $\mathcal{X}$ by its variation energy $V_{\mathcal{X}}$, and define $\mathcal{X}_c = \{(i,j,x^c_{i,j})\}$ to be the solution of the minimization problem:

$$\min_{\{x_{i,j}\}} V_{\mathcal{X}}, \ \text{ subject to } \mathcal{X} \cap \mathcal{B} = \mathcal{B}, \tag{2.3}$$

where $\mathcal{X} = \{(i,j,x_{i,j})\}$. For example, an image with uniform intensity values has zero variation energy, and thus contains the smallest amount of information by the above definition. Because of the quadratic nature of the objective function

[1]The term "minimum information principle" is motivated by a concept, referred to as "*no news is good news*," introduced in [24].

$V_{\mathcal{X}}$, we may write it in matrix notation: $\mathbf{x}^T L_V \mathbf{x} + \mathbf{x}^T H \mathbf{x}_{\mathcal{B}} + \mathbf{x}_{\mathcal{B}}^T D \mathbf{x}_{\mathcal{B}}$, where $\mathbf{x}$ is a vector in $\mathbb{R}^{M^2-|\mathcal{B}|}$ containing elements $x_{i,j}$, $(i,j) \notin f_1(\mathcal{B})$ in a certain order, $\mathbf{x}_{\mathcal{B}}$ is a vector in $\mathbb{R}^{|\mathcal{B}|}$ with elements $x_{i,j}$, $(i,j) \in f_1(\mathcal{B})$, the matrix L_V is a non-negative definite (positive definite when $\mathcal{B} \neq \emptyset$) matrix, and the superscript ^{T} indicates vector transpose. Then, when $\mathcal{B} \neq \emptyset$, the unique solution of problem (2.3) is given by the vector $\mathbf{x}_c \equiv -L_V^{-1} H \mathbf{x}_{\mathcal{B}}/2$. For the given example, the resulting image $\mathcal{X}_c$ is shown in Fig. 2.2; this image, obtained from our minimization problem, is very similar to the corresponding illustration shown in Fig. 73 of [20].

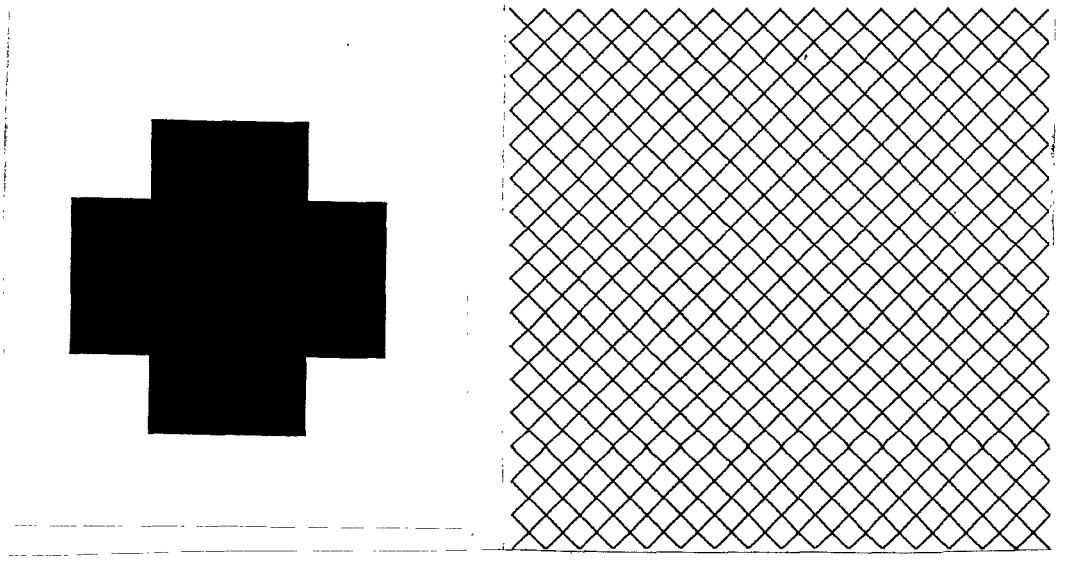
Therefore, in an image information system for which the human visual system is the final receiver, the necessary and sufficient information to be transmitted and/or stored for the images in Fig. 2.1 is the edge information.

To summarize, we explain the phenomenon in Fig. 2.2 with a conjecture that the visual perception is derived from the edge information of the image. Furthermore, the perception can be illustrated by an image which is the solution of a minimization problem similar to (2.3). In this sense, we may take the mechanism governed by this minimization problem as the one which forms our perception.

We now question whether all edges are of the same importance in forming the visual perception. In the next subsection, we describe a binocular phenomenon in an effort to answer this question in the context of image coding.

2.2.2 Strong Edges and Textures

Similar to Fig. 2.1, shown in Fig. 2.3 are two images, (a) and (b), separately presented to the two eyes. In Fig. 2.3, image (a), examined by the left eye, is a black cross, while image (b) for the right eye is a network of slanted black



(a) (b)

Figure 2.3: (a) A cross; (b) a network of lines.

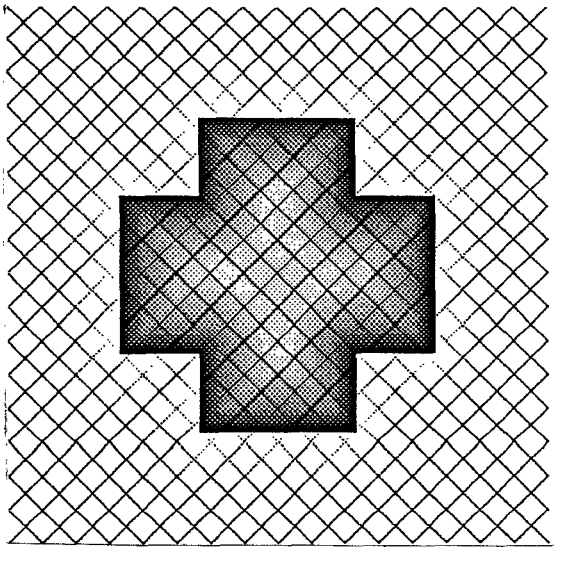


Figure 2.4: Binocular perception of the two images in Fig. 2.3.

lines over a white background. Without any special attention to either image, the usual binocular perception would be similar to the image shown in Fig. 2.4. That is, the image of the cross prevails along its edges; it is only at the center of the cross and away from its edges that the pattern of the network texture may be visible [20].

In view of this phenomenon, the two images in Fig. 2.3 do not have edge information of the same importance, for otherwise in the vicinity of the edge of the cross the network texture should not be invisible. Thus, in this case, we cannot combine the edge information "linearly" as in (2.1). We speculate that an edge has a *strength* property associated with it. Edges of relatively high (low) strength have stronger (weaker) influence on our perception, and may be called *stronger (weaker) edges.* Weaker edges commonly correspond to *textures* in the usual sense, and thus are referred to as textures. As compared with textures, we may simply refer to stronger edges as *strong edges.*

Apparently, edges with larger intensity variations and shorter widths (distance between the corresponding edge brims) are relatively more significant to the visual perception, and are generally called sharp edges. However, edges do not individually influence the visual perception; rather, they interact with other neighboring edges. For example, a thin black line in the network of Fig. 2.3 (b) actually consists of two edges of the same intensity variation and width as for the edge of the cross, but it has less perceptual significance when compared with the edge of the cross. Based on this observation, we conjecture that every edge has an original strength which is proportional to (i) the intensity variation between its two edge brims and (ii) its width. Neighboring edges interact with each other in an inhibitive way. The more closely two edges stand, the more severely

their strengths are weakened. When two edges of equal original strength stand side by side, as in the case of a thin black line on a white background, their strengths vanish. The resulting strength of an edge after it interacts with the neighboring edges will be called the *strength* of the edge. Thus stronger edges are those edges of higher intensity variation and shorter width which are relatively isolated, while weaker edges are those of lower intensity variation and longer width, being relatively crowded with other edges.

We now use this qualitative notion of edge strength to explain the phenomenon of Fig. 2.4. Similar to Subsection 2.2.1, we denote the subsets of lower and upper edge brims of images (a) and (b) in Fig. 2.3 by $\mathcal{B}_a$ and $\mathcal{B}_b$, respectively. We combine the information in $\mathcal{B}_a$ and $\mathcal{B}_b$ based on the concept of edge strength in the following way. When stronger edges meet weaker ones, the stronger ones dominate. This domination of the stronger edge will be extended to its neighborhood and the influence will diminish gradually as a function of the distance from the stronger edge. After a certain distance from the stronger edge, weaker edges will begin to show their effects which will eventually dominate at distances far enough from the stronger edge.

Our experiments indicate that this phenomenon for Figs. 2.3 and 2.4 can be explained as follows. In the close neighborhood of the edges represented by $\mathcal{B}_a$, the image for the perception is formed by solving a problem similar to (2.3) with $\mathcal{B}_a$ as the fixed-point set; we call this image $\mathcal{X}_A$. For distances sufficiently far from the edge in $\mathcal{B}_a$, where the influence of the strong edge diminishes, the image of the perception is formed by solving a problem like (2.3) with $\mathcal{B}_b$ as the fixed-point set; we call this image $\mathcal{X}_B$. In between these two extreme cases, the influence of the strong edge in $\mathcal{B}_a$ decreases, and the perception is formed by linearly combining

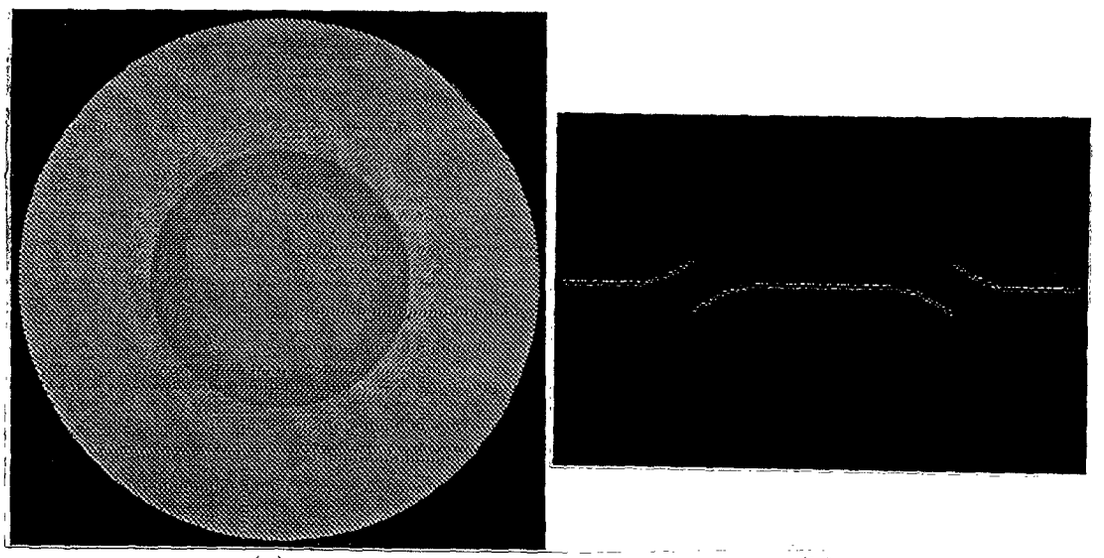
$\mathcal{X}_A$ and $\mathcal{X}_B$ $(\lambda\mathcal{X}_A + (1-\lambda)\mathcal{X}_B)$ where λ decreases quadratically from 1 to 0 as the distance from the strong edge gets large. Our results indicate that this linear combination with the chosen weighting factor results in an image which agrees with our perception as well as the description given in [20, pp. 496-497]. Again as in Subsection 2.2.1, we tested our explanation by actually constructing Fig. 2.4 according to the process described above.

We conclude that stronger edges are of higher importance to the visual perception. Therefore, to design an image information transmission system, one should treat edges in accordance with their strengths to achieve high efficiency. This concept is especially important for image transmission systems at low bit rates.

2.2.3 Smooth Areas of Image Signals

We now investigate the influence of smooth areas of an image on the visual perception. In the smooth areas of an image, intensity values change slowly. For example, in Fig. 2.1 (a), the areas inside and outside of the black bar are smooth areas where intensity values are actually constant. The smooth areas of constant intensity values are of little interest here, since we have indicated in Subsection 2.2.1 that they have no influence on the perception. Instead we will consider the case of smooth areas with non-constant intensity values.

Let us examine the patterns in Fig. 2.5 which are also considered in [21] [24]. In Fig. 2.5 (a) we have two concentric disks with their intensity values along the diameter as illustrated in Fig. 2.5 (b). While the perception of the image in Fig. 2.5 (a) varies from person to person, generally, one feels that it is a small dark disk in the middle of a larger and brighter one. In the extreme case, the



(a) (b)

Figure 2.5: (a) Two concentric disk image; (b) intensity values along the diameter of the disks.

inner disk is perceived as having constant intensity values. However, as shown in Fig. 2.5 (b), in the regions near the center of the disks and the border of the bigger disk the intensity values are, in fact, identical.

We now describe this phenomenon quantitatively. We denote the set of lower and upper edge brims by $\mathcal{B}$ representing the two circular edges of the inner and outer disks in Fig. 2.5 (a). We denote the image in Fig. 2.5 (a) by $\mathcal{X}$ and the image of the visual perception by $\mathcal{X}_p = \{(i, j, x^p_{i,j})\}$. Then we conjecture that $\mathcal{X}_p$ is the solution of the following minimization problem:

$$\min_{\{y_{i,j}\}} \lambda \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} (y_{i,j} - x_{i,j})^2 + V_{\mathcal{Y}}, \tag{2.4}$$

subject to $\mathcal{Y} \cap \mathcal{B} = \mathcal{B}$, where $\lambda \geq 0$ is a weighting factor on the squared errors between $\mathcal{X}$ and $\mathcal{Y}$ in the objective function, $\mathcal{Y} = \{(i, j, y_{i,j})\}$, and $\mathcal{B} \neq \emptyset$. Notice that when $\lambda = 0$, $\mathcal{X}_p$ would be an image of two disks with constant intensity

values for the inner disk and a linear transition of intensity values on the outer disk, which corresponds to the extreme case mentioned before. When λ gets larger, $\mathcal{X}_p$ would be closer to $\mathcal{X}$ in the Euclidean sense, and would have larger variation energy than the case with $\lambda = 0$.

We summarize the psychovisual results described in the above three subsections by stating that basically the edge information in an image is responsible for the visual perception, stronger edges are of higher importance to the visual perception and the smooth areas influence the visual perception together with the edge information.

In the next two sections, we will introduce a three-component image model based on these observations.

2.3 Characterization of Strong Edge Information and Generation of the Stressed Image

As pointed out in Section 2.2, for the human visual perception, the strong edge information of an image plays a more important role as compared with textures and smooth areas. Therefore, in designing image coding systems, strong edges need special attention. To do this, we may first characterize and then extract this strong edge information so that it can be treated separately. In what follows, we will consider the characterization of strong edge information which leads to the concept of a stressed image.

2.3.1 Characterization of Strong Edge Information

In this subsection, we first provide a mathematical description of edges, strong or weak, and then develop a scheme for the discrimination of strong edges.

In what follows, we explain briefly a traditional way of describing edges [22] [23] [25]. For an image $\mathcal{X} = \{(i, j, x_{i,j}),\ i, j = 0, 1, \ldots, M-1\}$, we define the directional variations $v^r_{i,j}$ and $v^c_{i,j}$ at pixel (i, j),

$$v^r_{i,j} = x_{i,j} - x_{i,j+1}, \tag{2.5}$$

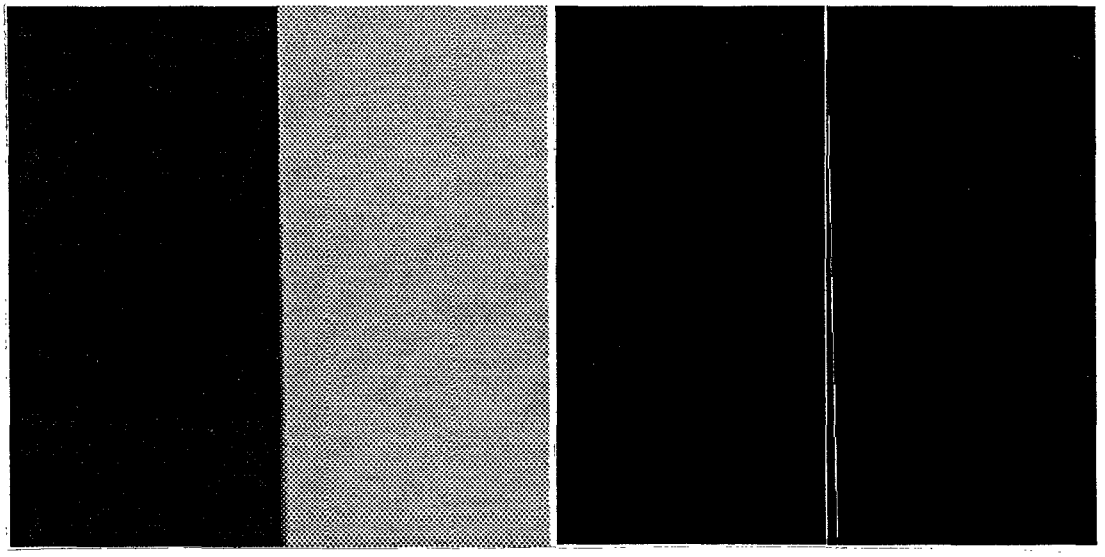
for $i = 0, 1, \ldots, M-1$, and $j = 0, 1, \ldots, M-2$,

$$v^c_{i,j} = x_{i,j} - x_{i+1,j}, \tag{2.6}$$

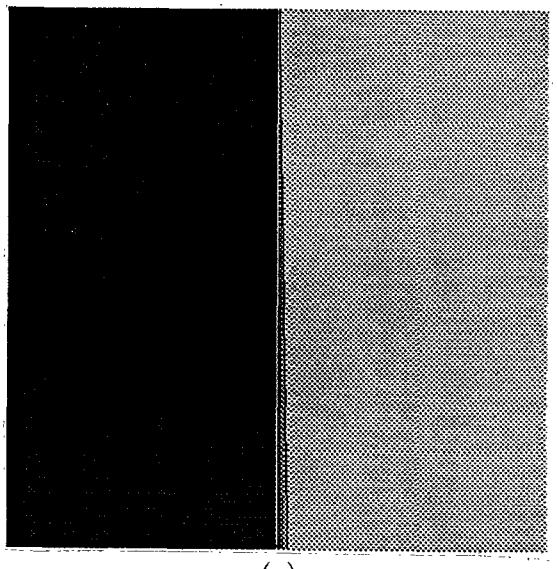
for $i = 0, 1, \ldots, M-2$, and $j = 0, 1, \ldots, M-1$. We define the locations of edges to be those points where $v^r_{i,j}$ and/or $v^c_{i,j}$ assume large absolute values.[2] More precisely, we say that there is an edge point at $(i, j+0.5)$ if $|v^r_{i,j}| > T_v$, and an edge point at $(i+0.5, j)$ if $|v^c_{i,j}| > T_v$, for a positive threshold T_v. We then describe the corresponding edge at location $(i, j+0.5)$ or $(i+0.5, j)$ by the intensity values of neighboring pixels, $x_{i,j}$ and $x_{i,j+1}$, or, $x_{i,j}$ and $x_{i+1,j}$, respectively.

This scheme works well for edges which are one-pixel wide, but not for edges of multi-pixel width as illustrated by the following example. In Fig. 2.6 (a), we have an image with an edge which has an intensity jump from 20 to 220, and a variable-width: one pixel wide at the top and ten pixels wide at the bottom. The intensity values associated with the 256th scanline (center row) in Fig. 2.6 (a)

[2]The actual schemes introduced in [22] [23] [25] are further extensions of the concept here; they are combined with techniques of Gaussian filtering and gradient operator, and will be briefly described in Section 2.5.

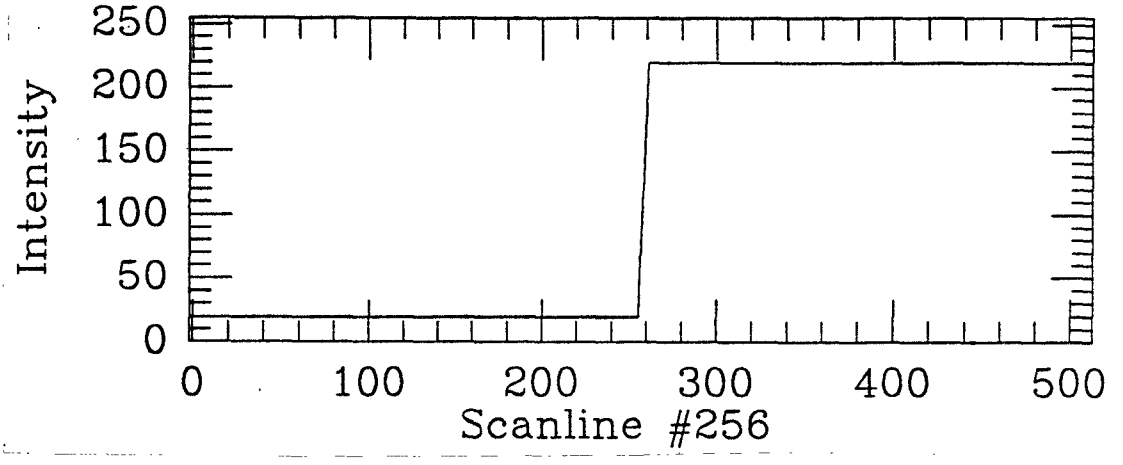


(a) (b)

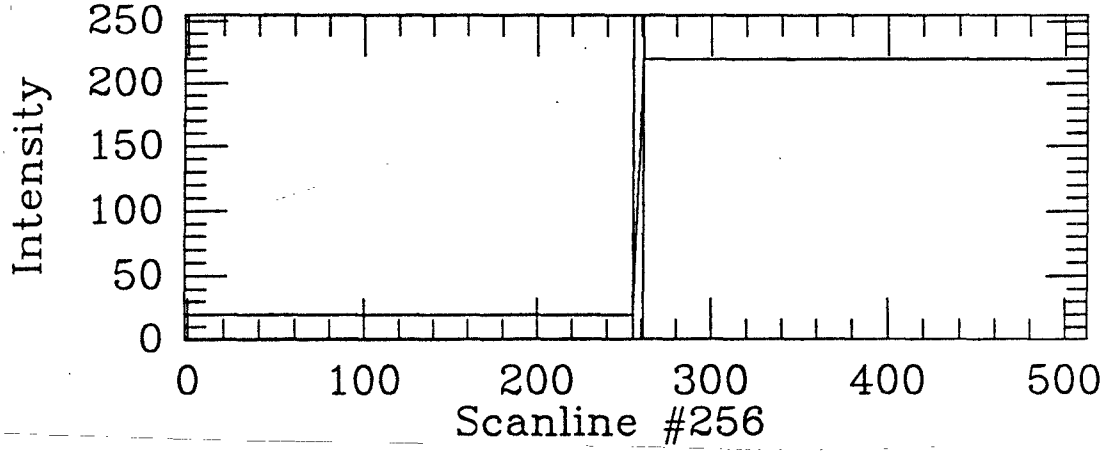


(c)

Figure 2.6: (a) An edge of variant width; (b) lower and upper edge brims; (c) image (a) superposed with image (b).



(a)



(b)

Figure 2.7: (a) Scanline #256 of image (a) in Fig. 2.6; (b) with indications of edge brims.

is shown in Fig. 2.7 (a). On this scanline, the edge is 6 pixels wide: from pixel $(256, 255)$ to $(256, 261)$, where the intensity values change linearly from 20 to 220. Notice that $|x_{256,j} - x_{256,j+1}| = (220 - 20)/6$ for $255 \leq j \leq 260$. Therefore, for $T_v \geq 200/6$, no point on this scanline will be identified as an edge point. On the other hand, for $T_v < 200/6$, the points $(256, j + 0.5)$, $255 \leq j \leq 260$ will all be taken as edge points. While a complete description of the edge on the scanline is provided by these six detected edge points, this description is redundant, since the two intensity values at locations $(256, 255)$ and $(256, 261)$ completely define the edge here. These two pixels, $(256, 255, 20)$ and $(256, 261, 220)$, are referred to as edge brim points in Section 2.2.

We now develop a new way of describing edges in the following. We define the second-order directional variations $D^r_{i,j}$ and $D^c_{i,j}$ for $\mathcal{X}$ at pixel (i, j),

$$D^r_{i,j} = x_{i,j-1} - 2x_{i,j} + x_{i,j+1}, \tag{2.7}$$

for $i = 0, 1, \ldots, M - 1$ and $j = 1, \ldots, M - 2$,

$$D^c_{i,j} = x_{i-1,j} - 2x_{i,j} + x_{i+1,j}, \tag{2.8}$$

for $i = 1, \ldots, M - 2$ and $j = 0, 1, \ldots, M - 1$. We then use these second-order directional variations to describe the *edge brim points* and then the corresponding edges. More specifically, we define the *pixel row-curvature energy* $C^r_{i,j}$ and the *pixel column-curvature energy* $C^c_{i,j}$ by

$$C^r_{i,j} \equiv \begin{cases} (D^r_{i,j})^2 & \text{for } i = 0, 1, \ldots, M - 1 \text{ and } j = 1, \ldots, M - 2, \\ 0 & \text{otherwise,} \end{cases} \tag{2.9}$$

and,

$$C^c_{i,j} \equiv \begin{cases} (D^c_{i,j})^2 & \text{for } i = 1, \ldots, M - 2 \text{ and } j = 0, 1, \ldots, M - 1, \\ 0 & \text{otherwise,} \end{cases} \tag{2.10}$$

respectively, and will refer to them collectively as *pixel curvature energies.* We then define the set $\mathcal{B}_T(\mathcal{X})$ of edge brim points of $\mathcal{X}$ as follows,

$$\mathcal{B}_T(\mathcal{X}) \equiv \left\{(i,j,x_{i,j}) :\ C^r_{i,j} > T \text{ or } C^c_{i,j} > T\right\}, \tag{2.11}$$

where $T \geq 0$. We apply this characterization of edge brim points to the image in Fig. 2.6 (a) with a threshold $T = 32$; the detected edge brim points are indicated by bright points in Fig. 2.6 (b). We superimpose the image (b) on image (a), and show the result in image (c) where dark points are used for the upper edge brim points. These detected edge brim points are exactly at the places where the edge starts and ends. To demonstrate this, we repeat Fig. 2.7 (a) along with two vertical lines indicating the edge brim points in Fig. 2.7 (b). For the case of Fig. 2.6, the set $\mathcal{B}_T(\mathcal{X})$ completely defines the edge; in fact, we can recover the image in Fig. 2.6 (a) merely from the information in $\mathcal{B}_T(\mathcal{X})$ by solving the minimization problem (2.3).

We note that definition (2.11) gives a set of edge brim points regardless of whether they correspond to strong edges. To circumvent this problem, we introduce the concept of a *stressed image*: $\mathcal{X}^s = \{(i,j,x^s_{i,j})\}$ associated with the image $\mathcal{X}$. We first state here the properties which the stressed image $\mathcal{X}^s$ is required to possess, and then, in the next subsection, develop a scheme to generate the stressed image. We require that, at strong edges, the stressed image $\mathcal{X}^s$ closely approximate the original image $\mathcal{X}$, i.e., the squared-errors, $(x_{i,j} - x^s_{i,j})^2$, are small at these pixels; the pixel curvature energies of $\mathcal{X}^s$ of these pixels have no additional constraint other than that imposed by the squared-errors constraint. In other areas such as smooth and texture areas, we require the pixel curvature energies of $\mathcal{X}^s$ to be small while placing a loose constraint on the squared-error $(x_{i,j} - x^s_{i,j})^2$. Thus $\mathcal{B}_T(\mathcal{X}^s)$, as compared with $\mathcal{B}_T(\mathcal{X})$, will only

contain the edge brim points corresponding to strong edges because large pixel curvature energies occur only at the strong edges. Note that this stressed image is generally smooth, except at strong edges where it usually assumes large pixel curvature energies, and thus is "stressed". Note also that in the edge extraction schemes introduced in [22] [23] [25], the textures are suppressed by uniformly smoothing the image. While here we also smooth the image to suppress the textures, we do the smoothing *non-uniformly.* The term "stressed image" is used to indicate this non-uniformity of smoothing.

2.3.2 Generation of the Stressed Image

In this subsection, we consider the generation of a stressed image $\mathcal{X}^s$ from an original image $\mathcal{X}$. Since the main property of the stressed image $\mathcal{X}^s$ is described in terms of (i) the squared-errors $(x_{i,j}-x^s_{i,j})^2$ and (ii) the pixel curvature energies of $\mathcal{X}^s$, we consider the following quantity at pixel (i,j), which is a combination of the above two quantities,

$$E_{i,j}(\lambda^1_{i,j},\lambda^2_{i,j},\lambda^3_{i,j}) = \lambda^1_{i,j}(x_{i,j}-x^s_{i,j})^2 + \lambda^2_{i,j}C^r_{i,j} + \lambda^3_{i,j}C^c_{i,j}, \tag{2.12}$$

where parameters $\lambda^1_{i,j},\lambda^2_{i,j},\lambda^3_{i,j}$ are three non-negative real numbers. We define the summation of these $E_{i,j}$ as

$$E(\mathcal{X}^s,\mathcal{X},\Lambda) \equiv \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} E_{i,j}(\lambda^1_{i,j},\lambda^2_{i,j},\lambda^3_{i,j}), \tag{2.13}$$

where Λ represents the collection of the parameters $\lambda^1_{i,j},\lambda^2_{i,j}$ and $\lambda^3_{i,j}$.

We then consider the following minimization problem for a given parameter set Λ

$$\min_{\{y_{i,j}\}} E(\mathcal{Y},\mathcal{X},\Lambda), \tag{2.14}$$

where $\mathcal{Y} \equiv \{(i,j,y_{i,j})\}$. Note that the objective function $E(\mathcal{Y},\mathcal{X},\Lambda)$ is convex as a function of $\mathcal{Y}$, since each $E_{i,j}(\lambda^1_{i,j},\lambda^2_{i,j},\lambda^3_{i,j})$ is a convex function of $\{y_{i,j}\}$ [26]. Therefore, $\{y^*_{i,j}\}$ is a solution of this minimization problem if and only if it is a solution of the following system of linear equations,

$$\nabla_{\{y_{i,j}\}} E(\mathcal{Y},\mathcal{X},\Lambda) \equiv 0, \tag{2.15}$$

where $\nabla_{\{y_{i,j}\}} E(\mathcal{Y},\mathcal{X},\Lambda)$ denotes the gradient of $E(\mathcal{Y},\mathcal{X},\Lambda)$ with respect to $\{y_{i,j}\}$. We may write the objective function in problem (2.14) in matrix notation:

$$[y_{i,j}]^T L[y_{i,j}] - 2[\lambda^1_{i,j}x_{i,j}]^T[y_{i,j}] + [\lambda^1_{i,j}x_{i,j}]^T[x_{i,j}], \tag{2.16}$$

where vectors $[y_{i,j}] \in \mathbb{R}^{M^2}$ and $[\lambda^1_{i,j}x_{i,j}] \in \mathbb{R}^{M^2}$ contain elements $y_{i,j}$ and $\lambda^1_{i,j}x_{i,j}$, respectively, in a certain order (one-to-one function) r: $\{0,1,\ldots,M-1\} \times \{0,1,\ldots,M-1\} \rightarrow \{0,1,\ldots,M^2-1\}$, i.e., the $(r(i_0,j_0))$th element in the vectors $[y_{i,j}]$ and $[\lambda^1_{i,j}x_{i,j}]$ are y_{i_0,j_0} and $\lambda^1_{i_0,j_0}x_{i_0,j_0}$, respectively. The matrix $L = [\ell_{u,v}]$ is a non-negative definite matrix, where $\ell_{u,v}$ is the coefficient of the term $y_{r^{-1}(u)}y_{r^{-1}(u)}$ when $u = v$, and half of the coefficient of the term $y_{r^{-1}(u)}y_{r^{-1}(v)}$ when $u \neq v$, in $E(\mathcal{Y},\mathcal{X},\Lambda)$. With this matrix notation, (2.15) simplifies to the following:

$$L[y_{i,j}] = [\lambda^1_{i,j}x_{i,j}]. \tag{2.17}$$

We now show the existence and uniqueness of the solution for the minimization problem (2.14) when $\lambda^1_{i,j} > 0$ for all (i,j).

Lemma 2.3.1 *The matrix L is positive definite if $\lambda^1_{i,j} > 0$ for all (i,j).*

Proof: Setting $x_{i,j} = 0$ for all (i,j) in the objective function (2.16), we have

$$E(\mathcal{Y},\mathcal{Z},\Lambda) = [y_{i,j}]^T L[y_{i,j}], \tag{2.18}$$

where $\mathcal{Z} = \{(i,j,0)\}$. While

$$E(\mathcal{Y},\mathcal{Z},\Lambda) = \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} \lambda_{i,j}^1 (y_{i,j})^2 + \lambda_{i,j}^2 C_{i,j}^r + \lambda_{i,j}^3 C_{i,j}^c, \tag{2.19}$$

we have that $[y_{i,j}]^T L[y_{i,j}] = 0$ if and only if $y_{i,j} = 0$ for all (i,j), when $\lambda_{i,j}^1 > 0$ for all (i,j) (under the assumption that $\lambda_{i,j}^2, \lambda_{i,j}^3$ are non-negative). QED.

From now on, we assume that the parameter set Λ is such that all $\lambda_{i,j}^1$ are greater than zero, and therefore we will talk about *the* solution of problem (2.14), which is $L^{-1}[\lambda_{i,j}^1 x_{i,j}]$. The problem (2.14) is solved iteratively using the Gauss-Seidel iteration and with the Multi-Grid technique to speed up the convergence. The Gauss-Seidel iteration and the Multi-Grid technique are well known [27]. Nevertheless, the details of the implementation of these two techniques are included in Appendix A.

We now consider the influences of the choices of $\lambda_{i,j}^1, \lambda_{i,j}^2$ and $\lambda_{i,j}^3$ on the solution, $\mathcal{Y}^* = \{(i,j,y_{i,j}^*)\}$, of problem (2.14). A close examination of (2.12) indicates that if $\lambda_{i,j}^1$ is large compared with $\lambda_{i,j}^2$ and $\lambda_{i,j}^3$, $(x_{i,j} - y_{i,j}^*)^2$ would be small, and the constraint on the pixel curvature energies of $\mathcal{Y}^*$ would be loose; on the other hand, if $\lambda_{i,j}^2$ and $\lambda_{i,j}^3$ are large as compared to $\lambda_{i,j}^1$, the pixel curvature energies of $\mathcal{Y}^*$ would be small, and the constraint imposed on the squared-error $(x_{i,j} - y_{i,j}^*)^2$ would be relaxed. To make this observation more precise, we consider the relationship between the original image $\mathcal{X}$ and the solution $\mathcal{Y}^*$ governed by the minimization problem (2.14) as a filtering operation with input $\mathcal{X}$ and output $\mathcal{Y}^*$. In Appendix C, we investigate the frequency response of this filter (in its continuous form for simplicity) and show that it is a low-pass filter with cutoff frequencies at (i,j) controlled by the parameters $\lambda_{i,j}^1$, $\lambda_{i,j}^2$ and $\lambda_{i,j}^3$, and thus is space-variant. The relationship between the cutoff frequencies and the parameters given in Appendix C can be described as follows: Larger values of

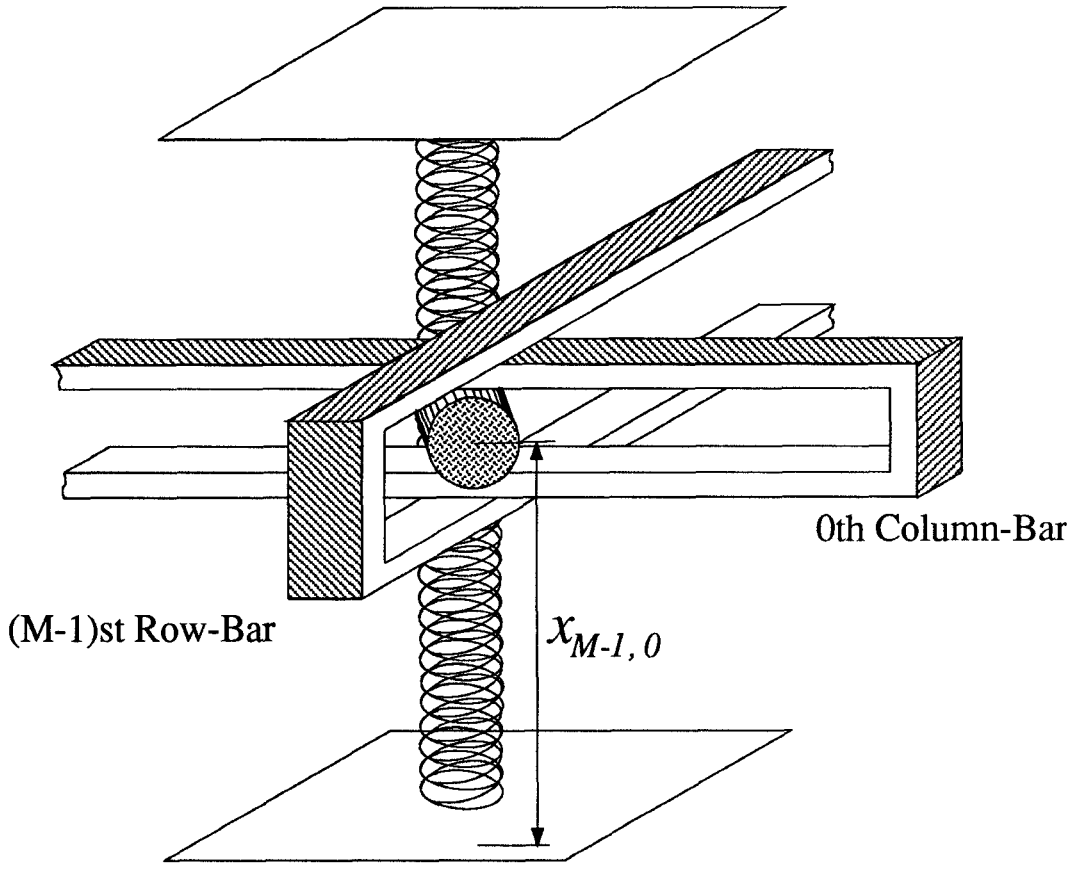


Figure 2.8: Mechanical structure of EMM at pixel $(M-1,0)$.

$\lambda^1_{i,j}$ give higher cutoff frequencies in both directions, while larger values of $\lambda^2_{i,j}$ and $\lambda^3_{i,j}$ lead to lower cutoff frequencies in row-direction and column-direction, respectively. It is important to note that it is not the absolute values of $\lambda^1_{i,j}$, $\lambda^2_{i,j}$, and $\lambda^3_{i,j}$ but rather the ratios $\lambda^2_{i,j}/\lambda^1_{i,j}$ and $\lambda^3_{i,j}/\lambda^1_{i,j}$ which influence the solution $\mathcal{Y}^*$.

The minimization problem (2.14) actually has an interesting interpretation in that the solution $\mathcal{Y}^*$ can be thought of as the stable configuration of a mechanical structure, a part of which is depicted in Fig. 2.8. In this structure, at each pixel location (i,j) we have a vertical spring with both ends fixed on a floor (height

$= 0$) and a ceiling (height $= 255$), and a cylinder fixed on it at height $x_{i,j}$. This cylinder is constrained to move only in the vertical direction. For each row and column we have a flexible bar of the shape shown in Fig. 2.8 with a slot in the middle. The cylinders associated with each (i,j) are fitted inside the corresponding column bars, and the column bars, in turn, are fitted inside row bars. Shown in Fig. 2.8 is the structure at pixel $(M-1,0)$. We now suppose that this mechanical structure assumes a configuration described by the heights $y_{i,j}$ of the cylinders. Then the potential energy of this configuration $\{y_{i,j}\}$ is approximated by $E(\mathcal{Y},\mathcal{X},\Lambda)$ defined in (2.13), where $\lambda^1_{i,j}(x_{i,j}-y_{i,j})^2$, $\lambda^2_{i,j}C^r_{i,j}$ and $\lambda^3_{i,j}C^c_{i,j}$ approximate the potential energies of the spring, the row bar and the column bar at pixel (i,j), respectively. The parameters $\lambda^1_{i,j}, \lambda^2_{i,j}$ and $\lambda^3_{i,j}$ control the rigidness of the spring, the row bar, and the column bar at pixel (i,j) [28].

To visualize the formation of the stable configuration for this mechanical structure, we note that if all the bars are taken out, the cylinders on the vertical springs for each pixel (i,j) will assume their stable positions at $x_{i,j}$, i.e., this configuration represents the original image $\mathcal{X}$. After we slide in all the row and column bars whose rigidness (or flexibility) are determined by the parameters, $\{\lambda^2_{i,j}\}$ and $\{\lambda^3_{i,j}\}$, the configuration of the structure will change, and will reach the final stable configuration $\{y^*_{i,j}\}$ which has the minimum potential energy, i.e., $\mathcal{Y}^*$ is the solution of problem (2.14). Due to this analogy to the mechanical system, we refer to (2.14) as an *Energy Minimization Model* (EMM) problem.

To obtain the stressed image $\mathcal{X}^s$ from the original image $\mathcal{X}$, it suffices to solve the EMM problem with a proper parameter set Λ, namely, small ratios $\lambda^2_{i,j}/\lambda^1_{i,j}$ and $\lambda^3_{i,j}/\lambda^1_{i,j}$ at locations of strong edges and larger ratios at other places.

However, we can not specify the parameter set Λ a priori, since this requires the knowledge of the locations of strong edges - the very purpose of generating the stressed image.

We now proceed to explain qualitatively an approach for solving this problem with the help of the above mechanical structure. We assume that in the process of forming the stressed image $\mathcal{X}^s$, the bars and springs first have uniform rigidness, and the stable configuration reached by the structure is a uniformly smoothed version, denoted by $\mathcal{Y}^1 \equiv \{(i, j, y^1_{i,j})\}$, of the original image $\mathcal{X}$. Then we allow the bars to have more flexibility at locations corresponding to large pixel curvature energies (where the bars are most severely bent), and let the structure stabilize to a new configuration, denoted by $\mathcal{Y}^2 \equiv \{(i, j, y^2_{i,j})\}$. Then again we measure the pixel curvature energies of this new configuration $\mathcal{Y}^2$, and allow higher flexibility of the bars at places of large pixel curvature energies. We continue the process by repeating the above procedure. Notice that the textures are suppressed in the first configuration $\mathcal{Y}^1$, and will continue to be suppressed in later configurations, while the strong edges which correspond to locations of large pixel curvature energies will be gradually approximated in the later configurations since we change the flexibilities of the bars to accentuate the strong edges.

To express the above idea mathematically, we start with a uniform parameter set Λ, i.e., parameter $\lambda^k_{i,j} = \lambda^k$ for all (i, j), $k = 1, 2, 3$, and solve problem (2.14) to get $\mathcal{Y}^1$. We then update the parameter set Λ by changing the ratios, $\lambda^2_{i,j}/\lambda^1_{i,j}$ and $\lambda^3_{i,j}/\lambda^1_{i,j}$, according to $C^r_{i,j}$ and $C^c_{i,j}$ of $\mathcal{Y}^1$. The updating strategy should be such that large values of $C^r_{i,j}$ and $C^c_{i,j}$ give rise to small ratios, $\lambda^2_{i,j}/\lambda^1_{i,j}$ and $\lambda^3_{i,j}/\lambda^1_{i,j}$, respectively. In the algorithm proposed here, we set $\lambda^2_{i,j}/\lambda^1_{i,j}$ and $\lambda^3_{i,j}/\lambda^1_{i,j}$

to be inversely proportional to $C^r_{i,j}$ and $C^c_{i,j}$, i.e.,

$$\lambda^2_{i,j}/\lambda^1_{i,j} = \beta/C^r_{i,j} \text{ and } \lambda^3_{i,j}/\lambda^1_{i,j} = \beta/C^c_{i,j} \tag{2.20}$$

where β is a constant. Then we solve problem (2.14) again with the new parameter set to obtain $\mathcal{Y}^2$. We repeat the above iteration until the relative variation of the objective function in (2.14) for the two consecutive iterations is smaller than a given threshold. The final solution $\mathcal{Y}^K$, where K is the total number of iterations, will be called a stressed image (denoted by $\mathcal{X}^s$ before) of the image $\mathcal{X}$.

2.4 Extraction of Strong Edge Information and a Three-Component Image Model

As indicated before, the strong edge information can be extracted by identifying pixels of large pixel curvature energies in a stressed image $\mathcal{X}^s$, and is represented by $\mathcal{B}_T(\mathcal{X}^s)$ in (2.11). While generating $\mathcal{B}_T(\mathcal{X}^s)$ based on (2.11) is straightforward, the choice of the threshold T is crucial and fine-tuning of T is computationally extensive. To circumvent these difficulties, further selection of the pixels in $\mathcal{B}_T(\mathcal{X}^s)$ is made by identifying pixels of large *local maximum* pixel curvature energies as the resulting edge brim points. More specifically, the definition of edge brim points in (2.11) is modified as follows:

$$\begin{aligned}
(i,j,x^s_{i,j}) \quad & \in \mathcal{B}_T(\mathcal{X}^s) \\
\text{if} \quad & D^r_{i,j-1}D^r_{i,j} \le 0,\ D^r_{i,j}D^r_{i,j+1} \le 0, \text{ and } C^r_{i,j} > T; \\
\text{or if} \quad & D^r_{i,j-1}D^r_{i,j} > 0,\ D^r_{i,j}D^r_{i,j+1} \le 0, \text{ and } C^r_{i,j} > \max\{C^r_{i,j-1}, T\}; \\
\text{or if} \quad & D^r_{i,j-1}D^r_{i,j} \le 0,\ D^r_{i,j}D^r_{i,j+1} > 0, \text{ and } C^r_{i,j} > \max\{C^r_{i,j+1}, T\};
\end{aligned}$$

$$
\begin{aligned}
&\text{or if} \quad D^r_{i,j-1}D^r_{i,j} > 0,\ D^r_{i,j}D^r_{i,j+1} > 0,\ \text{and}\ C^r_{i,j} > \max\{C^r_{i,j-1}, C^r_{i,j+1}, T\};\\
&\text{or if} \quad D^c_{i-1,j}D^c_{i,j} \le 0,\ D^c_{i,j+1}D^c_{i,j} \le 0,\ \text{and}\ C^c_{i,j} > T;\\
&\text{or if} \quad D^c_{i-1,j}D^c_{i,j} > 0,\ D^c_{i,j+1}D^c_{i,j} \le 0,\ \text{and}\ C^c_{i,j} > \max\{C^c_{i-1,j}, T\};\\
&\text{or if} \quad D^c_{i-1,j}D^c_{i,j} \le 0,\ D^c_{i,j}D^c_{i+1,j} > 0,\ \text{and}\ C^c_{i,j} > \max\{C^c_{i+1,j}, T\};\\
&\text{or if} \quad D^c_{i-1,j}D^c_{i,j} > 0,\ D^c_{i,j}D^c_{i+1,j} > 0,\ \text{and}\ C^c_{i,j} > \max\{C^c_{i-1,j}, C^c_{i+1,j}, T\}.
\end{aligned}
\tag{2.21}
$$

Therefore, the set $\mathcal{B}_T(\mathcal{X}^s)$ contains pixels not only of high pixel curvature energies, but also of curvature energies larger than those of the neighboring pixels with the same sign of the second-order variations.

To complete the process of the extraction of the strong edge information, we will consider the generation of the contours of edge brims from $\mathcal{B}_T(\mathcal{X}^s)$ in the next subsection. The formation of these contours is also important for the coding of strong edge information (ref. Chapter 3).

2.4.1 Generation of Edge Brim Contours

We now consider the problem of generating contours from the set $\mathcal{B}_T(\mathcal{X}^s)$ described in (2.21). We define a contour as a sequence of triples:

$$\overline{b} \equiv \{(i^k, j^k, x^s_{i^k,j^k})\}, \tag{2.22}$$

$k = 0, 1, \ldots, m-1$, such that

$$|i^{k-1} - i^k| \le 1,\ |j^{k-1} - j^k| \le 1, \tag{2.23}$$

and $(i^{k-1}, j^{k-1}) \neq (i^k, j^k)$, for $k = 1, \ldots, m-1$, and the *maximum-variation* of intensity values

$$\sigma_{\overline{b}} \equiv \max_{k=0,1,\ldots,m-1} |x^s_{i^k,j^k} - \overline{x}| \ \le T_c, \tag{2.24}$$

where $T_c \geq 0$, $\overline{x}$ is the average of the pixel intensity values on $\overline{b}$, and m is called the *length* of the contour. This definition is different from previous contour definitions [29] - [31] in condition (2.24); condition (2.24) is introduced to prevent a mixing of the upper and lower edge brims into one contour. In the following, we will describe the details of the contour extraction procedure used here.

Let us first consider the so-called *contour tracing* problem [29] - [31], which ignores condition (2.24). This problem is solved by a local search algorithm. To describe this algorithm, consider Fig. 2.9 where a generic pixel (i, j) is circled, and its neighboring pixels are labeled by indices $0, 1, \ldots, 7$. Starting from a pixel $(i^0, j^0) \in f_1(\mathcal{B}_T(\mathcal{X}^s))$, the tracing algorithm searches among its neighboring pixels (starting from 0 in the clockwise direction) for one in $\mathcal{B}_T(\mathcal{X}^s)$. As soon as the algorithm finds a pixel $(i^1, j^1) \in f_1(\mathcal{B}_T(\mathcal{X}^s))$ in these neighboring pixels, it moves the search center to (i^1, j^1), and repeats the search again until the search fails. The resulting sequence of pixels, $(i^0, j^0, x^s_{i^0,j^0})$, $(i^1, j^1, x^s_{i^1,j^1}), \ldots$, defines the contour extracted.

Suppose that we have a contour $(i^0, j^0, x^s_{i^0,j^0})$, $(i^1, j^1, x^s_{i^1,j^1})$, $\ldots$, $(i^{m-1}, j^{m-1}, x^s_{i^{m-1},j^{m-1}})$ satisfying the geometric condition (2.23). We now impose the constraint (2.24) on this contour. While the whole contour may not satisfy this constraint for a given threshold T_c, short segments of the contour are likely to satisfy this condition. Thus this added constraint essentially breaks the contour into segments, which leads to a problem similar to the *straight-line-fitting* problem in [32], with the exception that the straight-line-segments in [32] are replaced by constant-intensity curve-segments here.

The contour extraction algorithm actually implemented is a combination of the above techniques and works as follows. Starting from a pixel $(i^0, j^0, x^s_{i^0,j^0}) \in$

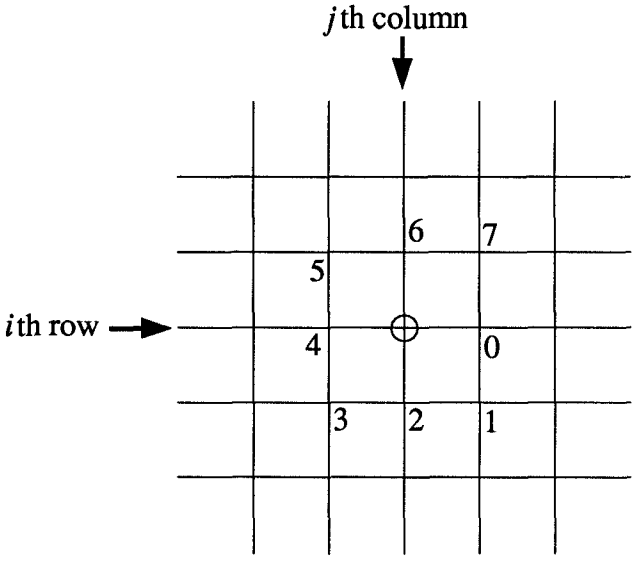


Figure 2.9: The eight neighboring pixels.

$\mathcal{B}_T(\mathcal{X}^s)$, in its eight neighboring pixels we search for pixels in $\mathcal{B}_T(\mathcal{X}^s)$, among which we choose a pixel that gives the minimum maximum-variation; if this maximum-variation is also lower than T_c, we take this pixel as the next pixel $(i^1, j^1, x^s_{i^1,j^1})$ on the contour. Then we move the search center to $(i^1, j^1, x^s_{i^1,j^1})$, delete $(i^0, j^0, x^s_{i^0,j^0})$ from $\mathcal{B}_T(\mathcal{X}^s)$, and repeat the above procedure again. The process stops when we cannot locate a pixel in $\mathcal{B}_T(\mathcal{X}^s)$ having maximum-variation lower than T_c among the neighboring pixels of the current search center. This is the basic form of the algorithm. In the following, we discuss two other considerations in the actual algorithm implemented, namely, *back-tracing* and *prediction.*

Note that when we take an arbitrary pixel in $\mathcal{B}_T(\mathcal{X}^s)$ to begin a contour tracing, we might start in the middle of an actual contour, and would only get a portion of this contour using our basic algorithm. We circumvent this problem by introducing a flag such that when in the first search step we have more than

one neighboring pixel in $\mathcal{B}_T(\mathcal{X}^s)$ giving maximum-variations lower than T_c, this flag is set to "1", and otherwise, it is set to "0". When this flag is "1", we first trace to one end of the contour, and then trace back to the other end of the contour.

Due to the iterative nature of the algorithm for generating the stressed image, it is possible that we lose a few pixels on some edge brim contours for a given threshold T_c. Some of the missing pixels may be recovered by a predictive process based the information of the generated contours. The details of the prediction is included in Appendix D.

Having determined all contours with the above algorithm, we eliminate those contours of small length, namely, those with length smaller than a certain threshold T_ℓ. Since strong edges are represented by two contours, namely, lower and upper edge brims, we may use this pairing property to further reduce spurious contours. More precisely, let us define a d-neighborhood of a pixel $(i, j, x^s_{i,j})$ as follows,

$$N(d, (i^k, j^k, x^s_{i^k,j^k})) \equiv \{(i, j, .) : \max\{|i - i^k|, |j - j^k|\} \leq d\}, \tag{2.25}$$

where $d \geq 0$. We say that a pixel in $\mathcal{B}_T(\mathcal{X}^s)$ is paired in distance d if and only if the d-neighborhood of this pixel contains pixels on another contour of $\mathcal{B}_T(\mathcal{X}^s)$; and that a contour in $\mathcal{B}_T(\mathcal{X}^s)$ is paired in distance d if and only if it has at least T_ℓ pixels being paired in distance d. Having checked all contours in $\mathcal{B}_T(\mathcal{X}^s)$, we eliminate those contours not paired in distance d.

We will denote the final collection of the pixels on the remaining contours by $\mathcal{B}_T(\mathcal{X}^s)$ again, and we will refer to this set as the collection of extracted strong edge brim contours.

2.4.2 A Three-Component Image Model

We now proceed to generate an image $\mathcal{P} = \{(i, j, p_{i,j})\}$, $i, j = 0, 1, ..., M-1$, solely from $\mathcal{B}_T(\mathcal{X}^s)$, such that the difference image $\mathcal{X} \ominus \mathcal{P} \equiv \{(i, j, x_{i,j} - p_{i,j})\}$ has no strong edges represented by $\mathcal{B}_T(\mathcal{X}^s)$. We generate the image $\mathcal{P}$ based on the minimum information principle introduced in Section 2.2 (ref. eq. (2.3)). The detailed algorithm is included in Appendix B. We will call the image $\mathcal{P}$ the *primary image* (or *primary component*) of the original image $\mathcal{X}$, since the strong edge information contained in image $\mathcal{P}$ is most important to human visual system.

Notice that, the stressed image $\mathcal{X}^s$, from which the set $\mathcal{B}_T(\mathcal{X}^s)$ is derived, can be considered as the output of a space-variant filter for the input $\mathcal{X}$. This filter is basically low-pass except at locations of strong edges. Thus the stressed image $\mathcal{X}^s$ contains the low-frequency component, or smooth component, and the strong edge information of the image $\mathcal{X}$. Therefore, the difference image $\mathcal{X} \ominus \mathcal{X}^s$ consists of the high-frequency component, but without the strong edge information. We define $\mathcal{T} = \mathcal{X} \ominus \mathcal{X}^s = \{(i, j, t_{i,j})\} = \{(i, j, x_{i,j} - x^s_{i,j})\}$ and $\mathcal{S} = \mathcal{X}^s \ominus \mathcal{P} = \{(i, j, s_{i,j})\} = \{(i, j, x^s_{i,j} - p_{i,j})\}$ corresponding to the texture and the smooth component, respectively. Thus, the image $\mathcal{X}$ can be expressed as

$$\mathcal{X} \equiv \mathcal{T} \oplus \mathcal{S} \oplus \mathcal{P} \equiv \{(i, j, t_{i,j} + s_{i,j} + p_{i,j})\}. \tag{2.26}$$

In the sequel, this model will be referred to as the *three-component image model.* The block diagram for generating these three components is provided in the next chapter, Fig. 3.1, where the quantization of $\mathcal{B}_T(\mathcal{X}^s)$ is considered.

In the next section, we present an experiment for this three-component image model and a comparison with the LGO.

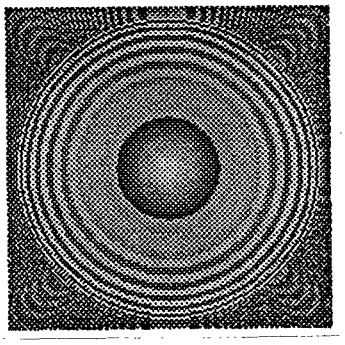


Figure 2.10: Test image.

2.5 Experimental Results and Comparisons

Consider the test image shown in Fig. 2.10. This 256×256 image has a circular strong edge in the middle and textures or smooth areas in the rest of the image. A scanline (row 128) of the test image is plotted in Fig. 2.11. Note that around the center of the scanline we have a convex curve with the shape of a sine wave from 0 to π, then two strong edges, and up-down waveforms with linearly increasing frequency and amplitudes.

To get a picture of the edge location of the edge, we draw a grid in Fig. 2.12 with the intersections of vertical and horizontal lines indicating the pixel locations; the upper and lower edge brims are indicated by small square-dots (Fig. 2.12 is only the center portion of the image). Two typical edge segments are shown in Fig. 2.13. Note that the width of the edge is not limited to one pixel in contrast with the restriction in [22] [23] (see Subsection 2.5.2).

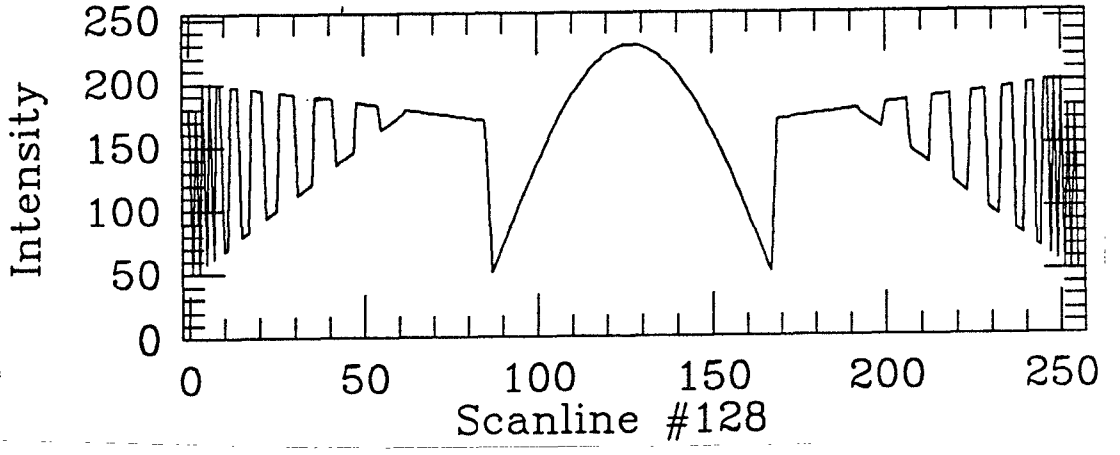


Figure 2.11: Scanline #128 of the test image.

2.5.1 An Example of the Three-Component Image Model

The stressed image is generated with the scheme developed in Section 2.3. For simplicity, we assumed that $\lambda^2_{i,j} = \lambda^3_{i,j} \equiv \lambda_{i,j}$, for all (i,j). Thus $\lambda_{i,j}/\lambda^1_{i,j}$ is the only independent parameter. Therefore, instead of updating two parameters $\lambda^2_{i,j}$, $\lambda^3_{i,j}$, we may simply update $\lambda^1_{i,j}$ by the formula: $\lambda^1_{i,j} = \gamma(C^r_{i,j} + C^c_{i,j})$, ($\gamma$ is a constant), which is similar reciprocally to (2.20).

The stressed image is shown in Fig. 2.14 (a), which is clearly stressed only at the strong edge of the test image. The computer time used on a SPARCstation-1 is about 1400 seconds. Edge brim points and the corresponding contours are extracted by the scheme introduced in Section 2.4, with $T = 512$, $T_c = 32$, $T_\ell = 16$ and $d = 3$ (the number of iterations used to solve Equation (2.15) is 40). The resulting two contours are shown in Fig. 2.14 (b). An enlarged picture of these contours is shown in Fig. 2.15. These two contours correspond to the lower and upper edge brims. In Fig. 2.16, we superimpose Fig. 2.15 and Fig. 2.12 to

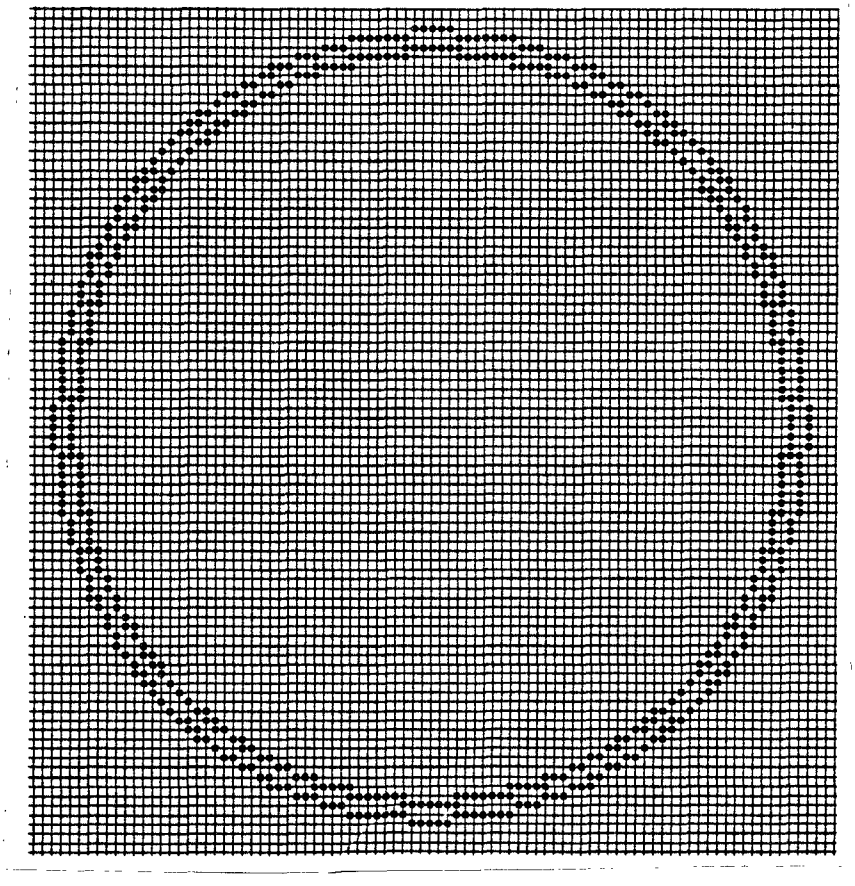


Figure 2.12: Enlarged picture of the edge-brim locations in the test image.

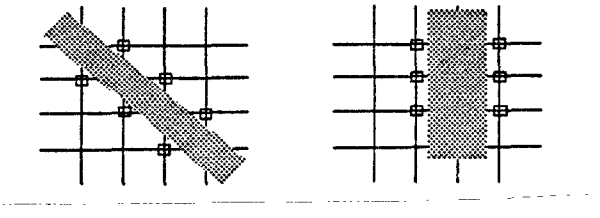
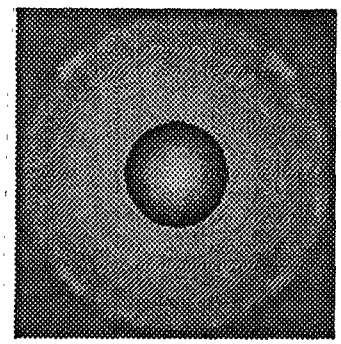
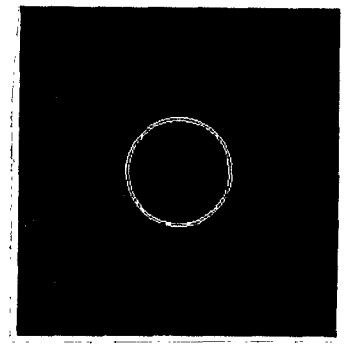


Figure 2.13: Typical segments of the edge in the test image.



(a) The stressed image



(b) The picture of edge brim contours

Figure 2.14: The stressed image and the picture of edge brim contours extracted for the test image.

demonstrate the effectiveness of our scheme in locating the strong edge.

To better understand the behavior of the stressed image, we have plotted scanline 128 for the test image (solid curve) and for the stressed image (dotted curve) in Fig. 2.17 (a), and with vertical lines indicating the locations of edge brim contours on this scanline in Fig. 2.17 (b). Note that the scanline associated with the stressed image is smooth except at the strong edge locations where it matches the edge, and that the locations of edge brim contours coincide with the upper and lower edge brims of the strong edge.

The primary image associated with the extracted strong edge is shown in Fig. 2.18. The other two components, namely, the smooth and texture components, are derived as in Subsection 2.4.2. Notice that the smooth and texture components are differences of two images, and thus may have negative intensity values. We have added constants to these two components in order to bring their intensity values in the displayable range, namely, $[0, 255]$. The resulting images of these two components are shown in Fig. 2.19. Note that the primary

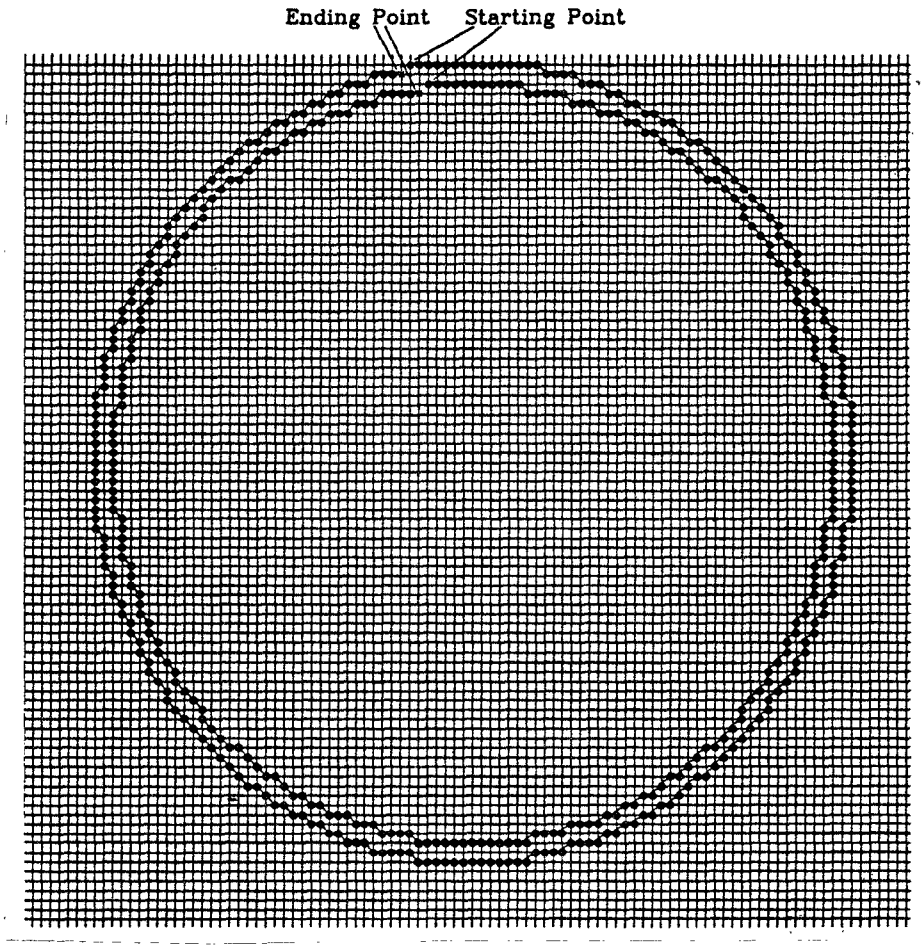


Figure 2.15: Enlarged picture of the contours extracted from the stressed image shown in Fig. 2.14 (a).

image contains the strong edge information, the smooth component provides the background slow intensity variations and the texture component contains all the textures.

Note that there are some traces of brighter pixels at around the location of the circular edge in the texture image. The reason for this phenomenon is basically the orientation anisotropy of the EMM problem since there are only two directions for the bars and the discrete structure of the problem (ref. Section 2.3). This orientation anisotropy can be most clearly seen for edge segments with an

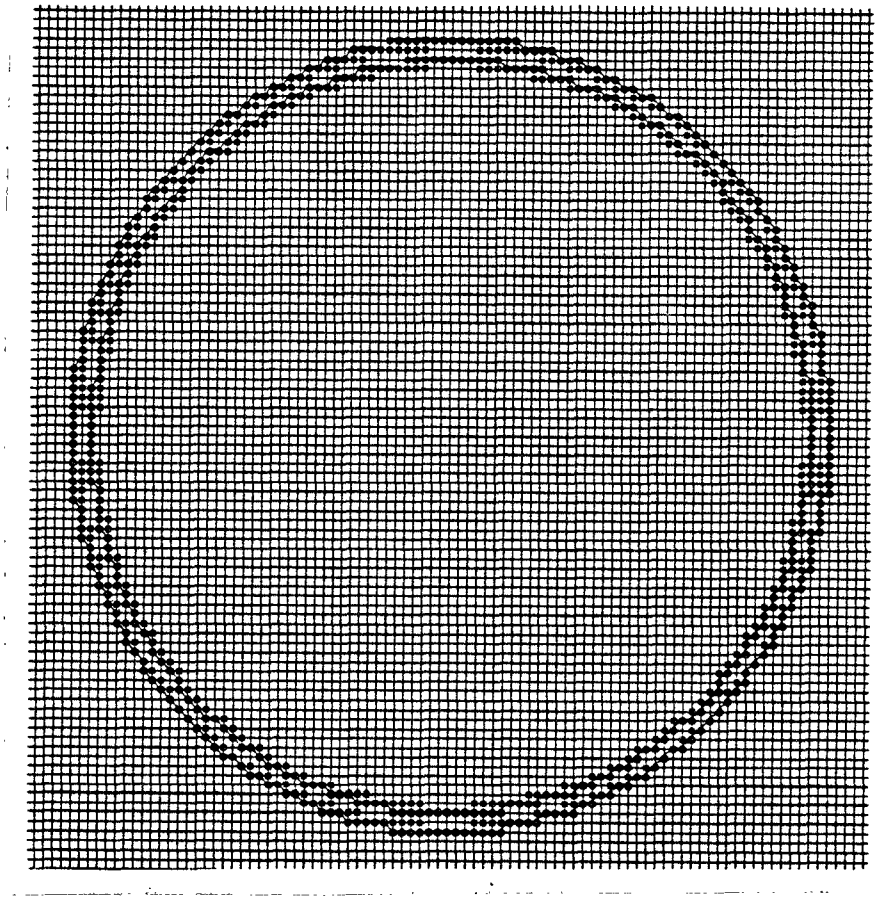
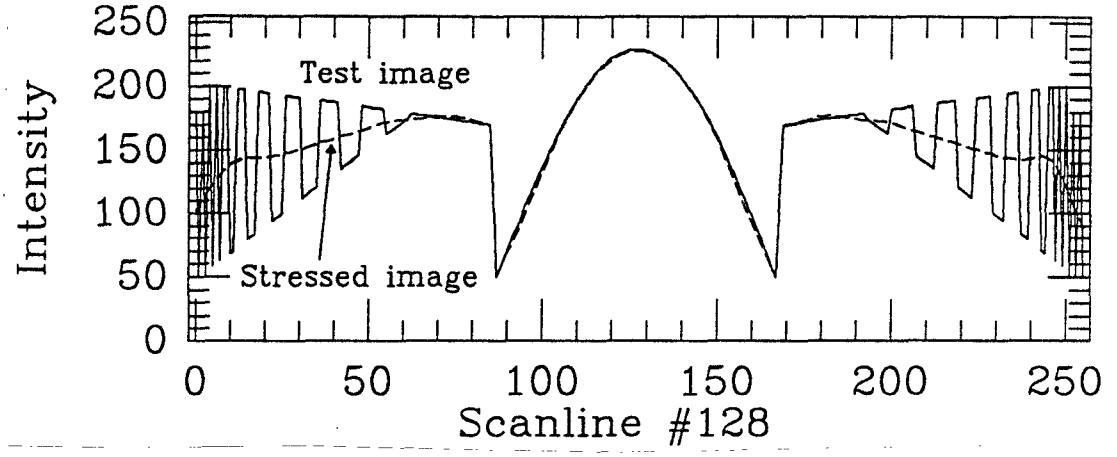


Figure 2.16: The superimposition of Fig. 2.15 and Fig. 2.12.

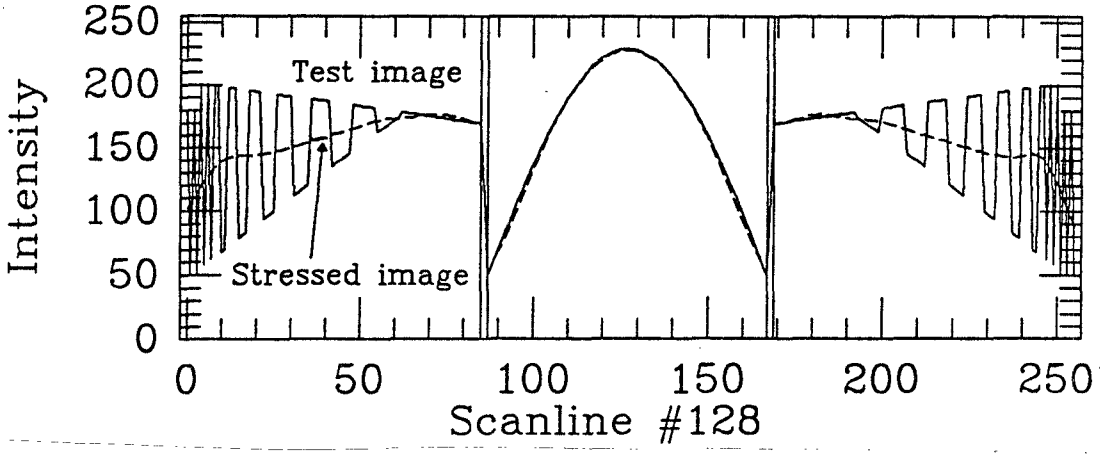
orientation slightly tilted away from the vertical or the horizontal directions. However, since the primary image gives a perceptually good copy of the edge, these traces of brighter pixels can be treated as textures.

Some other examples of the three-component image model for real-world images are provided in the next chapter where the image coding issues are investigated.

In what follows, we compare the performance of the strong edge extraction of the three-component image model with that of the LGO scheme.



(a) Intensity values along scanline #128 for the test and stressed images.



(b) Vertical lines indicating location of edge brim points on this scanline.

Figure 2.17: Scanline #128 of the test image and the stressed image.

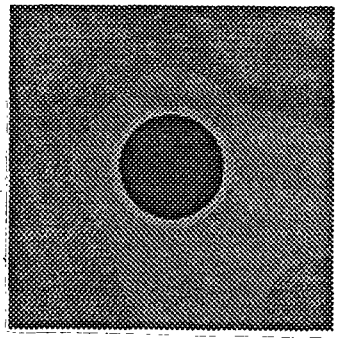
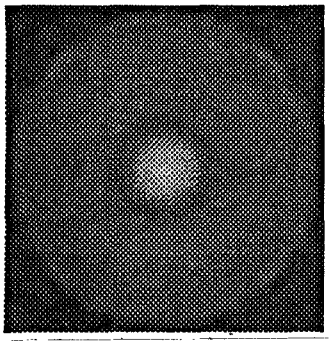
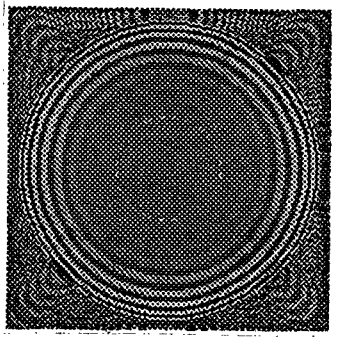


Figure 2.18: The primary image for the test image.



(a) Smooth component



(b) Texture component

Figure 2.19: The smooth and texture components.

2.5.2 The Laplacian-Gaussian Operator

The LGO scheme refers to an edge extraction scheme which is introduced by Marr and Hildreth [25]. In the following, we provide a brief description of the LGO and discuss some of its applications in the context of image coding [22], [23]. Then some limitations of the LGO are illustrated by an example.

For simplicity, let us consider a real-valued function $x(t)$ of a real variable t. We define the edge locations of $x(t)$ by the zero-crossing locations of the second derivative of $x(t)$. To suppress weak edges, a Gaussian filter is used to smoothen

$x(t)$ before second derivative zero-crossings are located. More precisely, a 1-D Gaussian kernel $g_\sigma(t)$ of standard deviation σ is convolved with $x(t)$:

$$y_\sigma(t) = x(t) \otimes g_\sigma(t) = \int_{-\infty}^{\infty} x(u) \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(t-u)^2}{2\sigma^2})\, du, \tag{2.27}$$

and the points at which $y''_\sigma(t) = 0$ are defined as the edge locations. Note that the standard deviation σ of the Gaussian kernel controls the strengths of the extracted edges, i.e., larger values of σ correspond to higher edge strengths [25].

In the 2-D case, we consider a function $x(t,s)$ of two variables t and s. The edges with strengths above a threshold are defined as the points where the Laplacian of $y_\sigma(t,s)$, $\nabla^2 y_\sigma(t,s) \equiv \partial^2 y_\sigma/\partial t^2 + \partial^2 y_\sigma/\partial s^2$, vanishs, where $y_\sigma(t,s)$ is the convolution of $x(t,s)$ and a 2-D Gaussian kernel: $G_\sigma(r,\theta) = \exp(-r^2/(2\sigma^2))/(2\pi\sigma^2)$, where (r,θ) denotes the polar coordinates, with $r \in [0,\infty)$, $\theta \in [0, 2\pi)$. Note that

$$\nabla^2 y_\sigma(t,s) = \nabla^2(x(t,s) \otimes G_\sigma(r,\theta)) = x(t,s) \otimes (\nabla^2 G_\sigma(r,\theta)), \tag{2.28}$$

and

$$\nabla^2 G_\sigma(r,\theta) = \frac{\partial^2 G}{\partial r^2} + \frac{1}{r}\frac{\partial G}{\partial r} = \frac{1}{\pi\sigma^4}(\frac{r^2}{2\sigma^2} - 1)\exp(-\frac{r^2}{2\sigma^2}). \tag{2.29}$$

To apply the LGO to digital images, the 2-D kernel $\nabla^2 G_\sigma(r,\theta)$ is discretized and convolved with the image $\{x_{i,j}\}$. We denote the result of this convolution by $\{y_{i,j}\}$ and locate its zero-crossings in the horizontal and vertical directions. For example, we say that there is a zero-crossing at $(i, j+0.5)$ if $y_{i,j}y_{i,j+1} \leq 0$. As in [22], we apply a gradient operator across each zero-crossing; if the absolute value of the output of the gradient operator is above a certain threshold, the zero-crossing is taken as an edge location. The resulting edge locations are transmitted along with the intensity values of their neighboring pixels for a complete description of the corresponding edges.

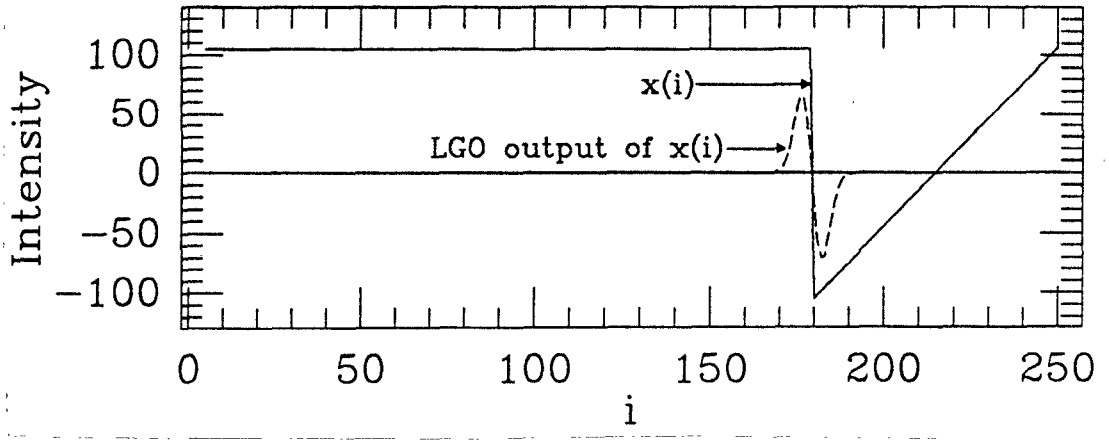


Figure 2.20: An example for the locational accuracy of LGO, $\sigma = 3$.

As indicated above, the standard deviation σ of the Gaussian filter controls the strengths of the extracted edges. Additionally, σ influences another important property of the resulting zero-crossings, namely, the accuracy of their locations with respect to the actual edges. Smaller values of σ give better accuracy, and larger values of σ result in poor accuracy [33]. We illustrate this situation with the following example.

Consider the 1-D signal $x(i)$ shown by the solid curve in Fig. 2.20 with an edge at $i = 179.5$. More specifically, $x(i) = 105$ for $i < 180$, $x(180) = -105$ and $x(i)$ linearly increases afterward. The dashed curve in Fig. 2.20 is a normalized version of $-y''_\sigma(i)$ (ref. eq. (2.27)) for $\sigma = 3$. Notice that $-y''_\sigma(i)$ has an isolated zero-crossing at $i = 179.5$. Thus, in this case the LGO locates the edge accurately; the intensity values of the left and right pixels of this zero-crossing are $x(179) = 105$ and $x(180) = -105$. These intensity values plus the location of the zero-crossing provide a complete description of the edge. Fig. 2.21 shows $-y''_\sigma(i)$ for $\sigma = 11$. In this case, the zero-crossing point is $i = 177.5$ which is clearly an inaccurate result.

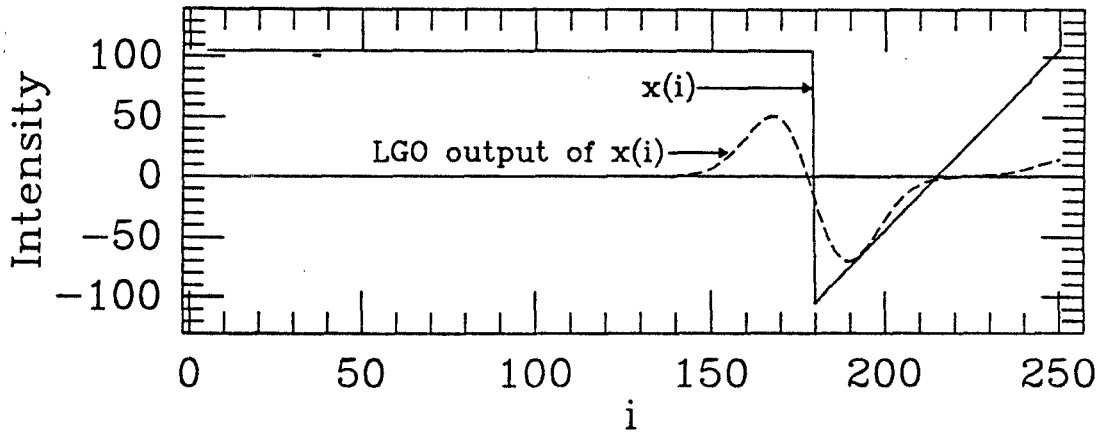


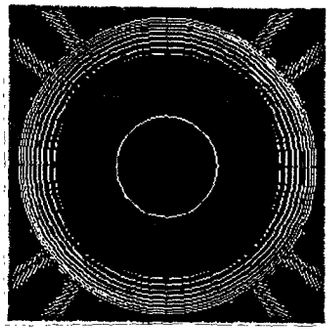
Figure 2.21: An example for the locational accuracy of LGO, $\sigma = 11$.

Thus, on the one hand, small values of σ are needed to ensure accurate detection of the edge location and, on the other hand, large values of σ should be used for detection of "strong" edges. These requirements are conflicting and constitute a drawback of the LGO edge-detection scheme. In addition, the LGO scheme implicitly assumes the edge is *one* sample wide. This is in contrast with real-world situations (especially for images) where edges are typically several pixels wide.

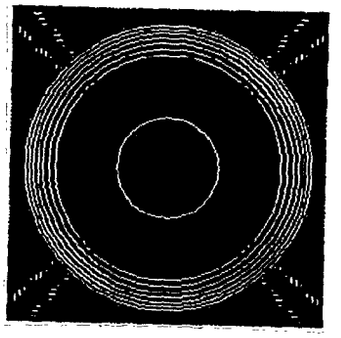
In what follows we proceed to use the LGO scheme to detect the strong edge of the test image presented in Fig. 2.10.

2.5.3 An Example of Locating Strong Edges with the LGO

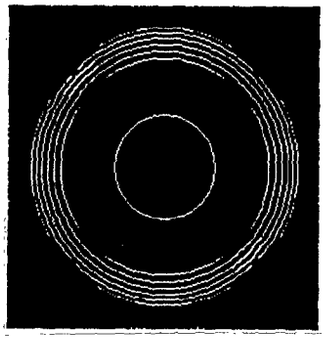
We now present the results of the LGO-based edge detection for the test image with five different values of σ namely, $1, 2, 3, 4$ and 5 pixel-widths. In each case, the threshold for the output of the gradient operator is determined empirically as the largest possible to keep the zero-crossings for the strong edge and to



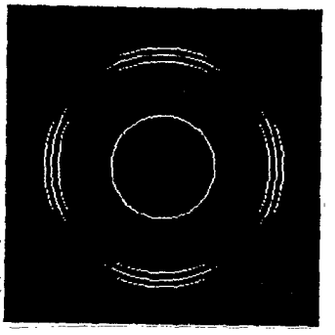
$\sigma = 1$ pixel-width (33 s)



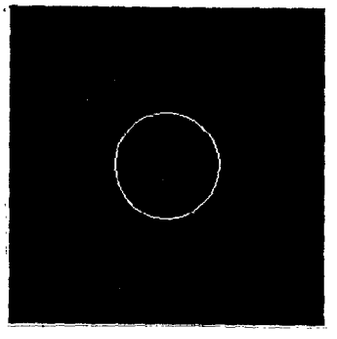
$\sigma = 2$ pixel-width (125 s)



$\sigma = 3$ pixel-width (345 s)



$\sigma = 4$ pixel-width (603 s)



$\sigma = 5$ pixel-width (1019 s)

Figure 2.22: Zero-crossing results with the LGO scheme for the test image.

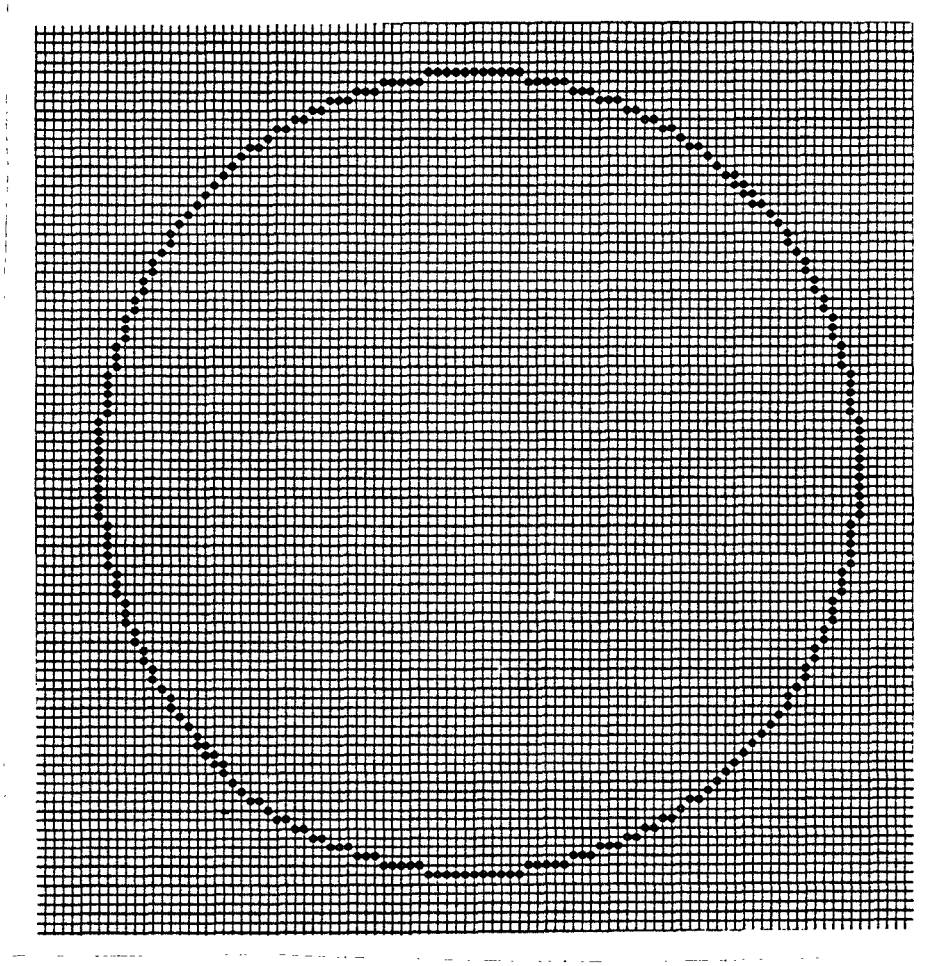


Figure 2.23: Enlarged picture of the zero-crossings for $\sigma = 5$ pixel-widths.

suppress, as much as possible, the zero-crossings corresponding to the textures. The results are summarized in Fig. 2.22 in the form of images where a bright point at pixel (i, j) indicates a zero-crossing at $(i, j+0.5)$ and/or $(i+0.5, j)$. The computer times, in seconds (s), used on a SPARCstation-1 to do the convolution (ref. (2.28)) are also included.

As shown in Fig. 2.22, the result for $\sigma = 5$ pixel-widths detects the strong edge only, while all the other results contain zero-crossings corresponding to the textures, as well. The enlarged picture of the zero-crossing points for $\sigma = 5$ pixel-widths is shown in Fig. 2.23 (compared with the actual edge location in

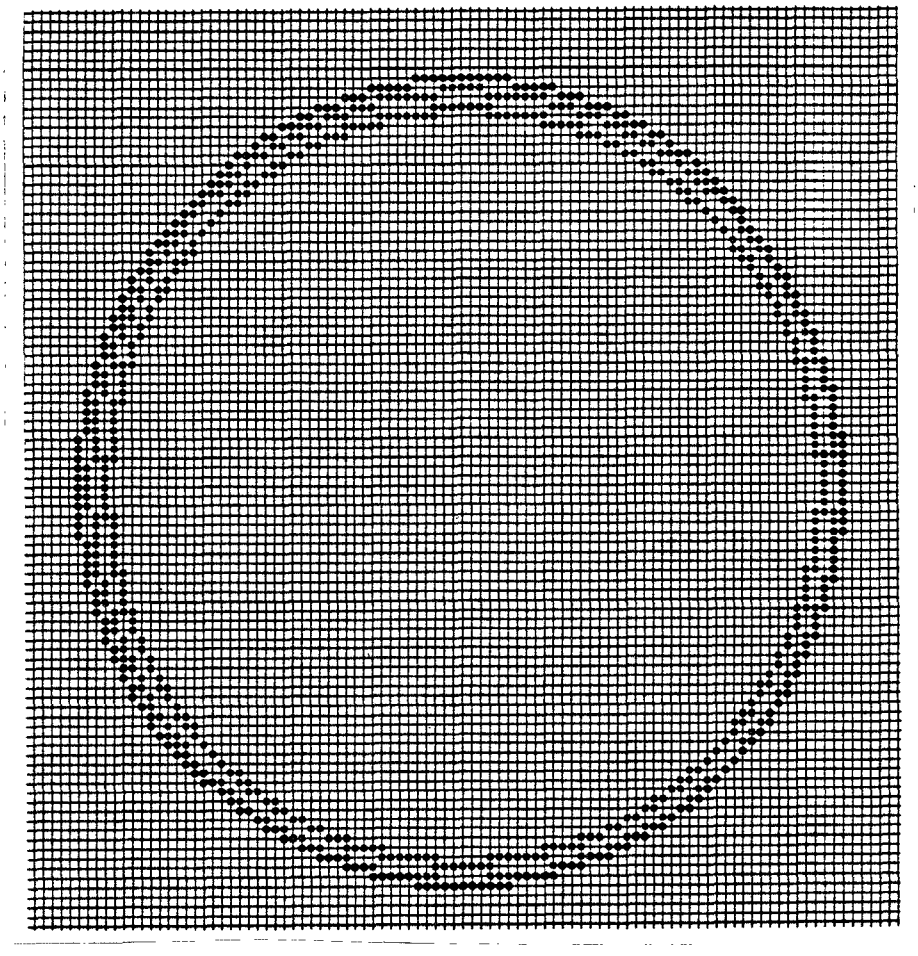


Figure 2.24: The superimposition of Fig. 2.23 and Fig. 2.12.

Fig. 2.24), where the positions of the zero-crossings are one half pixel-width to the right or below the square-dots (ref. Subsection 2.5.2). Thus, the set of the edge brim points consists of the pixels on two circles. One circle is shown in Fig. 2.23, and the other one (not shown) is one pixel away to the right and below of the first one. The primary image of these edge brim points is generated and shown in Fig. 2.25. Notice that simply there is no edge in Fig. 2.25, as compared with Fig. 2.18 obtained using the three-component image model.

Therefore, the representation of the edge in the test image generated with the LGO gives location error, and, more seriously, fails to provide the intensity

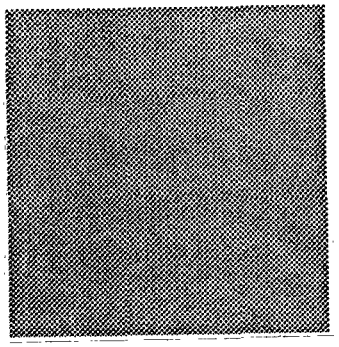


Figure 2.25: The primary image of the neighboring pixels of zero-crossings shown in Fig. 2.23.

variation of the edge, while the strong edge extraction scheme associated with the three-component image model provides a superior edge extraction performance.

2.6 Summary and Conclusions

We have studied several psychovisual evidences of the HVS and attempted to describe them mathematically. In particular, the strong edges and the areas of smooth intensity variations are carefully studied and their effect as well as their interaction in formation of the perception are described in terms of simple minimization problems. An algorithm, based on a space-variant low-pass filtering operation, for the generation of the stressed image is developed from which the strong edges can be easily extracted. This has led to introducing a three-component image model based on the (i) strong edges, (ii) areas of smooth intensity variations and (iii) textures - the three components which apparently play different roles in the formation of the perception.

In a specific comparison, we have shown the superiority of the strong edge extraction scheme developed here over the LGO-based edge extraction method.

The three-component model was motivated by the need for subjective-based criteria for the design of low bit-rate image coding systems (such as adaptive DCT and subband coding) in which the three components of the model are treated based on their relative perceptual importance. The design of such image coding systems is the topic of the next chapter.

Chapter 3

Low Bit-Rate Image Coding Using the Three-Component Image Model

3.1 Introduction

In this chapter,we develop image coding systems based on an amalgamation of feature-based and waveform coding techniques. Specifically, to take into account the properties of the HVS, the image signal is first decomposed into components of distinctive significance to the human perception, namely, the *strong edge*, *texture* and *smooth* components. This is accomplished by a scheme developed in the last chapter. The geometric and intensity information of the strong edges (also called the primary component) is encoded separately. The texture and smooth components are encoded using waveform coding methods. Both entropy-coded adaptive DCT and entropy-coded subband coding are considered. Within this

framework, we demonstrate the advantages of the proposed approach in terms of objective and subjective performance improvements. Furthermore, this study has enabled us to make meaningful comparisons between two rival schemes, namely, adaptive DCT and subband coding, under similar modeling assumptions. In addition, we develop a modification of our DCT-based scheme in which the well-known contrast sensitivity of the HVS [1], [34], [35], is used for perceptual tuning of the parameters of the coder. It is shown that the combination of this perceptual tuning and the three-component based approach further improves the subjective performance of the coder, especially at low bit rates.

The rest of the chapter is organized as follows. In Section 3.2, a brief description of the three-component image model along with some decomposition examples are provided. The encoding of the primary component is described in Section 3.3 followed by the description of two encoding schemes for the smooth and texture components in Section 3.4. Section 3.5 includes the description of the overall encoding schemes as well as extensive simulation results and comparisons. Section 3.6 is devoted to the contrast sensitivity of the HVS and how it is used for perceptual tuning of the encoding system. Finally, Section 3.7 contains a summary and conclusions.

3.2 Three-Component Image Model

In this section, we briefly review the three-component image model of the chapter 2, put it into the perspective of image coding, and provide several examples of the real-world images.

The block diagram of the system used for generating the three components,

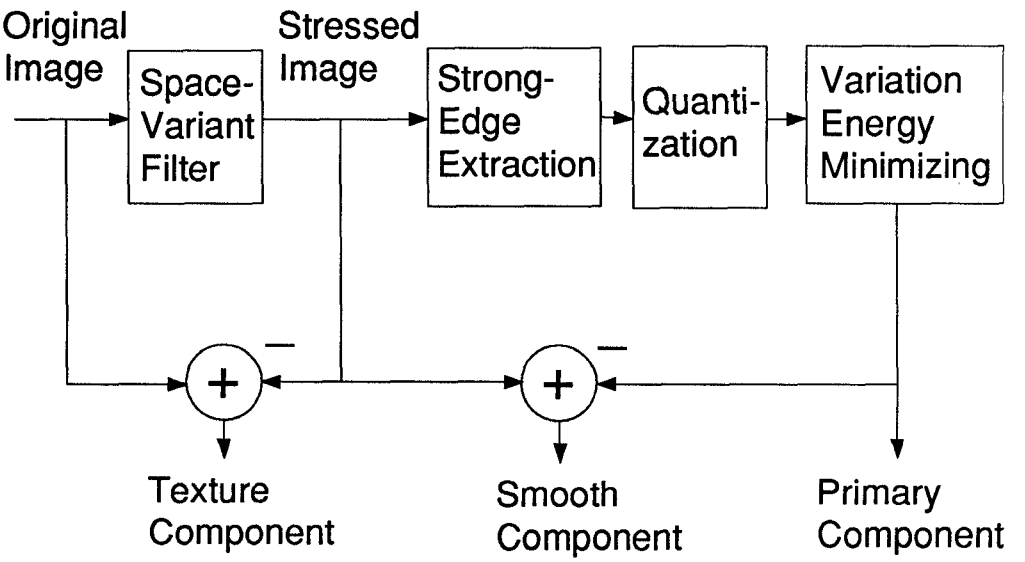


Figure 3.1: Generation of three-component image model.

namely, the *primary, smooth* and *texture components*, from the original image is illustrated in Fig. 3.1. The original image is first processed by a space-variant filter whose output is what we call the stressed image. The stressed image is essentially a low-pass version of the original image except near the "strong edges" where the filter acts as an all-pass filter to preserve the sharpness of these edges. The stressed image, therefore, contains the smooth intensity variations and the strong edge information of the original image (ref. Appendix C). The space-variant filter is implemented using an iterative procedure described by a flow chart in Fig. 3.2; $\{x_{i,j}\}$ is the original $M \times M$ image, ν is the iteration index and N_ν is the number of iterations before updating of the weighting factors.

The strong edge information is subsequently extracted from the stressed image by identifying pixels of local maximum curvature energy, and is used to generate the primary component of the original image (ref. Appendix B) follow-

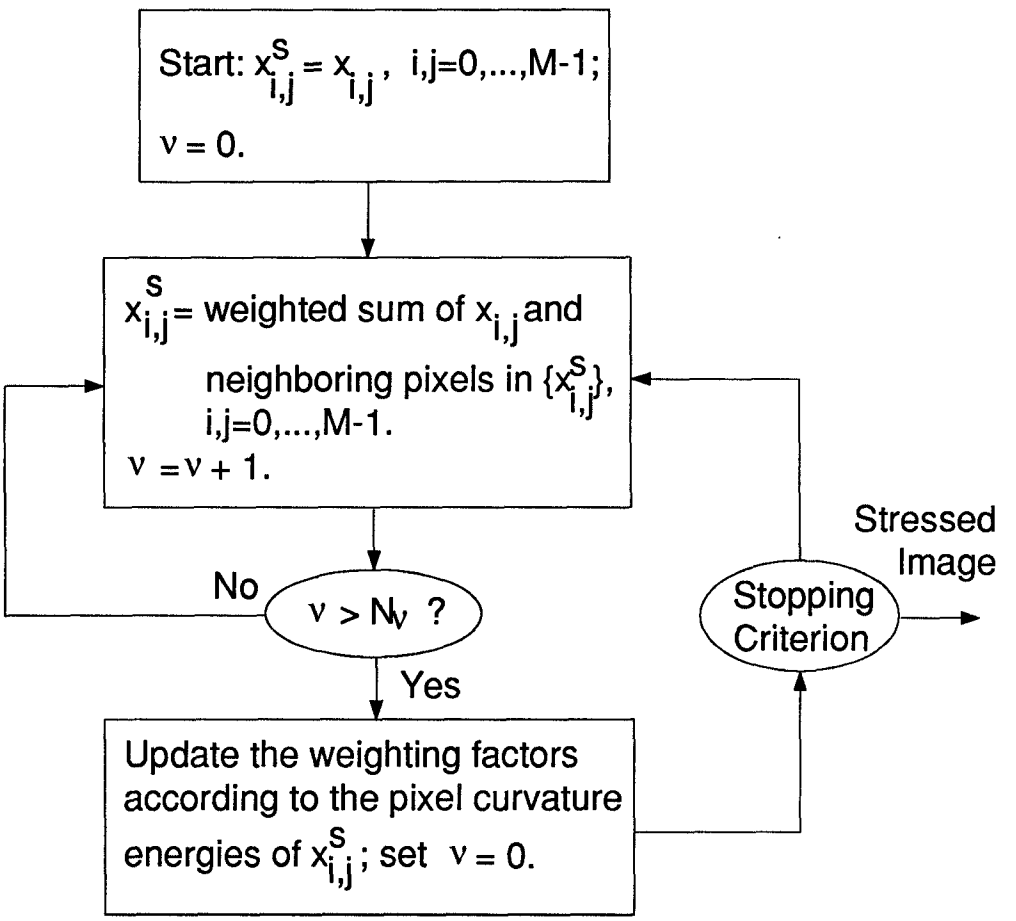


Figure 3.2: Flow chart describing the generation of the stressed image.

(a) (b)

Figure 3.3: (a) Lenna image, (b) the stressed image associated with A-configuration.

ing quantization of the intensity values (and possibly the geometric information) of the strong edge brim contours. The primary component is similar, in concept, to the sketch image of [22]. The quantization and encoding procedure of the strong edge information is described in the next section.

Two examples of the three-component image model, referred to as the A- and B-configuration, are provided in the following for the 512×512 Lenna. The parameters for generating these two examples are $N_\nu = 20$, $T = 64$, $T_c = 32$, $T_\ell = 8$ and $d = 3$ for the A-configuration and $N_\nu = 10$, $T = 64$, $T_c = 32$, $T_\ell = 4$ and $d = 3$ for the B-configuration. In Figs. 3.3 and 3.6,[1] the original image Lenna, the stressed images, the strong edge brim contours and the re-

[1]Since the smooth and texture components can have negative values, for display purposes, we have added a constant to these components to render all pixel values non-negative.

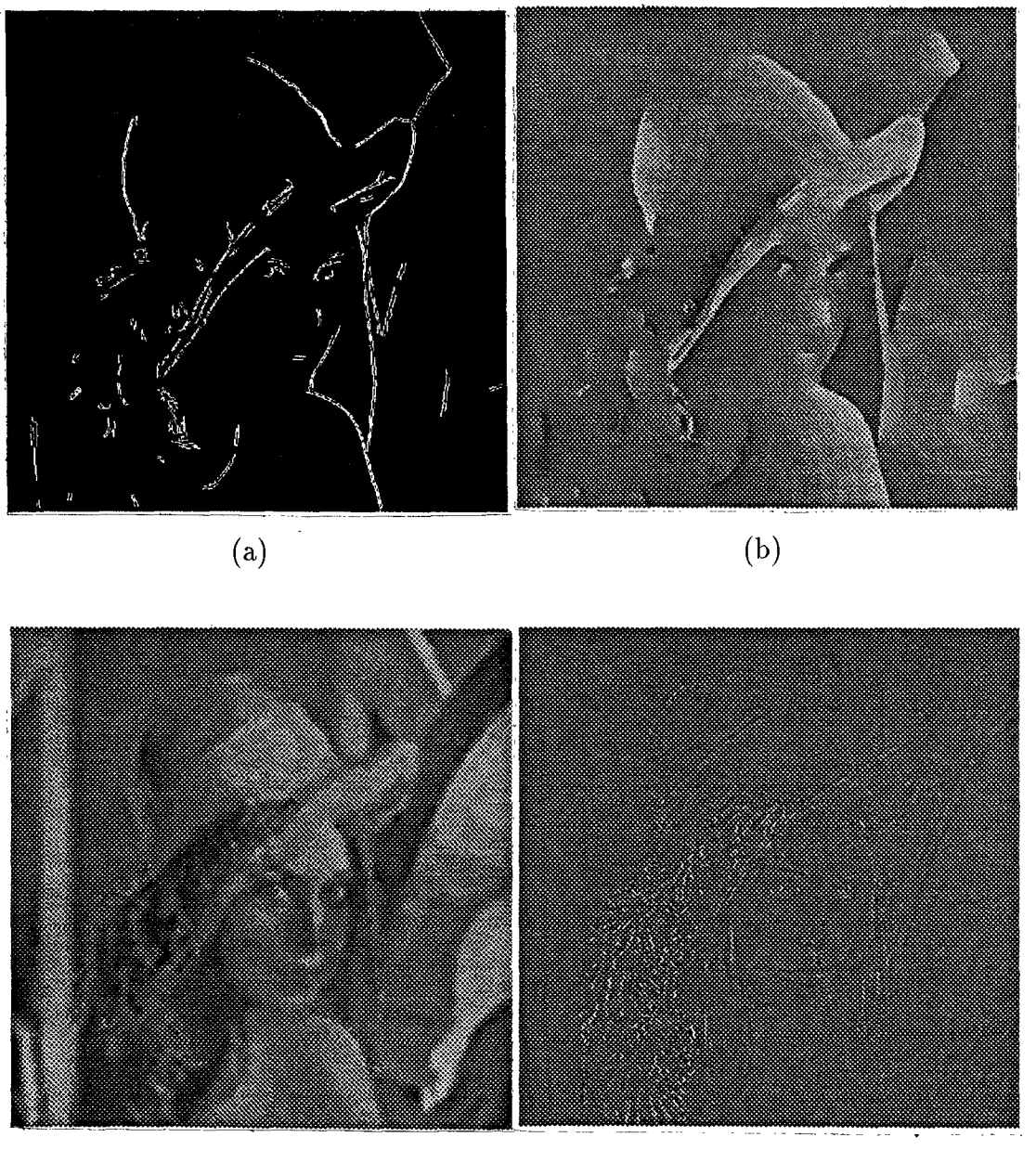


Figure 3.4: (a) the strong edge brim contours, (b) the primary component (1-ring coding), (c) the smooth component and (d) the texture component associated with A-configuration.

(a) (b)

Figure 3.5: (a) Lenna image, (b) the stressed image associated with B-configuration.

sulting three components for the two configurations are shown. In both cases, the primary component is obtained from the quantized strong edge information (see Section 3.3), and the corresponding quantization distortion is included in the smooth component, which can be discarded by smoothing the smooth components since the primary components provide good replicas of the corresponding strong edges. These distortions can be noticed near the strong edges in Figs. 3.4 (c) and 3.6 (c). Note that the A-configuration results in more strong edges than the B-configuration due to its larger N_ν. Generally, smaller values of N_ν lead to extracting a smaller number of strong edges, since, for smaller N_ν, the iterative solutions of the EMM problem have rather larger deviation from the solution of the problem; the pixel curvature energies of the iterative solutions are significant only at the locations of very strong edges. Then after the updating of the param-

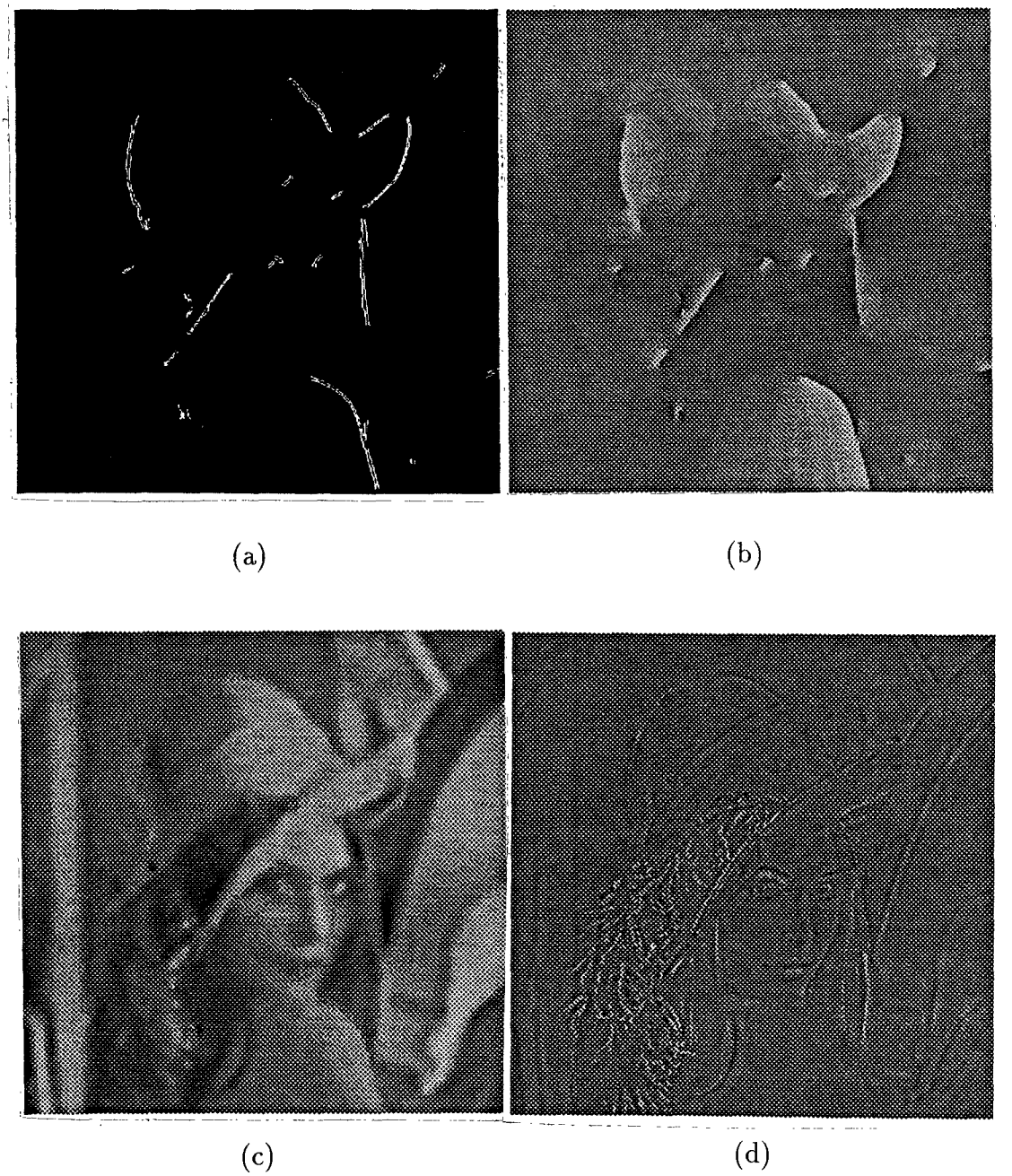


Figure 3.6: (a) the strong edge brim contours, (b) the primary component (1-ring coding), (c) the smooth component and (d) the texture component associated with B-configuration.

eter set, the rigidness of the bars will be reduced significantly only at these very strong edges; therefore the whole iteration procedure will be accentuated to a small number of very strong edges. Some other examples of the three-component model are provided in Appendix E.

3.3 Coding of Primary Component

The primary component of the image is specified by the collection of strong edge brim contours. These contours are defined by their pixel locations and intensity values. More specifically, a strong edge brim contour, denoted by $\overline{b}$, is defined as a sequence of triples:

$$\overline{b} \equiv \{(i^k, j^k, x_{i^k,j^k})\}_{k=0}^{m-1}, \tag{3.1}$$

where m is the length of the contour, (i^k, j^k) specifies the location of the kth pixel on the contour and x_{i^k,j^k} is the intensity value at (i^k, j^k); the consecutive pixels on the contour are connected and distinct, i.e., $|i^{k-1} - i^k| \leq 1$, $|j^{k-1} - j^k| \leq 1$ and $(i^{k-1}, j^{k-1}) \neq (i^k, j^k)$, $k = 1, \ldots, m-1$, and the variation of the intensity values on the contour is constrained by

$$\max_{k=0,1,\ldots,m-1} |x_{i^k,j^k} - \frac{1}{m} \sum_{\ell=0}^{m-1} x_{i^\ell,j^\ell}| \leq T_c, \tag{3.2}$$

where $T_c > 0$.

Our objective is to code the contours (location and intensity) with little or no perceptual distortion in reconstructing the primary component. To this end, the sequence of contour pixel locations, $\overline{c} \equiv \{(i^0, j^0), (i^1, j^1), \ldots, (i^{m-1}, j^{m-1})\}$, is encoded using chain coding [29], [31], [36]. When the well-known Freeman code [31] (also known as the 1-ring code [36]) is used, the sequence is losslessly

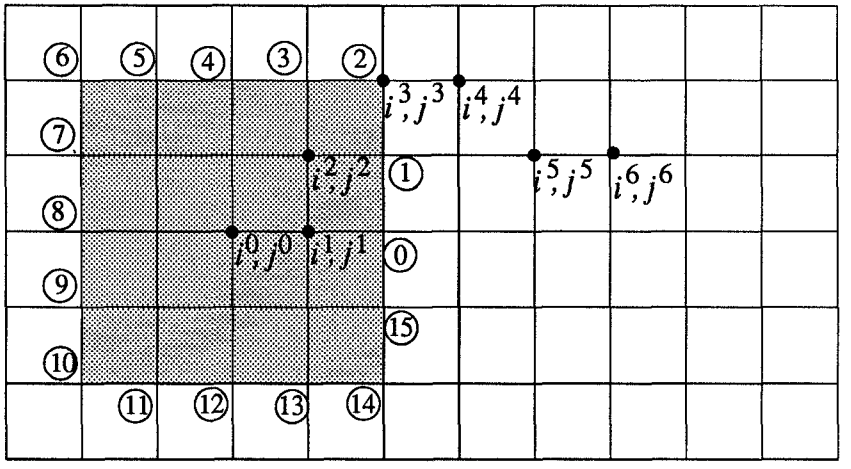


Figure 3.7: Example of 2-ring coding.

encoded. To achieve lower bit rates, the N-ring code with $N > 1$ can be used. This procedure results in lossy coding of the contour locations. An example of the 2-ring code is given below.

Consider the sequence $\bar{c}$ of length $m = 7$ shown in Fig. 3.7, where the pixel locations are marked with solid dots. The 2-ring coding starts at (i^0, j^0). The first intersection of the 2-ring centered at (i^0, j^0) (boundary of the shaded region) with $\bar{c}$ is identified. The intersection point, (i^3, j^3), is encoded with the corresponding index s^1 on the ring (in this case $s^1 = 2$, as indicated in Fig. 3.7) and the center of the 2-ring is moved to (i^3, j^3). Following the same procedure, a new intersection is identified at (i^5, j^5) which is encoded with index $s^2 = 15$. Then the 2-ring is shifted to pixel (i^5, j^5), but no intersection point is identified; this terminates the encoding operation. The resulting encoded message is $\{(i^0, j^0), 2, 15\}$. The decoding is straightforward. Those points on $\bar{c}$ that correspond to intersection points are correctly reconstructed; the inter-

mediate points are reconstructed by interpolation. For this example, the reconstructed sequence can be either $\{(i^0,j^0),(i^2,j^2),(i^3,j^3),(i^4,j^4),(i^5,j^5)\}$ or $\{(i^0,j^0),(i^2,j^2),(i^3,j^3),(i^4+1,j^4),(i^5,j^5)\}$. One of the two possibilities is selected according to a prescribed decoding rule.

Let $\{s^1,s^2,s^3,\ldots\}$ denote the sequence of indices on the N-ring obtained in successive steps of the chain coding operation. Because the contours are generally smooth, it is more efficient to code the differences between the neighboring indices than to code them directly. Therefore, upon defining $r^i = (s^i - s^{i-1}) \bmod 8N$, $i = 2,3,\ldots$, a code sequence $\{(i^0,j^0),s^1,r^2,r^3,\ldots\}$, is generated where (i^0,j^0) is called the starting location, s^1 the first direction, and $r^2,r^3,\ldots$, the relative directions.

To encode the contour intensity values, sequence $\{x_{i^0,j^0},x_{i^1,j^1},\ldots,x_{i^{m-1},j^{m-1}}\}$ is quantized to a constant sequence $\{\overline{x},\overline{x},\ldots,\overline{x}\}$, where $\overline{x} = [m^{-1}\sum_{\ell=0}^{m-1} x_{i^\ell,j^\ell}]$ and $[x]$ denotes the integer closest to x. Simulation results show that this quantization gives rise to little perceptual degradation on edges (for the choices of T_c considered in this work) as compared with the case where the intensity values are losslessly coded.

The sequence $\{\overline{x},m,(i^0,j^0),s^1,r^2,r^3,\ldots\}$ is used to represent each strong edge brim contour. While $\overline{x},m$ and (i^0,j^0) are encoded by a fixed number of bits, s^1 and $\{r^i\}$ are encoded separately by means of arithmetic coding [37], [38].

The average rate, denoted by r_p bits/pixel, for encoding the primary component is tabulated in Table 3.1 for the 512×512 Lenna using the two configurations considered in Section 3.2. The corresponding primary components are shown in Figs. 3.4 (b) and 3.6 (b) for the 1-ring code; the results for 2- and 3-ring codes are perceptually indistinguishable and are therefore omitted.

Configuration	1-ring	2-ring	3-ring
A	0.048	0.038	0.032
B	0.017	0.013	0.011

Table 3.1: Bit rate (bits/pixel) for encoding the primary component for two different parameter sets.

Remark

It is worth noting that the constraint (3.2) may be changed to a more general one:

$$|x_{i^k,j^k} - \frac{1}{n}\sum_{\ell=0}^{n-1} x_{i^k-\ell,j^k-\ell}| \leq T_c', \tag{3.3}$$

for $k = 0, 1, \ldots, m-1$, and $T_c' > 0$, where $n = \min\{k+1, n_1\}$ and $n_1 \geq 1$ is a finite window size. With this new constraint, some of the contours obtained from (3.2) will be connected and thus the coding rate may be decreased due to the reduced number of starting locations to be encoded. However, we need to specify more than one intensity value for some contours with this new constraint. The trade-off between the above two factors is controlled by the threshold T_c' and the window size n_1. After extensive studies of these trade-offs, we have concluded that the reduction in the bit rate is negligibly small. Hence, the simpler constraint (3.2) is used for the rest of the work.

3.4 Coding of Smooth and Texture Components

In this section, we consider the coding of the smooth and texture components. The basic approach is to encode the *sum* of these components using waveform coding techniques. Two different entropy-coded image coding techniques, adaptive (2-D) DCT coding and 2-D subband coding, are used in this work.

3.4.1 Adaptive DCT Coding

Let the smooth and texture components of the original image $X = \{x_{i,j}\}$ be denoted by $S = \{s_{i,j}\}$ and $T = \{t_{i,j}\}$, respectively, where $i, j = 0, 1, \ldots, M-1$. Let S and T be segmented into $L \times L$ blocks $\mathbf{s}_{m,n}$ and $\mathbf{t}_{m,n}$, $m, n = 0, 1, \ldots, (M/L)-1$, respectively, and denote the block of 2-D DCT coefficients of $\mathbf{s}_{m,n}$ and $\mathbf{t}_{m,n}$ by $\overline{\mathbf{s}}_{m,n}$ and $\overline{\mathbf{t}}_{m,n}$, respectively.

The structure of the adaptive DCT coding scheme used here is similar to that in [5]. After computing the 2-D DCT of $\mathbf{u}_{m,n} \equiv (\mathbf{s}_{m,n} + \mathbf{t}_{m,n})$, the blocks $\overline{\mathbf{u}}_{m,n} \equiv (\overline{\mathbf{s}}_{m,n} + \overline{\mathbf{t}}_{m,n})$ are classified into one of a finite number of classes. In [5], the classification is done based on the ac energies of the blocks. As indicated in [39], the classification scheme in [5] neglects the frequency distribution of the ac energy which should be considered for a more efficient coding. In our case, due to the three component model, the low and high frequency energies of a block are already separated and are contained in $\overline{\mathbf{s}}_{m,n}$ and $\overline{\mathbf{t}}_{m,n}$, respectively. Thus, a more efficient classification is possible by using the ac energies of $\overline{\mathbf{s}}_{m,n}$ and $\overline{\mathbf{t}}_{m,n}$. The classification procedure used here consists of two stages. In the first stage, the transform blocks are classified into one of K_1 classes by comparing

the ac energies of $\bar{s}_{m,n}$ against T_k^1, $k = 0, 1, \ldots, K_1 - 2$; the thresholds $\{T_k^1\}$ are chosen such that each of the resulting K_1 classes contains the same number of blocks. In the second stage, each resulting class of the first stage, say k, is further divided into K_2 classes by comparing the ac energies of $\bar{t}_{m,n}$ against another set of thresholds, $T_{k,\ell}^2$, $\ell = 0, 1, \ldots, K_2 - 2$. These thresholds are also chosen such that the resulting $K = K_1 K_2$ classes have the same number of blocks in them.[2]

We have conducted a study, similar to that of [40], to determine the approximate distribution of the 2-D DCT coefficients of the different classes by comparing their empirical distributions against the so-called generalized Gaussian distribution [41] using the Kolmogorov-Smirnov test [40]. This is done using a database consisting of a large number of different images. Based on this study, we have concluded that the $(0,0)$th, $(0,1)$th and $(1,0)$th coefficients have an almost Gaussian distribution; all other coefficients (except those that have very small variances and hence are subsequently assigned very small bit rates) are best approximated by a Laplacian distribution. This observation holds for the coefficients in all classes in the case where $K = 4$. From now on, we will use these assumptions for the design of the overall system.

The 2-D DCT coefficients are quantized with uniform-threshold quantizers (UTQs) [41] and subsequently encoded using Huffman codes (HCs). This leads to performance close to that of optimal entropy-constrained scalar quantization [11]. We denote the variance-normalized mean squared error (MSE) associated with the UTQ-HC pair operating at rate r bits/sample for the Gaussian and

[2]The constraint that each class should contain the same number of blocks is not a requirement; it just makes the system simpler to implement. Better performance results can be expected if this constraint is removed.

Laplacian distributions by $d_G(r)$ and $d_L(r)$, respectively. If the variance and the coding rate associated with the (u,v)th coefficient in the kth class are denoted, respectively, by $\sigma_k^2(u,v)$ and $r_{u,v}^k$, the overall MSE is given by [3]

$$D = \frac{1}{L^2}\{\sigma^2(0,0)d_G(r_{0,0}) + \frac{1}{K}\sum_{k=0}^{K-1}[\sigma_k^2(0,1)d_G(r_{0,1}^k) + \sigma_k^2(1,0)d_G(r_{1,0}^k) + \sum_{(u,v)\neq(0,0),(0,1),(1,0)} \sigma_k^2(u,v)d_L(r_{u,v}^k)]\}, \quad (3.4)$$

where $\sigma^2(0,0)$ and $r_{0,0}$ are the variance and the coding rate, respectively, for the dc coefficient which is encoded in the same manner regardless of the class it belongs to. The average bit rate for encoding the smooth and texture components, r_{st}, is given by

$$r_{st} = r_o + \frac{1}{L^2}\{r_{0,0} + \frac{1}{K}\sum_{k=0}^{K-1}\sum_{u,v\neq(0,0)} r_{u,v}^k\}, \text{ bits/pixel}, \quad (3.5)$$

where r_o is the bit rate for coding the overhead information. We will elaborate on r_o later.

At this point, it remains to determine the optimal allocation of bit rates among the 2-D DCT coefficients of the different classes. This bit allocation which is the solution of the following constrained minimization problem,

$$\min_{\{r_o, r_{0,0}, r_{u,v}^k\}:\, r_{st}\leq r_d} D, \quad (3.6)$$

where r_d is the design average bit rate, can be obtained efficiently using the steepest descent algorithm described in [42]. To reduce the overhead information for the bit allocation maps and to have a simple system, we have limited $r_{0,0}$ and $\{r_{u,v}^k\}$ to be an integer multiple of 0.1 bits; they are also constrained to be

[3]Throughout this paper, the 2-D DCT used is as defined in [16]. This transformation is unitary and hence the MSE in the spatial domain is the same as that in the transform domain.

less than 5.0 bits for the Laplacian distribution and 8.0 bits for the Gaussian distribution.

To reconstruct the 2-D DCT coefficients at the receiver side, estimates of $\{\sigma_k^2(u,v)\}$ are required ($\sigma^2(0,0)$ is transmitted directly). If the variances of quantization errors of the DCT coefficients, denoted by $\epsilon_k^2(u,v)$, are known, then $\{\sigma_k^2(u,v)\}$ can be estimated from the allocated bit rates $\{r_{u,v}^k\}$. This is because, for $(u,v) \neq (0,0)$, we have

$$\epsilon_k^2(u,v) = \begin{cases} \sigma_k^2(u,v) d_G(r_{u,v}^k), & (u,v) = (0,1) \text{ or } (1,0), \\ \sigma_k^2(u,v) d_L(r_{u,v}^k), & \text{otherwise.} \end{cases} \tag{3.7}$$

As mentioned before, the performance of the UTQ-HC pairs is very close to that of optimal entropy-constrained scalar quantizers. These quantizers, in turn, have a performance which for high bit rates is about 0.255 bits/sample above the rate-distortion function [41], [43]. Approximating the rate-distortion function by the Shannon Lower Bound [44] and using this observation, we have

$$d_G(r) \approx \frac{\pi e}{6} 2^{-2r}, \qquad d_L(r) \approx \frac{e^2}{6} 2^{-2r}. \tag{3.8}$$

Substituting (3.8) into (3.4) to solve (3.6), one can easily show that when the bit rates are optimally chosen, the resulting quantization error of the DCT coefficients have the same variances. The details are included in Appendix F. In the ideal case if the distribution of the coefficients is exactly what we have assumed, the knowledge of this *common* quantization error variance, called the normalization factor and denoted by c, at the receiver, can be used to compute the variances $\{\sigma_k^2(u,v)\}$ using the bit maps (see eq. (3.7)). In the system actually implemented, the average of those $\epsilon_k^2(u,v)$ corresponding to coefficients with $r_{u,v}^k > 0$, $(u,v) \neq (0,0)$ is used as the value of c.

Before closing this subsection, we should mention that the overhead information that needs to be sent consists of the classification information ($\log_2 K$ bits per block), the mean and variance of the dc coefficient (32 bits for each), the normalization factor c (32 bits) and the bit maps [4] ($2K\lceil\log_2 L\rceil + \lceil\log_2 B\rceil(1 + \sum_{k=0}^{K-1}(u_k v_k - 1))$). For an $M \times M$ image, this amounts to $r_o = (\log_2 K)/L^2 + (96 + 2K\lceil\log_2 L\rceil + \lceil\log_2 B\rceil(1 + \sum_{k=0}^{K-1}(u_k v_k - 1)))/M^2$, bits/pixel. For example, for encoding the 512×512 Lenna at 0.5 bits/pixel with $L = 16$, $r_o = 0.017$ bits/pixel, $r_p = 0.048$ bits/pixel with 1-ring coding and the A-configuration leaving 0.435 bits/pixel for r_{st}.

3.4.2 Subband Coding

The entropy-coded 2-D subband coding (SBC) system used here for encoding the sum of the smooth and texture components (hereafter called the smooth-texture component) is identical to that of [11]. The smooth-texture component is analyzed into several narrow-band (or subband) images which are then encoded separately. In the decoder, a replica of the smooth-texture component is synthesized from the decoded subband images. The analysis and synthesis are performed using 2-D separable quadrature mirror filter (QMF) banks. The input image is split into 16 subbands. The reader is referred to [8] for details. All subbands except the lowest frequency subband (LFS) are encoded by means of

[4] The integers u_k and v_k are defined to be the smallest integers such that $u \geq u_k$ or $v \geq v_k$ implies $r^k_{u,v} = 0$. Thus, for the kth class, we need to transmit $2\lceil\log_2 L\rceil$ bits for u_k and v_k, and $\lceil\log_2 B\rceil(u_k v_k - 1)$ bits to specify $r^k_{u,v}$, $u < u_k$, $v < v_k$, $(u, v) \neq (0, 0)$, where $\lceil x\rceil$ denotes the smallest integer larger than x and B is the number of possible different values of $r^k_{u,v}$. Since $r_{0,0}$ is the same for all classes, the overall number of bits for the bitmaps is $2K\lceil\log_2 L\rceil + \lceil\log_2 B\rceil(1 + \sum_{k=0}^{K-1}(u_k v_k - 1))$.

appropriately designed UTQ-HC pairs; the LFS is encoded by means of a non-adaptive 2-D DCT encoding system in which the DCT coefficients are encoded also by UTQ-HC pairs. The overall system design, the bit allocation procedure and the performance results are reported in [11].

Perhaps we should mention that in the context of SBC, it suffices to have a 2-component model which separates the primary component from the smooth-texture component. This is because in a SBC system, the input image is separated into subband images based on their frequency contents and therefore the system has the ability to extract the smooth and texture components from the smooth-texture input.

3.5 Simulation Results and Comparisons

In this section we describe the overall structure of the different image coding systems developed based on the three-component model. We also present simulation results and perform appropriate comparisons. The two main encoding schemes are described next.

3.5.1 Description of systems

The ADCT-based scheme

The block diagram of the ADCT-based scheme is illustrated in Fig. 3.8. The primary component is encoded using the contour coding scheme of Section 3.3. The sum of the smooth and texture components is encoded using the ADCT coding scheme of Subsection IV.A. We will refer to this scheme as the 3C-ADCT-HC scheme (3C for the three-component model and HC for Huffman coding of

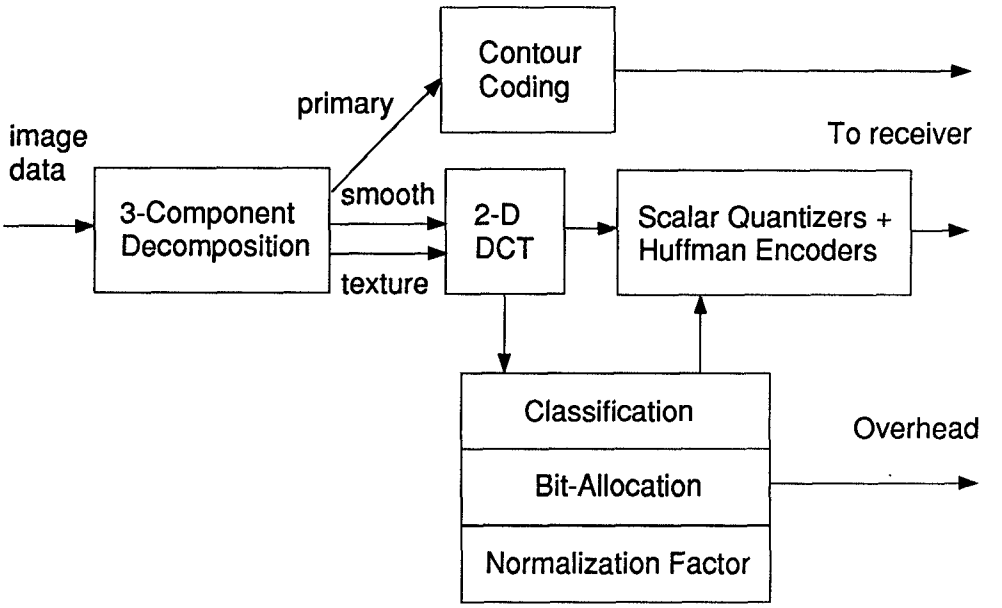


Figure 3.8: Block diagram of the ADCT-based image coding scheme using the three-component model.

the quantized DCT coefficients).

The SBC-based scheme

The block diagram of the SBC-based scheme is illustrated in Fig. 3.9. The only difference with 3C-ADCT-HC is in the coding of the smooth and texture components which in this case are encoded using the SBC coding scheme of Subsection IV.B. This scheme will be denoted by 3C-SBC-HC.

Other related schemes

For comparison purposes, we will also study the performance of an ADCT-based scheme like that of Subsection IV.A which operates on the original image directly and hence does not utilize the three-component model; in this scheme

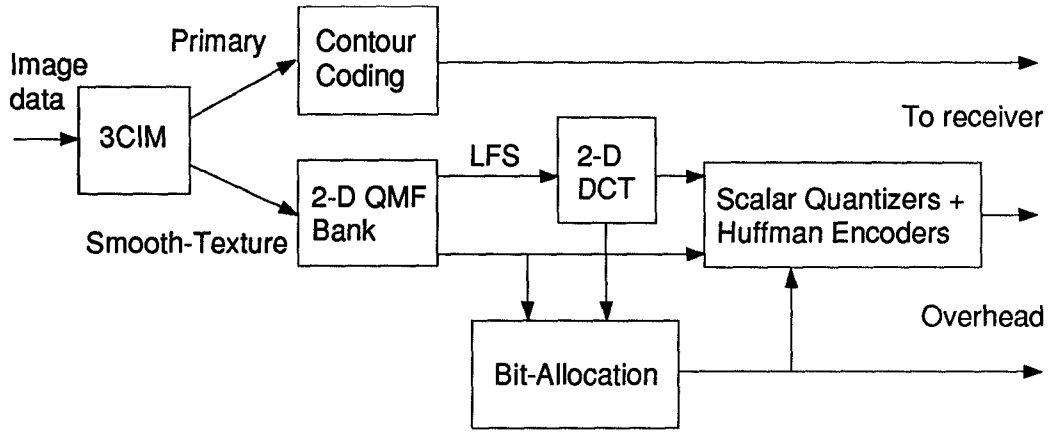


Figure 3.9: Block diagram of the SBC-based image coding scheme using the three-component model.

the classification method is exactly like that of [5]. This scheme, referred to as 1C-ADCT-HC, is essentially [5] an entropy-coded version of the system in [5]. Also, we consider the SBC-based coding scheme of [11] (called System B in [11]) directly applied to the image (thus ignoring the three-component model) and refer to it as 1C-SBC-HC. Finally, we refer to the Huffman coded version of JPEG as JPEG-HC. All JPEG simulation results are obtained using a software package described in [45].

[5]The other differences between 1C-ADCT-HC and the scheme in [5] are the bit assignment algorithm, the assumption on the distribution of the transform coefficients, and the estimation of $\{\sigma_k^2(u,v)\}$ at the receiver side.

3.5.2 Simulation results

We have simulated the performance of the above mentioned schemes on the 512×512 Lenna. The reconstructed images at different bit rates are illustrated in Figs. 3.10-3.15. For these simulations, in the 1C-ADCT-HC and 3C-ADCT-HC schemes the blocksize used is 16×16; JPEG-HC uses an 8×8 blocksize. In 1C-SBC-HC and 3C-SBC-HC, the lowest frequency subband is encoded using a nonadaptive DCT encoder with blocksize 4×4 as in [11]. In the 3C-ADCT-HC schemes, $K_1 = K_2 = 2$; in the 1C-ADCT-HC schemes, $K = 4$. In these figures, for all three-component based systems, the A-configuration described in Section 3.2 is used as the choice of parameters; 1-ring code is used, but 2- and 3-ring codes are also tried. In some cases, 2- or 3- ring codes yielded slightly better PSNRs. However, in no case did we observe a noticeable perceptual difference.

The peak signal-to-noise-ratio (PSNR) associated with these schemes as well as other schemes to be discussed shortly are summarized in Table 3.2. In this table, the PSNR results associated with the three-component model are the best ones obtained among the choices of A- and B- configurations and the 1-, 2- and 3-ring codes.

3.5.3 Discussion

The simulation results of Figs. 3.10-3.15 and their corresponding PSNRs tabulated in Table 3.2 deserve some discussion. Perhaps the most important observation to be made is that the 3C-ADCT-HC and 1C-ADCT-HC schemes provide the best results. The performance difference between these schemes and JPEG-HC is quite dramatic at low bit rates.

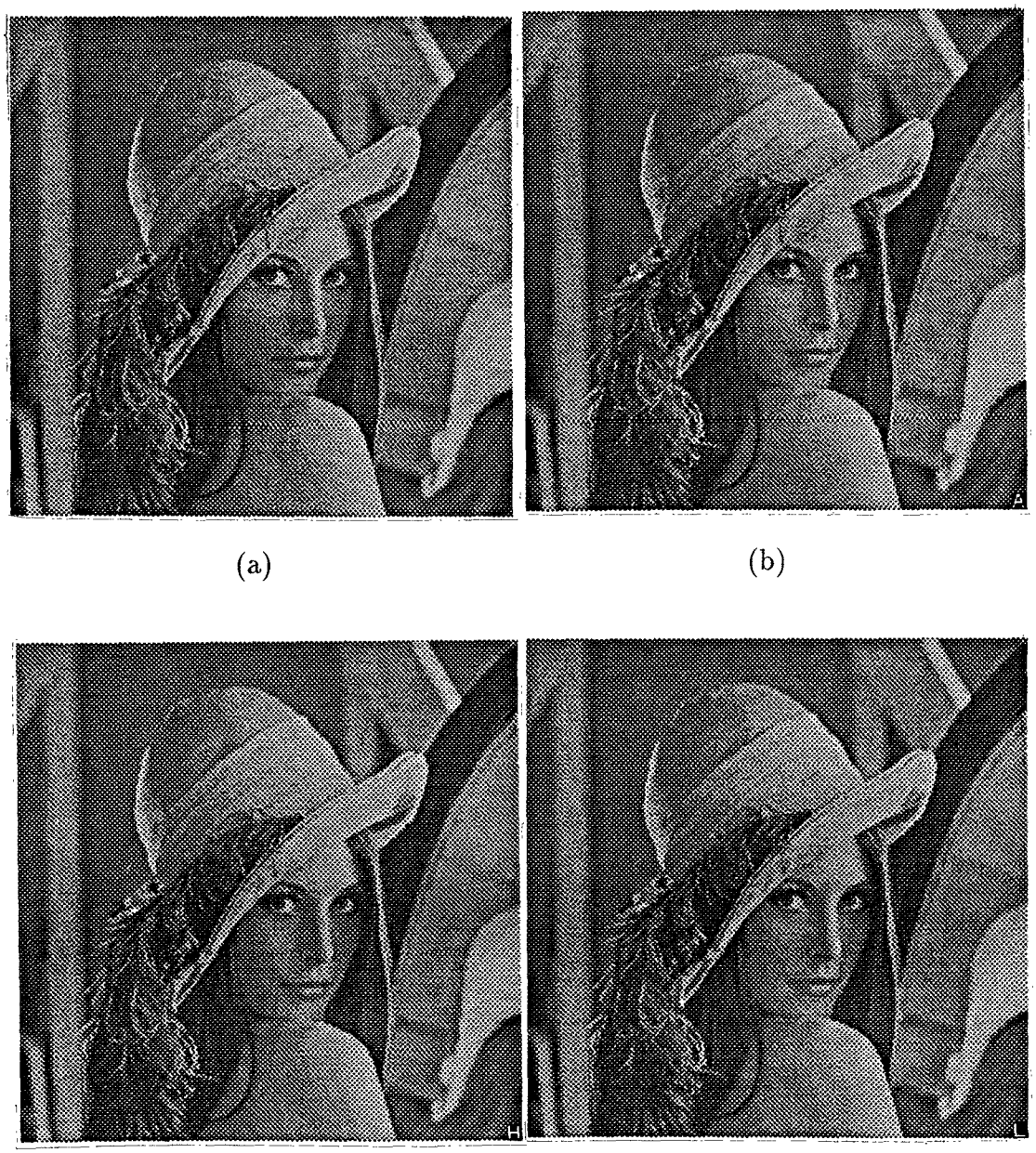


Figure 3.10: (a) Original, (b) JPEG-HC, (c) 1C-ADCT-HC, (d) 3C-ADCT-HC; design bit rate 0.5 bpp; actual bit rates in Table 2.

(a) (b)

(c) (d)

Figure 3.11: (a) Original, (b) JPEG-HC, (c) 1C-ADCT-HC, (d) 3C-ADCT-HC; design bit rate 0.25 bpp; actual bit rates in Table 2.

(a) (b)

(c) (d)

Figure 3.12: (a) Original, (b) JPEG-HC, (c) 1C-ADCT-HC, (d) 3C-ADCT-HC; design bit rate 0.125 bpp; actual bit rates in Table 2.

(a) (b)

Figure 3.13: (a) 1C-SBC-HC, (b) 3C-SBC-HC; design bit rate 0.5 bpp; actual bit rates in Table 2.

(a) (b)

Figure 3.14: (a) 1C-SBC-HC, (b) 3C-SBC-HC; design bit rate 0.25 bpp; actual bit rates in Table 2.

(a) (b)

Figure 3.15: (a) 1C-SBC-HC, (b) 3C-SBC-HC; design bit rate 0.125 bpp; actual bit rates in Table 2.

In comparing the subjective performances of 1C-ADCT-HC and 3C-ADCT-HC, it becomes evident that 3C-ADCT-HC performs better near the strong edges; the difference is quite visible at low bit rates. An additional very interesting observation is that even though 3C-ADCT-HC is designed to provide good *perceptual quality* and not to maximize PSNR, it, in fact, results in a PSNR larger than that of 1C-ADCT-HC – a system designed to maximize the PSNR. We will provide some explanation for this behavior later.

The important note to be made in comparing 1C-SBC-HC against 3C-SBC-HC is the significant reduction of "ringing" near the strong edges in 3C-SBC-HC; these ringing effects, which are well-known in subband coding systems, are quite visible in the 1C-SBC-HC scheme especially at low bit rates. While this superior perceptual performance is quite noticeable, the PSNR improvement of 3C-SBC-

Encoding Scheme	Design Bit Rate (bpp)			
	0.125	0.25	0.5	0.75
JPEG-HC	24.91 (0.148)	31.25 (0.283)	34.69 (0.513)	36.43 (0.748)
1C-ADCT-HC	30.10 (0.126)	32.92 (0.247)	35.91 (0.485)	37.91 (0.747)
3C-ADCT-HC	30.20 (0.126)	33.29 (0.250)	36.25 (0.492)	38.03 (0.747)
1C-SBC-HC	29.77 (0.125)	32.55 (0.250)	35.67 (0.500)	37.73 (0.748)
3C-SBC-HC	29.67 (0.125)	32.67 (0.251)	35.73 (0.494)	37.63 (0.750)
JPEG-AC	28.45 (0.128)	32.08 (0.265)	34.95 (0.491)	36.72 (0.755)
1C-ADCT-AC	30.10 (0.123)	32.92 (0.241)	35.91 (0.471)	37.91 (0.732)
3C-ADCT-AC	30.20 (0.123)	33.29 (0.243)	36.25 (0.479)	38.03 (0.731)

Table 3.2: PSNR (in dB) performance of various encoding schemes for 512×512 Lenna at different design bit rates. Numbers in parantheses are actual bit rates.

HC over 1C-SBC-HC is almost negligible.

In spite of the fact that 3C-ADCT-HC and JPEG-HC schemes are both DCT-based systems, the former system exhibits a significantly better performance than the latter. This improvement can be attributed to a combination of the following factors: (i) the method of block classification and DCT coefficient quantization used in 3C-ADCT-HC, (ii) the separate encoding of the primary component in 3C-ADCT-HC and (iii) the larger blocksize used in 3C-ADCT-HC.

To study the importance of these factors separately, we have provided, in Fig. 3.16, a plot of PSNR vs. bit rate for JPEG-HC with blocksize 8×8 as well as 1C-ADCT-HC and 3C-ADCT-HC with blocksizes 8×8 and 16×16. The

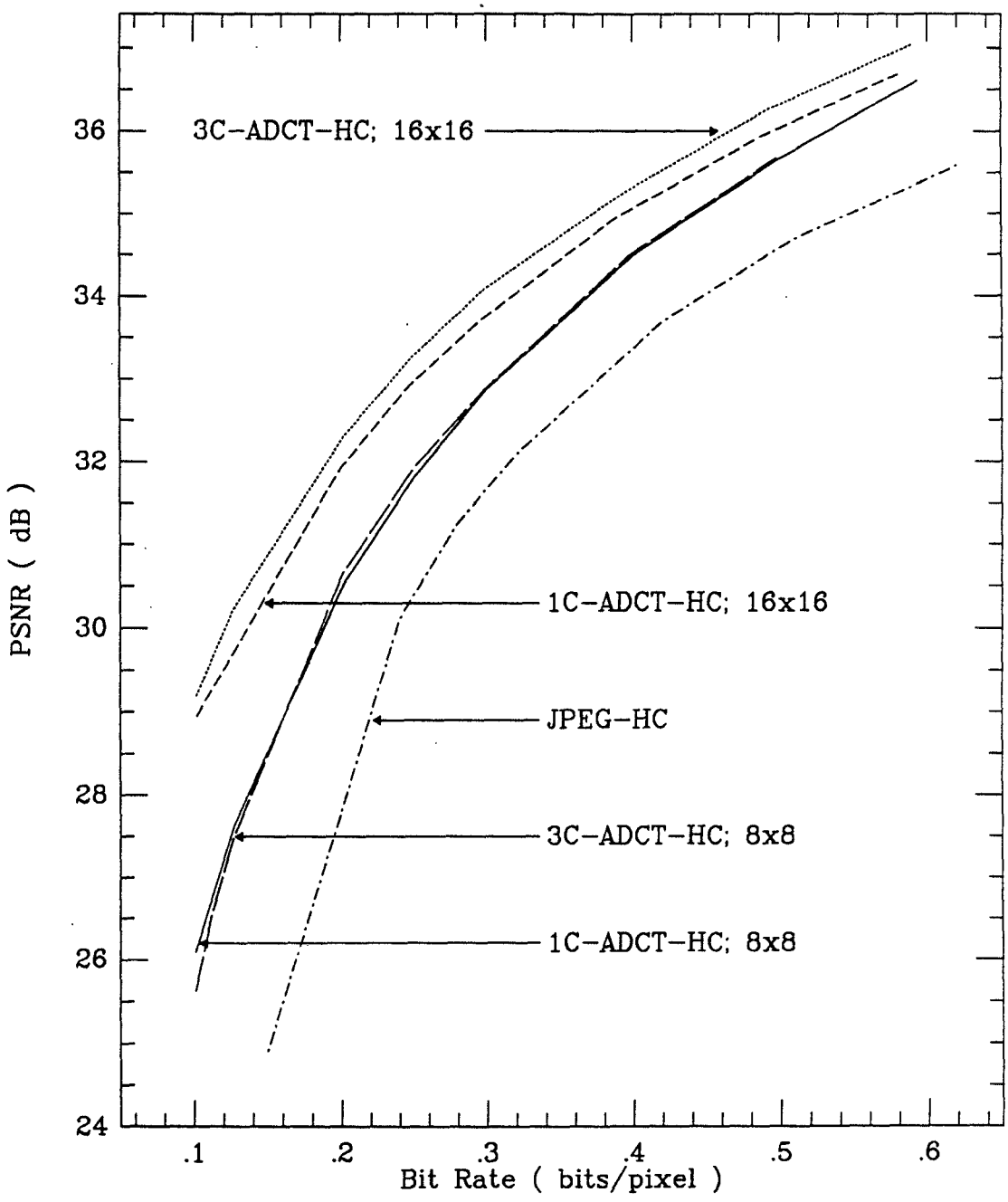


Figure 3.16: PSNR (in dB) vs. bit rate for JPEG-HC with blocksize 8×8 as well as 1C-ADCT-HC and 3C-ADCT-HC with blocksizes 8×8 and 16×16.

important conclusions are as follows. The significance of the larger blocksize is noticeable at low bit rates; at higher bit rates the improvement due to the larger blocksize is vanishingly small. The specific method of block classification and DCT coefficient encoding used in 3C-ADCT-HC is responsible for a significant portion of the PSNR improvement. Additional studies have revealed that the PSNR performance improvement of 3C-ADCT-HC over 1C-ADCT-HC (for the same blocksize) are primarily due to the more efficient 2-stage block classification used in 3C-ADCT-HC and not because of the separate encoding of the primary component. We verified this by simulating a modified version of 3C-ADCT-HC in which the 2-stage classification was replaced by a 1-stage classification similar to that of 1C-ADCT-HC; the PSNR performance of this modified scheme was even inferior to that of 1C-ADCT-HC. Finally, we should mention that the separate encoding of the primary component is responsible for the improved perceptual quality of 3C-ADCT-HC near the strong edges.

As an alternative to Huffman coding, we have also considered arithmetic coding of the quantized DCT coefficients in all DCT-based schemes considered. The arithmetic coded counterpart of JPEG-HC, 1C-ADCT-HC and 3C-ADCT-HC are denoted, respectively, JPEG-AC, 1C-ADCT-AC and 3C-ADCT-AC. For the theory behind arithmetic coding, its operation and advantages the reader is referred to [37], [38]. The PSNR of the arithmetic coded schemes are also tabulated in Table 3.2. Clearly, in all cases arithmetic coding performs better than Huffman coding in terms of reduced encoding rate.

In addition to comparisons with JPEG, we make comparisons against a number of other entropy-coded schemes which have been reported in the literature recently. Specifically, we consider the constant block distortion ADCT (CBD-

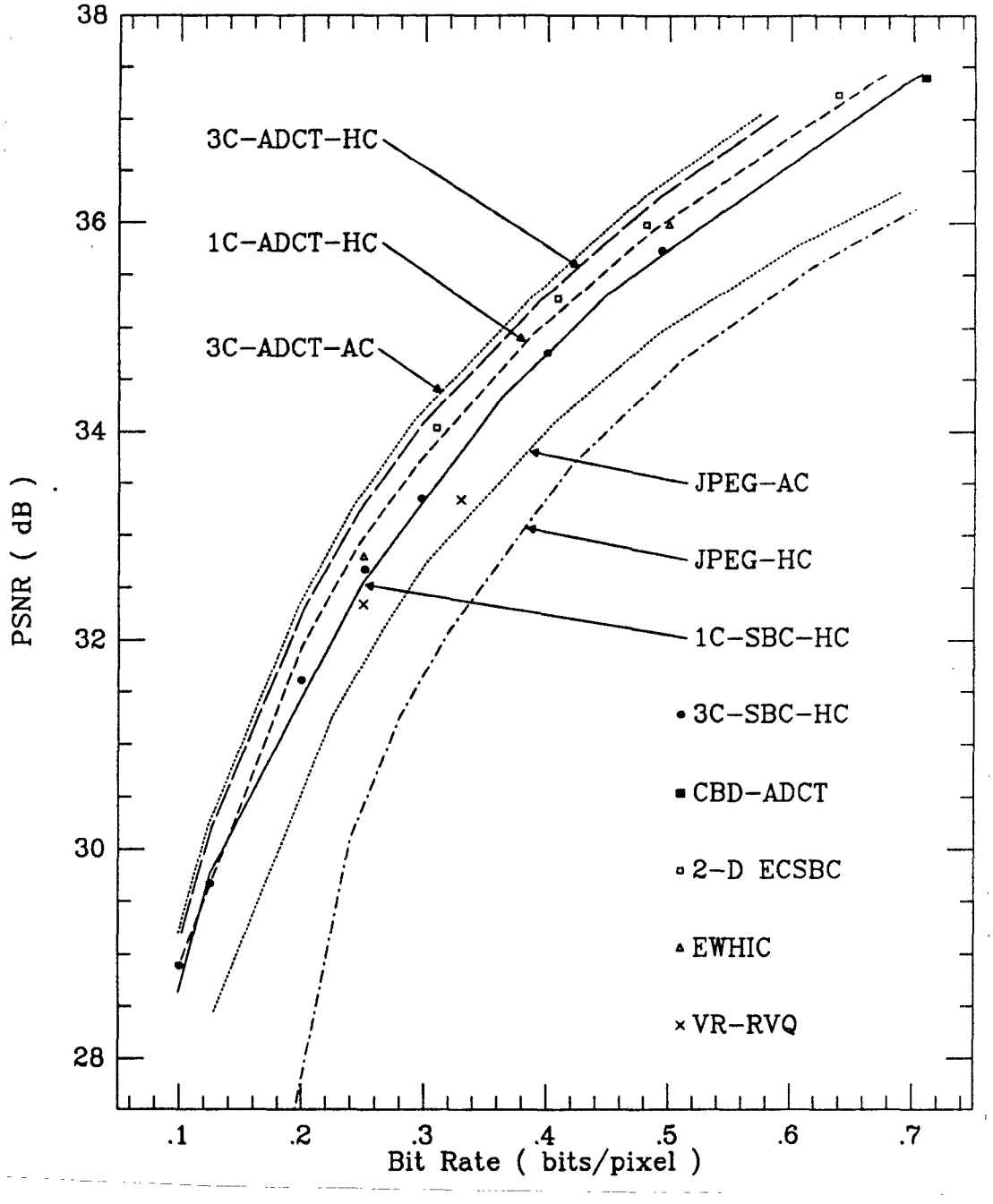


Figure 3.17: PSNR (in dB) vs. bit rate for various encoding schemes; 512×512 Lenna.

Encoding Scheme	Design Bit Rate (bpp)			
	0.125	0.25	0.5	0.75
JPEG-HC	23.27 (0.214)	25.95 (0.296)	29.17 (0.488)	31.62 (0.762)
1C-ADCT-HC	25.42 (0.129)	28.02 (0.256)	31.27 (0.503)	33.75 (0.746)
3C-ADCT-HC	25.61 (0.128)	28.41 (0.256)	31.62 (0.505)	33.83 (0.749)
1C-SBC-HC	25.55 (0.125)	27.98 (0.250)	31.07 (0.503)	33.36 (0.750)
3C-SBC-HC	25.50 (0.125)	28.15 (0.251)	31.18 (0.503)	33.36 (0.752)
JPEG-AC	23.27 (0.124)	27.49 (0.291)	30.21 (0.516)	32.18 (0.762)
1C-ADCT-AC	25.42 (0.120)	28.02 (0.238)	31.27 (0.492)	33.75 (0.728)
3C-ADCT-AC	25.61 (0.120)	28.41 (0.240)	31.62 (0.474)	33.80 (0.707)

Table 3.3: PSNR (in dB) performance of various encoding schemes for 256×256 Lenna at different design bit rates. Numbers in parantheses indicate actual bit rates.

ADCT) of [7], the 2-D adaptive entropy coded SBC (2-D ECSBC) of [12] (System C with arithmetic coding in [12]), the embedded wavelet hierarchical image coder (EWHIC) of [46] and the variable-rate residual vector quantizer (VR-RVQ) of [47]. Since the PSNRs of these schemes are not all reported at the bit rates considered in Table 3.2, we have plotted the PSNR vs. bit rate performance of all these schemes in Fig. 3.17. We shall leave the comparisons to the reader.

For the sake of completion, we have repeated our simulations for the 256×256 version of Lenna and generated a similar PSNR vs. bit rate plot in Fig. 3.18. Where available, the PSNR of other schemes is also included. These PSNR results are also tabulated in Table 3.3, where for 1C-ADCT-HC and 1C-ADCT-

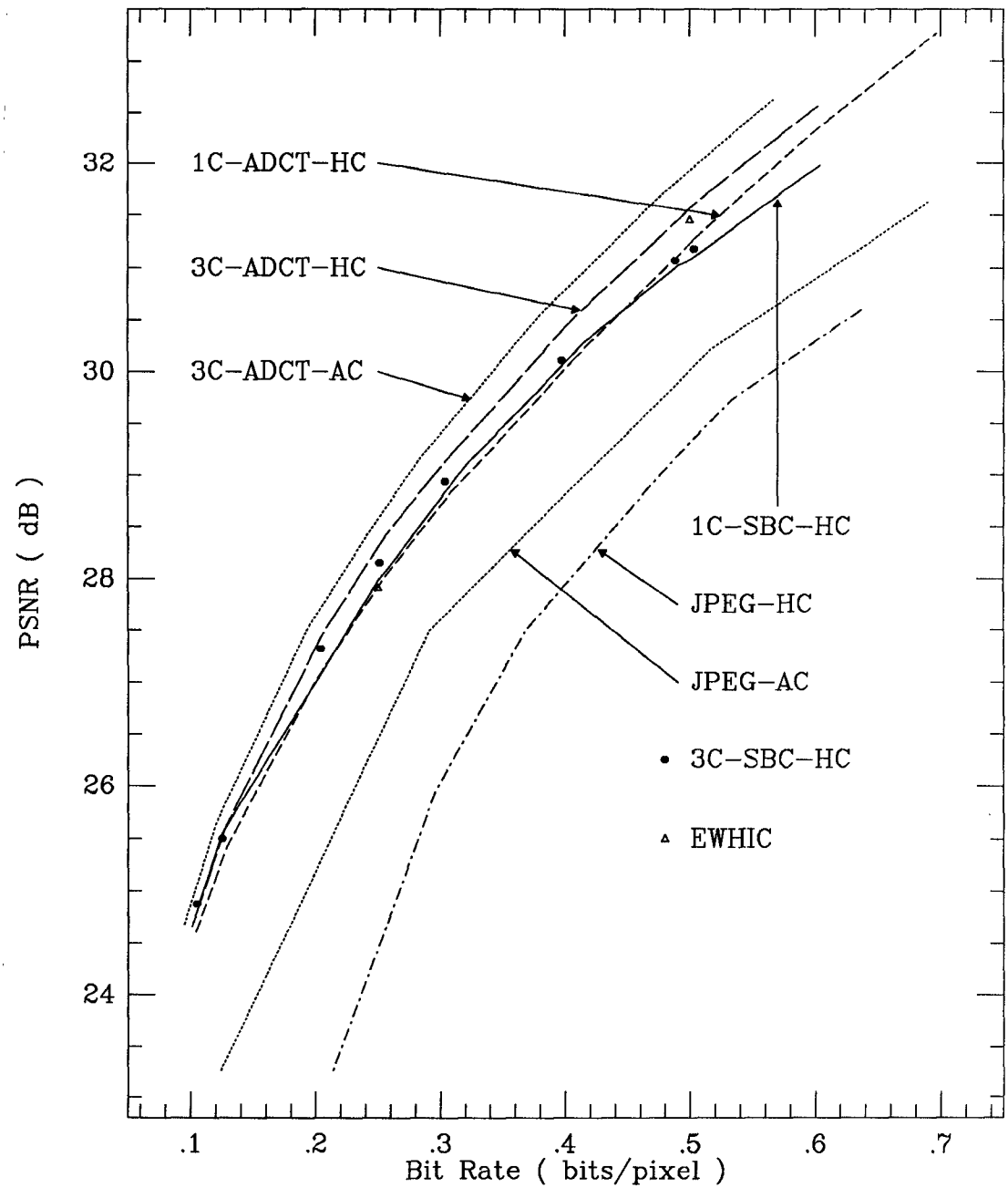


Figure 3.18: PSNR (in dB) vs. bit rate for various encoding schemes; 256×256 Lenna.

AC at design bit-rates 0.5 and 0.75 bits/pixel and for 3C-ADCT-HC at 0.75 bits/pixel, block size 8 is used.[6] An important observation here is the noticeable gain obtained by arithmetic coding over Huffman coding obtained in the 256×256 case.

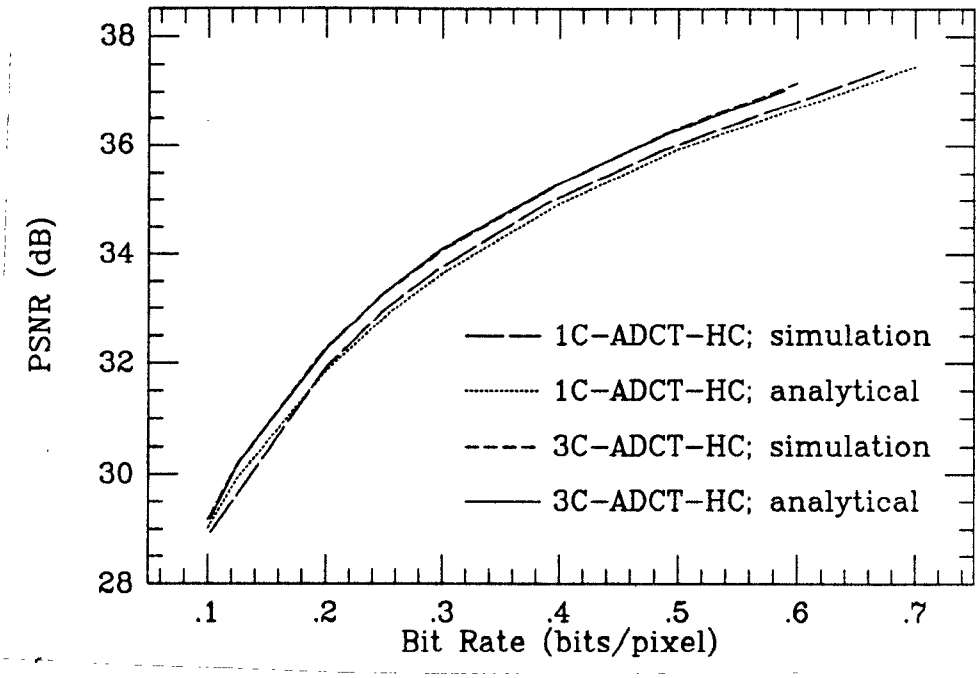
To close this section, we provide analytical PSNR results based on the average distortion formula in (3.4). These analytical results (in terms of PSNR) are computed and plotted in Fig. 3.19 for both 1C-ADCT-HC and 3C-ADCT-HC schemes. For the 1C-ADCT-HC scheme, the overall rate is r_{st} (see Eq. (3.5)) while for 3C-ADCT-HC, it is given by $r_{st} + r_p$. For comparison purposes, we have also plotted the actual simulation results in Fig. 3.19. The closeness of the analytical and simulation results, especially for the 3C-ADCT-HC case, is quite striking.

3.6 Perceptual Weighting of Distortions

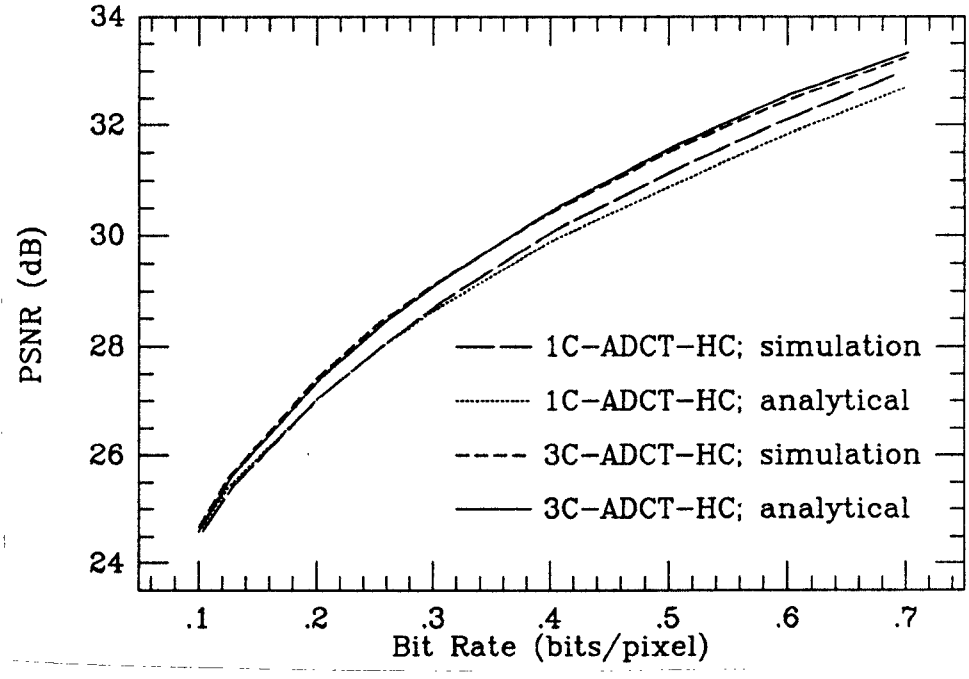
In this section, we will continue the effort to achieve high perceptual quality for the reconstructed images at low bit rates by adopting an approach based on the *contrast sensitivity* [1], [34], [35] of the human visual system (HVS). We apply the contrast sensitivity approach to the ADCT-based schemes only; similar ideas have been used in a subband coding framework in [48].

We briefly describe the concept of contrast sensitivity and refer the reader

[6]Experimental results show that for the ADCT-based schemes block-size 8 generally offers better (worse) PSNR performances at high (low) bit rates than block-size 16 does, and that the cross-points of the PSNR-rate-curves for block-size 8 and 16 are above 0.75 bits/pixel for encoding 512 × 512 Lenna, and are between 0.5 bits/pixel and 0.75 bits/pixel for encoding 256 × 256 Lenna.



(a)



(b)

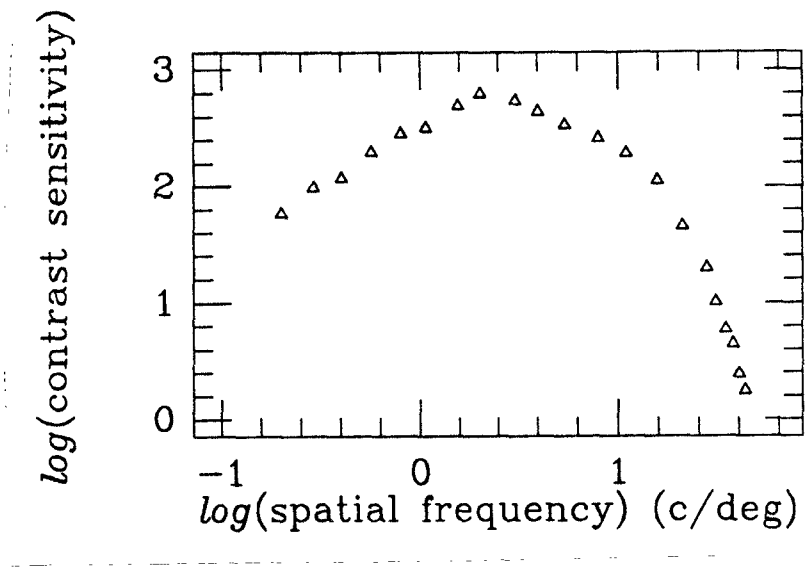
Figure 3.19: Analytical and simulation PSNR (in dB) vs. bit rate for 1C-ADCT-HC and 3C-ADCT-HC; (a) 512 × 512 Lenna, (b) 256 × 256 Lenna.

to [1], [34] for details. To determine the contrast sensitivity of the HVS, a test stimulus of the form of a grating pattern is presented to an observer. The grating pattern consists of vertical bars with sinusoidal-intensity scanlines [34]. The sine wave has mean $\overline{m}$, amplitude a and frequency f, defined as the reciprocal of the angle, subtended at the observer's eye, of one complete cycle, in cycles/degree (c/deg). The visibility threshold [1] of the pattern is measured for different choices of $\overline{m}$, a and f. It is shown that for a given spatial frequency f, the visibility threshold of the pattern depends primarily on $a/\overline{m}$, and not separately on a and $\overline{m}$. Upon defining $a/\overline{m}$ as the *contrast* of the grating, the *threshold contrast* is defined as the contrast below which the grating pattern is barely detectable. The reciprocal of the threshold contrast at frequency f is called the *contrast sensitivity* at f and denoted by $m(f)$. The contrast sensitivity curve is measured and presented in [34]. We have reproduced their curve (by reading the coordinates off the graph) and presented it in Fig. 3.20 (a). The important result is that the contrast sensitivity is very much frequency dependent; it peaks around $2 \sim 4$ c/deg and then rapidly falls off with increasing frequency.

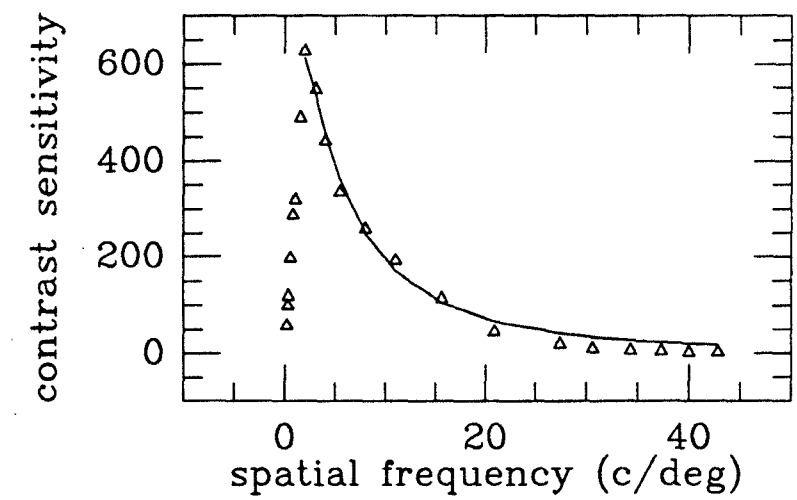
To get a closed-form approximation of the contrast sensitivity samples shown in Fig. 3.20 (a), we assume that $\log(m(f))$ is quadratic in $\log(f)$, which implies that $m(f)$ is of the form:

$$m(f) = a \times b^{(\log(\frac{f}{c}))^d}, \tag{3.9}$$

for some constants a, b, c and d. The optimum (in the sense of minimizing the



(a)



(b)

Figure 3.20: Contrast sensitivity for sinusoidal-intensity gratings (a) logarithmic scale and (b) normal scale along with the least-squares fit.

squared fitting error for $f > 2$ c/deg [7]) values of these constants are determined using CONSOLE, an optimization-based design tool [49]. These values are: $a = 621.31$, $b = 0.14$, $c = 1.73$ and $d = 1.83$. The original data points along with their least-squares fit are shown in Fig. 3.20 (b). In the sequel, we make a slight abuse of notation and use $m(f)$ to denote the least-squares fit to the empirical contrast sensitivity curve.

Next we describe how the contrast sensitivity of the HVS can be used for perceptual tuning of the ADCT-based image coding scheme. Consider a generic block of 2-D DCT coefficients, $\{d(u,v);\ u,v = 0,1,\ldots,L-1\}$. Let the error in quantizing $d(u,v)$ be denoted by $e(u,v)$. Then the resulting error in the spatial domain is given by [16]

$$\frac{2}{L}\alpha(u)\alpha(v)e(u,v)\cos\frac{\pi u(2i+1)}{2L}\cos\frac{\pi v(2j+1)}{2L},\quad i,j = 0,1,\ldots,L-1, \quad (3.10)$$

where $\alpha(0) = 1/\sqrt{2}$ and $\alpha(u) = 1,\ u \neq 0$.

Now consider the simplified case where $d(u,v) = 0$, for all $(u,v) \neq (0,0)$, and $e(u,v) = 0$, for all $(u,v) \neq (0,v_1)$, where $v_1 \neq 0$. Then the corresponding block in the spatial domain is given by

$$\frac{1}{L}(d(0,0) + \sqrt{2}e(0,v_1)\cos\frac{\pi v_1(2j+1)}{2L}),\quad i,j = 0,1,\ldots,L-1, \quad (3.11)$$

which is exactly of the form of the sinusoidal-intensity grating used as the test stimulus for determining the contrast sensitivities in [34]. By the definition of

[7] On the monitor used to view images in this study, a 16 × 16 block is of size 4 mm by 4 mm. Setting the viewing distance to be 570 mm as in [34], the angle subtended at the observer's eye of a 16 × 16 block is 0.4 degree. Thus, for $L = 16$, the $(0,1)$th and $(1,0)$th 2-D DCT coefficients have a spatial frequency of 2.5 c/deg; other coefficients have higher spatial frequencies. Therefore, in studying $m(f)$, the range $f > 2$ c/deg is adequate for this work.

the contrast sensitivity, the error $e(0, v_1)$ will be essentially undetectable by a human observer if [8]

$$e(0, v_1) \leq \frac{d(0,0) + c_1 L}{\sqrt{2} m(f(v_1))}, \tag{3.12}$$

where $f(v_1)$ is the spatial frequency of the waveform $\cos \frac{\pi v_1 (2j+1)}{2L}$ measured in c/deg, and the constant c_1 is the luminance measurement of the block when $d(u, v) = 0$ for all (u, v).

The above scenario represents an over-simplified case. In practice, $d(u, v)$ are not necessarily all zero for $(u, v) \neq (0, 0)$, and $e(u, v)$ are generally non-zero. A complete investigation of the perceptual effects of the quantization errors for such general cases, requires a study of the superposition of sinusoidal-intensity gratings in both horizontal and vertical directions displayed on non-uniform backgrounds.[9] Such an investigation is more of a psychovisual flavor and is beyond the scope of this paper.

However, the above over-simplified case can be used to provide a guideline for incorporating the contrast sensitivity of the HVS into our ADCT-based image coding system. Specifically, we argue that for good perceptual quality the quantization error $e(u, v)$ should be proportional to $(d(0,0) + c_1 L)/m(f(u, v))$, for $(u, v) \neq (0, 0)$, where $f(u, v)$ is the spatial frequency in c/deg of the waveform associated with the (u, v)th DCT coefficient. Obviously, such a condition necessitates a different quantization rule for each block. To circumvent this problem,

[8] Note that according to the definition of 2-D DCT in [16], $d(0, 0)/L$ is the mean intensity of the block.

[9] The findings in [34] suggest the existence, within the nervous system, of *linearly* operating *independent* mechanisms selectively sensitive to limited ranges of spatial frequencies. This observation justifies, to a certain extent, the independent treatment of the quantization error effects of each of the 2-D DCT coefficients.

we require, instead, that

(i) $e(u,v)$ be proportional to $\hat{d}(0,0)+c_1L$, where $\hat{d}(0,0)$ is the quantized version of $d(0,0)$, and

(ii) the variance of $e(u,v)$ be inversely proportional to $m^2(f(u,v))$.

In the following, we describe a scheme designed to achieve these two requirements.

Denote the (m,n)th block of 2-D DCT [10] coefficients by $\mathbf{d}_{m,n}$, $m,n=0,1,\ldots,(M/L)-1$, where $\mathbf{d}_{m,n}\equiv\{d_{m,n}(u,v);\ u,v=0,1,\ldots,L-1\}$. We assume that the classification is performed as described in Subsection 3.4.1. The dc coefficients $d_{m,n}(0,0)$ are quantized independently of the classification and the perceptual weighting described below; their quantized versions are denoted by $\hat{d}_{m,n}(0,0)$. To account for the perceptual role of the block luminance, the other 2-D DCT coefficients, $d_{m,n}(u,v)$, $(u,v)\neq(0,0)$, are modified as follows,

$$d'_{m,n}(u,v)=\frac{d_{m,n}(u,v)}{\hat{d}_{m,n}(0,0)+c_1L},\ m,n=0,1,\ldots,(M/L)-1. \tag{3.13}$$

The variances of these modified 2-D DCT coefficients are denoted by $\sigma'^2_k(u,v)$, $(u,v)\neq(0,0)$, where $k=0,1,\ldots,K-1$, is the classification index. A weighted version of these variances is used for the bit allocation described below. The quantization is performed on $d'_{m,n}(u,v)$; the quantized version of $d_{m,n}(u,v)$ is obtained from that of $d'_{m,n}(u,v)$ by multiplying back by the scale factor, $(\hat{d}_{m,n}(0,0)+c_1L)$. We denote the quantization errors for $d'_{m,n}(u,v)$ and $d_{m,n}(u,v)$ by $e'_{m,n}(u,v)$ and $e_{m,n}(u,v)$, respectively. Obviously,

$$e_{m,n}(u,v)=(\hat{d}_{m,n}(0,0)+c_1L)e'_{m,n}(u,v). \tag{3.14}$$

[10]The 2-D DCT could be performed either on the image itself or on its smooth-texture component.

To ensure that the quantization error variance for the (u,v)th coefficient is inversely proportional to $m^2(f(u,v))$, we introduce, the weighted variances $\sigma'^2_k(u,v)\ m^2(f(u,v))$, $k = 0,1,\ldots,\ K-1$, $(u,v) \neq (0,0)$. These weighted variances are subsequently used for bit allocation among all 2-D DCT coefficients of the K classes except the dc coefficient; the number of bits used for the dc coefficient is as given before the perceptual weighting. The rationale behind this weighting of the variances is as follows.

Let us use $\epsilon'^2_k(u,v)$ to denote the variance of $e'_{m,n}(u,v)$ when the (m,n)th block is in class k. Then, using arguments based on the Shannon Lower Bound similar to those given in Subsection IV.A, we have the following approximate relationship:

$$m^2(f(u,v))\epsilon'^2_k(u,v) = \text{const.}, \tag{3.15}$$

which implies that $\epsilon'^2_k(u,v)$ is inversely proportional to $m^2(f(u,v))$ — exactly what we had set out to achieve.

We have modified the ADCT coding part of both the 1C-ADCT-HC and 3C-ADCT-HC schemes according to the above mentioned perceptual weighting and obtained simulations at different bit rates. The reconstructed images are shown in Figs. 3.21 and 3.22; Figs. 3.21 (a), (c) and Fig. 3.22(a) illustrate the results for the 3C-ADCT-HC scheme while Figs. 3.21 (b), (d) and Fig. 3.22(b) are for the 1C-ADCT-HC system. In these simulations, we have used $f(u,v) = 2.5\sqrt{u^2+v^2}$ [11] and $c_1 = 512$. The choice of c_1 is determined by experimentation.

Comparing the 3C-ADCT-HC results in Figs. 3.10-3.12 against those in

[11] As mentioned before, the $(0,1)$th and $(1,0)$th 2-D DCT coefficients have a spatial frequency of 2.5 c/deg. The choice of $f_k(u,v) = 2.5\sqrt{u^2+v^2}$ guarantees consistency with this observation.

(a) (b)

(c) (d)

Figure 3.21: (a), (c) 3C-ADCT-HC with perceptual weighting at design bit rates 0.5, 0.25 bpp, respectively; (b), (d) 1C-ADCT-HC with perceptual weighting at design bit rates 0.5, 0.25 bpp, respectively.

(a) (b)

Figure 3.22: (a) 3C-ADCT-HC with perceptual weighting at design bit rates 0.125 bpp; (b) 1C-ADCT-HC with perceptual weighting at design bit rates 0.125 bpp.

Figs. 3.21 and 3.22, it is clearly seen that the perceptually-weighted scheme offers a better subjective performance; the improvement is quite visible at low bit rates. However, for the 1C-ADCT-HC results the reverse conclusion holds. Specifically, the perceptually-weighted 1C-ADCT-HC scheme suffers from a visible degradation near the strong edges. This phenomenon can be explained as follows. As compared with the straight 1C-ADCT-HC scheme, its perceptually-weighted version results in allocating more bits to the low frequency coefficients and fewer bits to the high frequency coefficients. Therefore, for texture regions, the quantization error generally increases but this has little perceptual effect. In smooth regions, the image is better reproduced and in particular the blockiness is reduced due to the extra bits received for low frequency coefficients. How-

ever, in those blocks dominated by strong edges, the perceptual weighting has a detrimental effect: The high frequency components (which are generally strong due to the sharp transitions of the strong edges) are more coarsely quantized and the resulting quantization errors are spread over the entire block in the spatial domain resulting in visible degradation of the uniform regions on both sides of the strong edge. In contrast with the 1C-ADCT-HC scheme, in the 3C-ADCT-HC scheme the strong edges are extracted and encoded separately; the perceptual weighting can thus be applied solely to the smooth and texture components without damaging the integrity of the strong edges.

3.7 Summary and Conclusions

We have developed a framework for image coding in which a combination of waveform coding and feature-based coding techniques is used to achieve high perceptual quality at low bit rates. The coding schemes revolve around a perceptually motivated three-component model by means of which the image is decomposed into three components. The primary component, which contains the strong edge information is encoded separately with little or no perceptual distortion. The remaining two components (smooth and texture) are encoded by entropy-coded ADCT or entropy-coded SBC. While the novelty of the proposed schemes reside primarily in the use of the three-component model and the separate encoding of its constituent components, there are some new elements in the adaptive DCT coding scheme, such as the classification procedure and the estimation of coefficient variances in the receiver, that have contributed to its good objective performance.

Based on extensive simulations and comparisons, we can make the following conclusions:

(i) The proposed encoding schemes based on the three-component model perform very well at all bit rates with the ADCT-based scheme consistently giving the best results. In all cases, the PSNR of the proposed schemes are significantly better than that of JPEG.

(ii) The main advantage of the three-component model is a more faithful reconstruction of the strong edge information; the difference is significant at low rates. In the SBC-based schemes, the three-component model results in a significant reduction of ringing around the strong edges.

(iii) The use of arithmetic coding in both ADCT- and SBC-based schemes results in performance improvements in terms of reduced bit rate; the improvements are more substantial for the 256×256 version.

We have also shown that the well-understood contrast sensitivity of the HVS can be used for perceptual tuning of the ADCT-based encoding scheme. We have developed a method for weighting the 2-D DCT coefficients based on the contrast sensitivity. When this perceptual tuning is used in conjunction with the three-component model, the subjective performance of the ADCT-based scheme is even further improved. Use of the perceptual weighting without the three-component model leads to a visible degradation of the strong edges and hence is not desirable.

We should note that both in the ADCT- and SBC-based schemes a very simple and low complexity quantization scheme is used for the DCT coefficients or

the subbands. Undoubtedly, use of more powerful quantization methods on the DCT coefficients or the subbands can result in performance improvements, of course, at the cost of some additional complexity. Specifically, it is well-known that the gap between the performance of entropy-coded quantizers used in this paper and the source rate-distortion function is about 1.53 dB. Therefore, other quantization methods that reduce this gap can be used to improve the overall performance. The entropy-constrained trellis coded quantization of Fischer and Wang [50] and the trellis-based scalar-vector quantization of Laroia and Farvardin [51] are two possible candidates. Work on these extensions is currently underway.

Chapter 4

Planar Curve Representation

4.1 Introduction

In image processing and computer vision, representation of curves or contours is an important problem, since the information to be processed and/or extracted is commonly expressed in the form of curves [52] - [64]. For example, an object can be recognized based on the bounding contour of the object [54], [56], [57], and a Landsat satellite image can be registered to a map using the coastal lines contained in the Landsat image [62].

The simplest way to represent a planar curve is to specify every sample points on the curve, which apparently is not the most efficient way. For example, considering a rectangular contour, the four corner-points are the most important points on the contour, since the locations of the four corners uniquely specify the rectangular contour. Notice that the four corners of the rectangular contour assume large curvatures along the contour. The importance of such significant curvature points is also recognized in a tachistoscopic study of visual perception [58]; significant curvature points are suggested to be high in information

content. In addition to simplifying and rendering structures to the representations of planar curves, the significant curvature points can be used as the feature points for the next level of processing, e.g., contour approximation and computer recognition of objects [54] - [57], [59].

To extract the significant curvature points, people have used Laplacian-Gaussian operator (LGO) based schemes [52] (and the references therein); in these LGO-based schemes, the problem is transformed into the domain of orientation along the curve, and the LGO is applied to detect the significant changes in orientation. Multiple LGOs are used in these schemes in order to overcome the poor detecting-accuracy problem of the LGO in the presence of noise. However, the usage of multiple LGOs leads to additional difficulties which will be discussed in detail later on.

In this chapter, we develop a non-LGO-based scheme for locating significant curvature points. The idea for this new scheme is similar to those used for generating the stressed image and for extracting strong edges presented in Chapter 2. More specifically, this new scheme is a complex-valued 1-D variant of the strong edge extraction scheme for the three-component image model. Since the LGO is not used, the difficulties of the LGO-based schemes do not present in the new scheme. A comparison between the new scheme and a LGO-based scheme will be provided after the development of the new scheme.

The rest of this chapter is organized as follows. In Section 4.2, the new scheme is developed. Then a comparison with the curvature primal sketch scheme of [52] is provided in Section 4.3. A number of applications are described in Section 4.4. Section 4.5 contains a summary and conclusions.

4.2 Generation of the Stressed Curve and Extraction of Significant Curvature Points

Let us consider a planar curve specified by a finite set of samples on the curve: $\mathcal{C} \equiv \{(x_i, y_i)\}_{i=0}^{m-1}$, where m is the length of the sampled curve. The curve $\mathcal{C}$ can be open or closed; in the latter case, sample (x_0, y_0) is the continuation of (x_{m-1}, y_{m-1}). We now define a new curve: $\mathcal{C}^s \equiv \{(x_i^s, y_i^s)\}_{i=0}^{m-1}$, referred to as the stressed curve associated with $\mathcal{C}$. The stressed curve $\mathcal{C}^s$, which is analogous to the stressed image $\mathcal{X}^s$ of an image $\mathcal{X}$ (ref. Chapter 2), is related to $\mathcal{C}$ through the following two types of quantities: (i) the sample squared-error distances to $\mathcal{C}$,

$$D_i(\mathcal{C}^s, \mathcal{C}) \equiv (x_i - x_i^s)^2 + (y_i - y_i^s)^2, \quad i = 0, 1, \ldots, m-1 \tag{4.1}$$

and (ii) the *sample curvature energies* of $\mathcal{C}^s$,

$$H_i(\mathcal{C}^s) \equiv \begin{cases} (x_{i-1}^s - 2x_i^s + x_{i+1}^s)^2 + (y_{i-1}^s - 2y_i^s + y_{i+1}^s)^2 & i = 1, \ldots, m-2, \\ (x_{m-1}^s - 2x_0^s + x_1^s)^2 + (y_{m-1}^s - 2y_0^s + y_1^s)^2 & i = 0 \text{ and} \\ & \text{when } \mathcal{C} \text{ is closed}, \\ (x_{m-2}^s - 2x_{m-1}^s + x_0^s)^2 + (y_{m-2}^s - 2y_{m-1}^s + y_0^s)^2 & i = m-1 \text{ and} \\ & \text{when } \mathcal{C} \text{ is closed}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

Similar to the generation of the stressed image, the stressed curve $\mathcal{C}^s$ is generated by recursively solving the following minimization problem,

$$\min_{\{(x_i^o, y_i^o)\}} F(\mathcal{C}^o, \mathcal{C}, \Lambda), \tag{4.3}$$

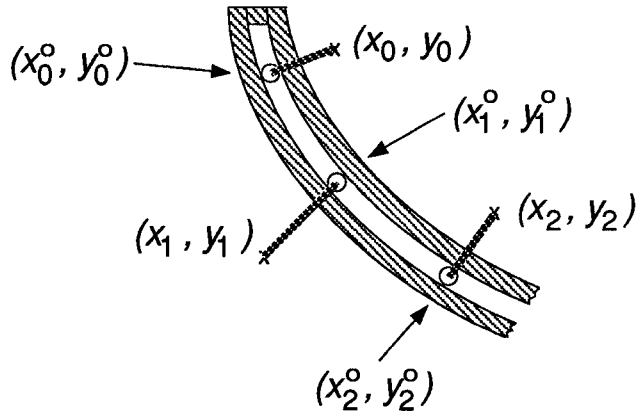


Figure 4.1: Mechanical structure for $F(\mathcal{C}^o, \mathcal{C}, \Lambda)$.

where $\mathcal{C}^o \equiv \{(x_i^o, y_i^o)\}_{i=0}^{m-1}$, $F(\mathcal{C}^o, \mathcal{C}, \Lambda)$ is defined by

$$F(\mathcal{C}^o, \mathcal{C}, \Lambda) \equiv \sum_{i=0}^{m-1} \lambda_i D_i(\mathcal{C}^o, \mathcal{C}) + H_i(\mathcal{C}^o), \tag{4.4}$$

where $\Lambda \equiv \{\lambda_i\}$ is the given parameter set.

The quantity $F(\mathcal{C}^o, \mathcal{C}, \Lambda)$ is similar to $E(\mathcal{X}^s, \mathcal{X}, \Lambda)$ defined in Equation (2.13) and has an interpretation of the potential energy of a mechanical structure. A portion of this mechanical structure is shown in Fig. 4.1. In this structure, at each point along the curve $\mathcal{C}$, we have a spring with one end fixed at that point and the other end attaching a ball which is fitted inside the slot of a flexible bar; the ball can move inside the slot without friction and always assumes a position closest to the corresponding point of $\mathcal{C}$ for any given shape of the flexible bar. The positions of the balls are (x_i^o, y_i^o), $i = 0, 1, \ldots, m-1$, which specify the corresponding configuration of the structure. Shown in Fig. 4.1 is the portion of the structure for the first three points of the curve $\mathcal{C}$. If the curve is closed, the end of bar near (x_0, y_0) is connected with the other end

of the bar near (x_{m-1}, y_{m-1}). The potential energies of the springs for a given configuration $\{(x_i^o, y_i^o)\}$ are approximated by $\lambda_i D_i(\mathcal{C}^o, \mathcal{C})$; the potential energy of the entire bar is approximated by $\sum_{i=0}^{m-1} H_i(\mathcal{C}^o)$. The parameters $\{\lambda_i\}$ controls the rigidness of the springs [28]. For a given parameter set Λ, the structure will reach to the final stable configuration which assumes the minimum potential energy. More specifically, let us denote the stable configuration by $\{(x_i^*, y_i^*)\}$. Then $\mathcal{C}^* \equiv \{(x_i^*, y_i^*)\}$ is the solution of the minimization problem (4.3). Note that when λ_i is very large, the corresponding spring would be very strong and (x_i^*, y_i^*) would be very close to (x_i, y_i); on the other hand, when λ_i is relatively small, the spring would be weak and the flexible bar would be less bended around that location.

The procedure for generating the stressed curve $\mathcal{C}^s$ can be described intuitively using this mechanical structure interpretation. Initially, all the springs are of the same rigidness, i.e., $\lambda_i =$ a constant for all i. Then the structure reachs the stable configuration; the balls in this stable configuration form a curve which is a uniformly smoothed version of the contour $\mathcal{C}$. In the next step, the curvature energies along the bar are examined; for the places of high curvature energies, the springs are made stronger, i.e., λ_is at those locations are increased. Then again, the structure is let to reach the new stable configuration, and the above procedure is repeated. This process is described more specifically in the rest of the chapter. Some of the details are omitted since they are similar to the corresponding procedures in Chapter 2. Additionally, the processes for generating $\mathcal{C}^s$ and for extracting significant curvature points on $\mathcal{C}^s$ are combined since $\mathcal{C}^s$ will not be used explicitly here. When the need for $\mathcal{C}^s$ arises, the following procedure can be modified accordingly.

The minimization problem (4.3) is solved via the following system of linear equations,

$$\nabla_{\{(x_i^o,y_i^o)\}}F(\mathcal{C}^o,\mathcal{C},\Lambda)\equiv 0, \tag{4.5}$$

where $\nabla_{\{(x_i^o,y_i^o)\}}F(\mathcal{C}^o,\mathcal{C},\Lambda)$ denotes the gradient of $F(\mathcal{C}^o,\mathcal{C},\Lambda)$ with respect to $\{x_i^o\}$ and $\{y_i^o\}$. This system of linear equations has a unique solution when $\lambda_i > 0$ for all i (ref. Subsection 2.3.2). This solution is obtained iteratively using the Gauss-Seidal iteration. One step of the Gauss-Seidal iteration for generating the $(k+1)$th intermediate solution $\mathcal{C}^{k+1} \equiv \{(x_i^{k+1}, y_i^{k+1})\}$ from $\mathcal{C}^k$ is described in the following conventional pseudo-code:

For $(i = 0$ to $m-1)$
$\{x_i^k = f_i(\mathbf{x}^k);$
$y_i^k = f_i(\mathbf{y}^k);\ \}$
For $(i = 0$ to $m-1)$
$\{x_i^{k+1} = x_i^k;$
$y_i^{k+1} = y_i^k;\ \}$

where $f_i(\mathbf{x}^k)$ is defined as follows, ($f_i(\mathbf{y}^k)$ is defined similarly),

$$f_i(\mathbf{x}^k) \equiv \frac{1}{\lambda_i+6}(\lambda_i x_i + 4x_{i-1}^k + 4x_{i+1}^k - x_{i-2}^k - x_{i+2}^k), \tag{4.6}$$

for $i = 2,\ldots,m-3$. For the rest of the samples, the formula for $f_i(\mathbf{x}^k)$ is different for open and closed curves:

$$f_0(\mathbf{x}^k) = \begin{cases} \frac{1}{\lambda_0+1}(\lambda_0 x_0 + 2x_1^k - x_2^k) & \text{open curve,} \\ \frac{1}{\lambda_0+6}(\lambda_0 x_0 + 4x_{m-1}^k + 4x_1^k - x_{m-2}^k - x_2^k) & \text{close curve.} \end{cases}$$

$$f_1(\mathbf{x}^k) = \begin{cases} \frac{1}{\lambda_1+5}(\lambda_1 x_1 + 2x_0^k + 4x_2^k - x_3^k) & \text{open curve,} \\ \frac{1}{\lambda_1+6}(\lambda_1 x_1 + 4x_0^k + 4x_2^k - x_{m-1}^k - x_3^k) & \text{close curve.} \end{cases}$$

$$f_{m-2}(\mathbf{x}^k) = \begin{cases} \frac{1}{\lambda_{m-2}+5}(\lambda_{m-2}x_{m-2} + 2x^k_{m-1} + 4x^k_{m-3} - x^k_{m-4}) & \text{open curve,} \\ \frac{1}{\lambda_{m-2}+6}(\lambda_{m-2}x_{m-2} + 4x^k_{m-3} + 4x^k_{m-1} - x^k_{m-4} - x^k_0) & \text{close curve.} \end{cases}$$

$$f_{m-1}(\mathbf{x}^k) = \begin{cases} \frac{1}{\lambda_{m-1}+1}(\lambda_{m-1}x_{m-1} + 2x^k_{m-2} - x^k_{m-3}) & \text{open curve,} \\ \frac{1}{\lambda_{m-1}+6}(\lambda_{m-1}x_{m-1} + 4x^k_{m-2} + 4x^k_0 - x^k_{m-3} - x^k_1) & \text{close curve.} \end{cases}$$

The convergence of the above Gauss-Seidel iteration is guaranteed by an argument similar to that in Subsection 2.3.2.

The parameter set Λ is updated according to the sample curvature energies $H_i(\mathcal{C}^k)$ (ref. Subsection 2.3.2). The updating scheme, which is similar to the one for generating stressed images (Equation (2.20)), is given by

$$\lambda_i = \alpha H_i(\mathcal{C}^k),\ i = 0, 1, \ldots, m-1. \tag{4.7}$$

The significant curvature points of $\mathcal{C}^k$ are located by identifying points of large local maximum sample curvature energies. More specifically, we define the set of significant curvature points on $\mathcal{C}^k$, denoted by $\mathcal{Q}_T(\mathcal{C}^k)$, as follows,

$$\mathcal{Q}_T(\mathcal{C}^k) \equiv \{(x_i, y_i) \in \mathcal{C}^k; H_i(\mathcal{C}^k) > H_{i-1}(\mathcal{C}^k), H_i(\mathcal{C}^k) > H_{i+1}(\mathcal{C}^k), H_i(\mathcal{C}^k) > T\}, \tag{4.8}$$

where $T \geq 0$ and $0 < i < m-1$; for closed curve $\mathcal{C}$, i can be 0 and $m-1$, and $H_{-1}(\mathcal{C}^k) = H_{m-1}(\mathcal{C}^k)$, $H_m(\mathcal{C}^k) = H_0(\mathcal{C}^k)$ in the above definition.

Now we summarize the entire iterative procedure for locating the significant curvature points of $\mathcal{C}^s$ in Fig. 4.2. The procedure starts with $\mathcal{C}^0 = \mathcal{C}$, generates $\mathcal{C}^{N_\nu}$ using N_ν Gauss-Seidel iterations with a uniform initial parameter set Λ, and extracts $\mathcal{Q}_T(\mathcal{C}^{N_\nu})$. Then the stopping criterion is checked; if the answer is no, the procedure updates the parameter set Λ and returns to the Gauss-Seidel iteration to generate $\mathcal{C}^{2N_\nu}$. This process is repeated until the stopping criterion is satisfied, namely, (i) the number of extracted significant curvature points is

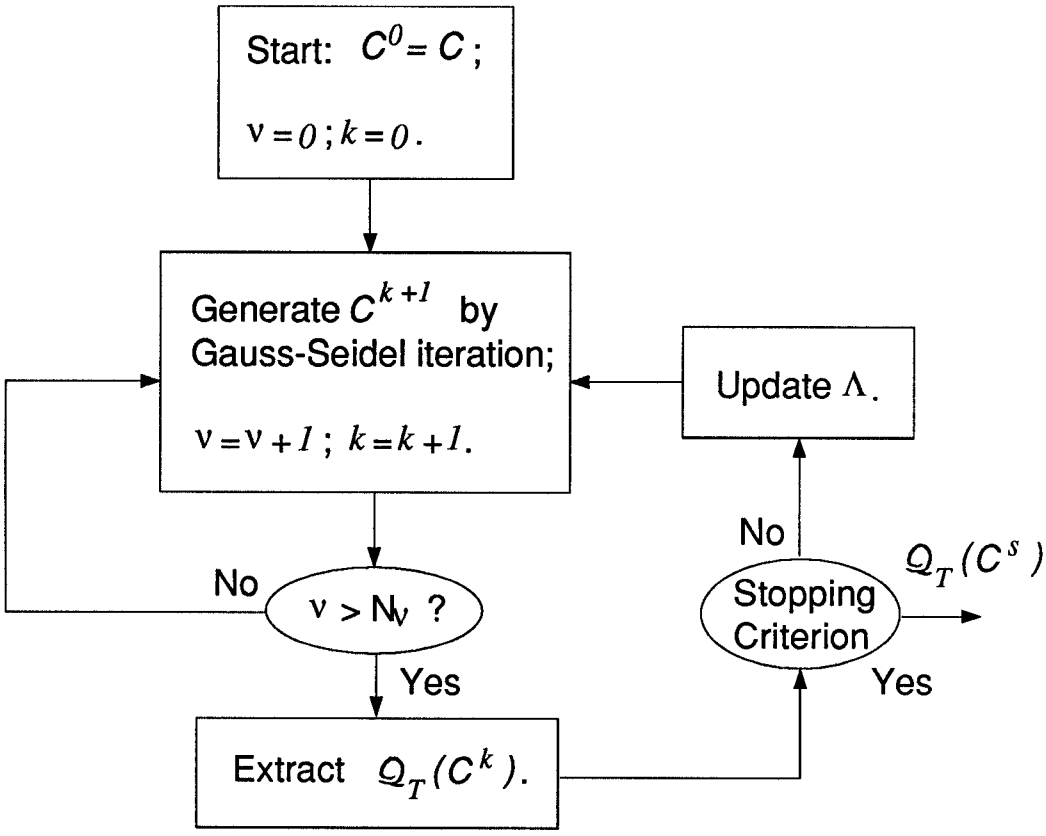


Figure 4.2: Flow chart describing the generation of the significant curvature point set $\mathcal{Q}_T(\mathcal{C}^s)$.

unchanged:

$$|\mathcal{Q}_T(\mathcal{C}^{\ell N_\nu})| = |\mathcal{Q}_T(\mathcal{C}^{(\ell-1)N_\nu})|, \tag{4.9}$$

for $\ell > 1$, and (ii) the Euclidean distance between the points of the current and the previous significant curvature point sets is small enough:

$$\max_{(x^{\ell-1},y^{\ell-1})\in\mathcal{Q}_T(\mathcal{C}^{(\ell-1)N_\nu});(x^\ell,y^\ell)\in\mathcal{Q}_T(\mathcal{C}^{\ell N_\nu})} \sqrt{(x^{\ell-1}-x^\ell)^2+(y^{\ell-1}-y^\ell)^2} < T_d, \tag{4.10}$$

where $T_d \geq 0$. (When $\ell = 1$, the stopping criterion is never satisfied.) The final output of the procedure is the significant curvature point set $\mathcal{Q}_T(\mathcal{C}^s)$.

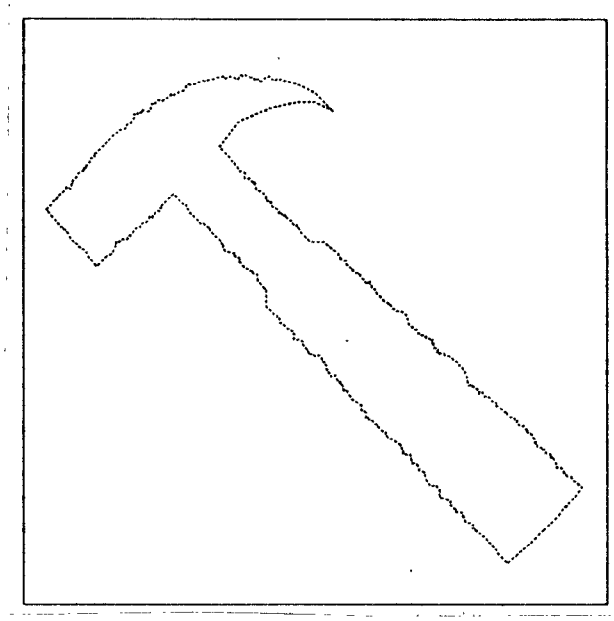


Figure 4.3: Warrington hammer test curve.

In the next section, we provide simulation results of the above procedure and a comparison with the curvature primal sketch scheme of [52].

4.3 Simulation Results and a Comparison

Consider a test curve shown in Fig. 4.3. This closed curve resembles an outline of a Warrington hammer. The significant curvature point set of this Warrington hammer test curve is generated with the scheme introduced in the last section. The parameters used are: $N_\nu = 100$, $\alpha = 1.5$, $m = 970$, $T = 10^{-3}$, $T_d = 0.1$, and initial parameter $\lambda_i = 10^{-5}$ for all i. The iterative procedure stops after once updating the parameter set. The resulting significant curvature point set is shown in Fig. 4.4; there are a total of seven significant curvature points which are marked by small crosses. The intermediate result, denoted by $\mathcal{Q}_T(\mathcal{C}^{N_\nu})$, is shown in Fig. 4.5; these intermediate significant curvature points gradually

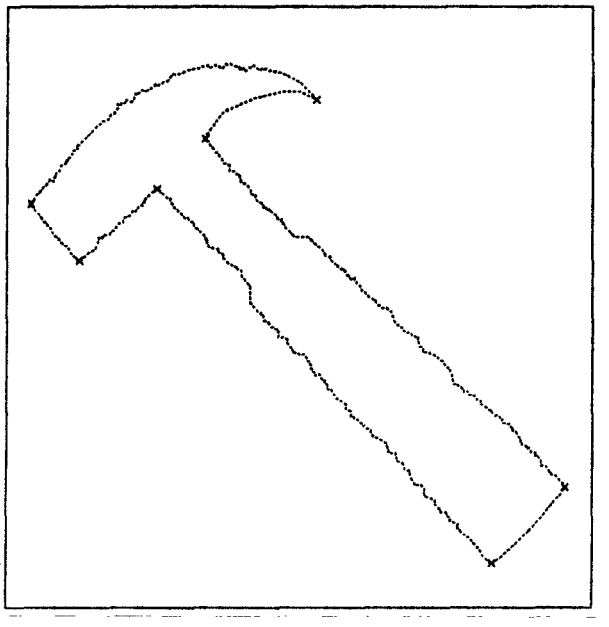


Figure 4.4: Warrington hammer test curve and the extracted significant curvature points.

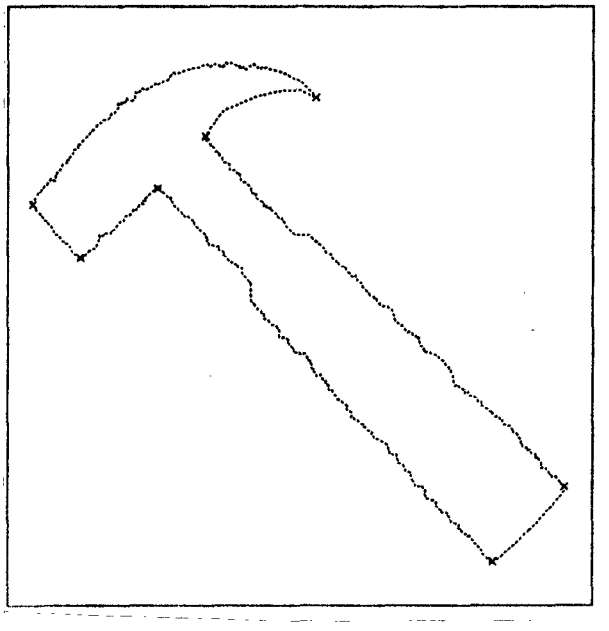


Figure 4.5: Intermediate set of significant curvature points $\mathcal{Q}_T(\mathcal{C}^{N_\nu})$.

approximate the final $\mathcal{Q}_T(\mathcal{C}^s) = \mathcal{Q}_T(\mathcal{C}^{2N_\nu})$. The computational complexity of this case is given in the following in terms of numbers of multiplications (M), additions (A) and bit-wise left shifts (S) per points: $M = 204$, $A = 805$, and $S = 802$.

The curvature primal sketch scheme of [52] uses the points at which the orientation of the curve changes significantly as the features to represent the curve. More specifically, the problem of locating these feature-points is transformed to orientation domain: The orientation is estimated at each point on the curve $\mathcal{C}$, and is denoted by $G(s)$, where s denotes arc length along the contour. Then the orientation $G(s)$ is convolved with $g'_\sigma(t)$ and $g''_\sigma(t)$, where $g_\sigma(t)$ is the Gaussian kernel with standard deviation σ. The locations of local extrema of $g'_\sigma \otimes G(s)$ and zero crossings of $g''_\sigma \otimes G(s)$ are extracted. These extracted locations indicate the possible existences of feature points and approximate the actual positions of the corresponding feature points.

Because of the uncertainty principle between the detectability of an event and its accurate localization in the presence of noise for the operators g'_σ and g''_σ [25], [52], (ref. Chapter 2), several values of σ have to be used to obtain feature points on multiple scales. Although, in principle, this multi-scale representation allows *coarse-to-fine tracking* [53] of feature points, the discretization of the scales generally results in some ambiguity in tracking across scales [52]. To annihilate such ambiguity, the curvature primal sketch scheme uses a set of curve segments as primitives; the behavior of the filtered response with g'_σ and g''_σ of these primitives over a range of scales are analyzed; the resulting descriptions are used in an interpretation process of the multi-scale representation of the feature points. In contrast to this primitive-based scheme, the technique developed in

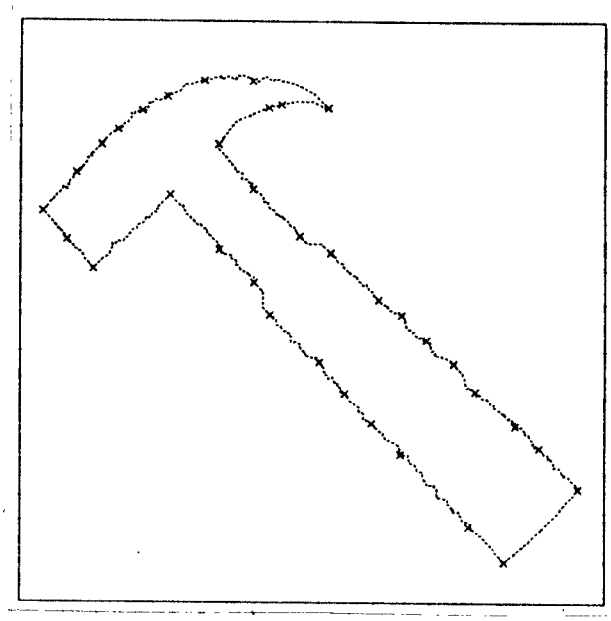
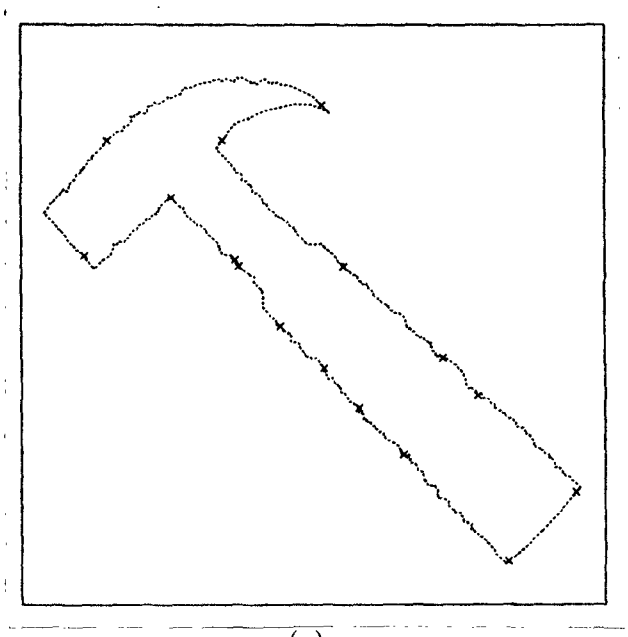


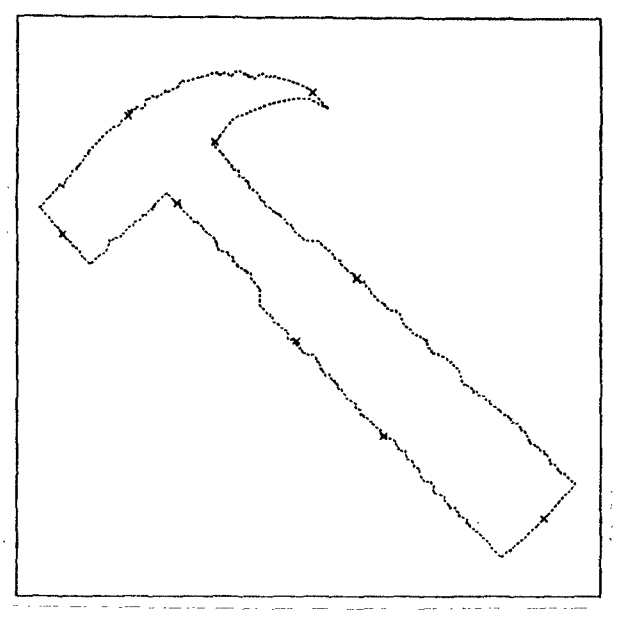
Figure 4.6: Zero-crossings of the filtered response with g''_{10}.

the previous subsection does not rely on the scale-space representation of the feature points, and, therefore, avoid entirely the difficulties associated with the ambiguity of scale-space tracking.

To demonstrate the behavior of the curvature primal sketch scheme, we apply the filter g''_σ to the Warrington hammer test curve of Fig. 4.2. Several results with $\sigma = 10, 20, 30$ are shown in Figs. 4.6 and 4.7, where the crosses mark the positions of the zero-crossings. Notice that, for $\sigma = 10$, the seven meaningful points extracted by our scheme (shown in Fig. 4.4) are all detected with fair accuracy, but a lot of spurious points are also located. For $\sigma = 20, 30$, the number of spurious points are gradually reduced, but those true detections have increasingly large locational deviations and some of them are even merged together, e.g., the two points for the end of the handle are combined into one point in the case of $\sigma = 30$. Therefore, from any one of the above three cases alone, the



(a)



(b)

Figure 4.7: Zero-crossings of the filtered response with (a) g''_{20} and (b) g''_{30}.

right detection of the seven points can not be achieved; this necessitates the usage of multiple values of σ and a tracking procedure [52]. The computation complexity for the filtering operation alone is summarized as follows. For $\sigma = 10$, $M = 87$, $A = 86$, $S = 0$; for $\sigma = 20$, $M = 177$, $A = 176$, $S = 0$; for $\sigma = 30$, $M = 267$, $A = 266$, $S = 0$. As demonstrated in [52], six scales are used to obtain satisfactory results. We assume that for the other three scales, the minimum computational complexities of the above three cases are incurred. Then, the total computational complexity for filtering operations is $M = 792$, $A = 786$ and $S = 0$ which is higher than that for generating $\mathcal{Q}_T(\mathcal{C}^s)$ with our scheme (since the multiplication operation is much more time consuming than the shift is, e.g., one multiplication of two 8 bits integers uses at least 8 times of the time for one shift.) The above total computational complexity does not include the operations for the filtering with g'_σ and for the tracking process required by the curvature primal sketch scheme.

Based on the above comparison, we may conclude that the new scheme extracts efficiently the significant curvature points at a reduced computational complexity as compared with the curvature primal sketch scheme.

4.4 Applications

Applications of using the significant curvature points as the features to represent planar curves are abundant. To name a few of these applications, the feature-points can be used as the knot points in a spline approximation to the contour [54] - [59], to generate a semantic network representation of the shape [60], [61], and to register a Landsat satellite image of an area to a map [62]. In this

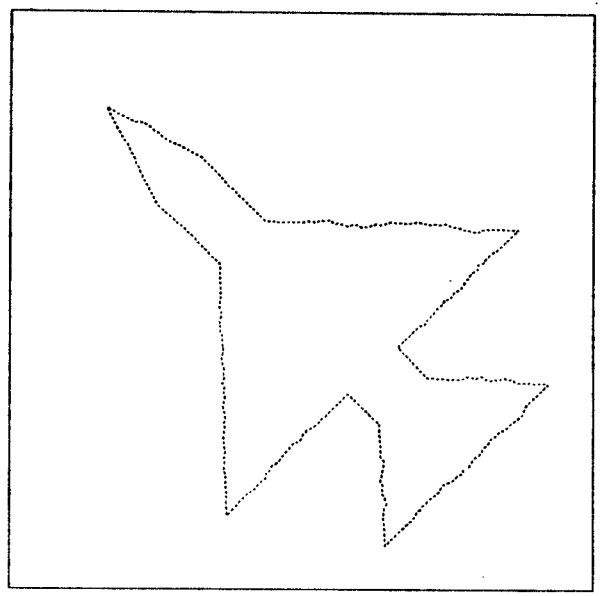


Figure 4.8: Fighter airplane contour.

subsection, we provide two examples related to the other two applications of the significant curvature points, namely, computer recognition of objects and sonar ranging of mobile robots.

Using the bounding contour of an object to recognize the object is a common practice in computer vision [52], [54] - [56], since the bounding contours are high in information content. For example, consider the contour shown in Fig. 4.8. It is rather simple for a person to recognize this contour as a bounding contour of a fighter airplane. To have a computer perform this recognition task, the following procedure can be followed. The significant curvature points are identified using the scheme developed here; they are shown in Fig. 4.9. These feature points are then used for the recognition which is a simple matching process: The set of the extracted feature points is matched to several previously generated feature point sets which correspond to different models of airplanes. The result of this

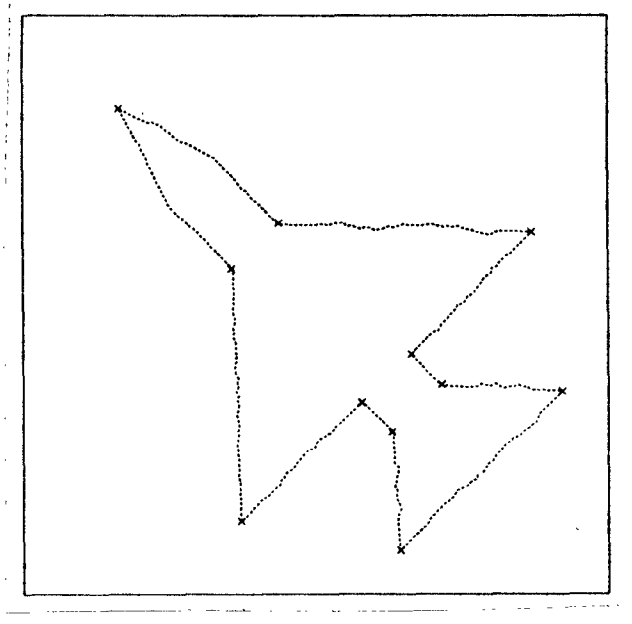


Figure 4.9: Fighter airplane contour and the extracted significant curvature points.

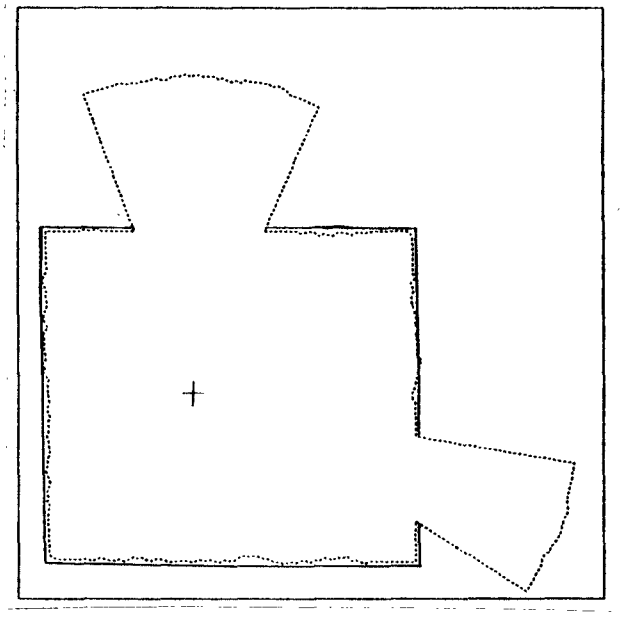


Figure 4.10: Sonar range contour.

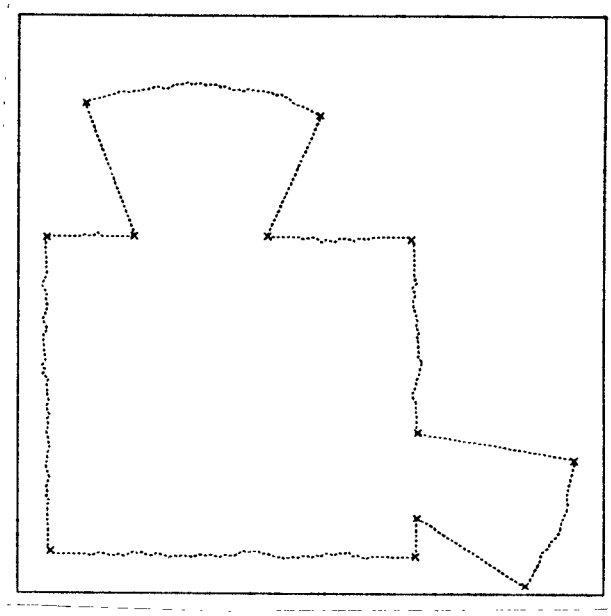


Figure 4.11: Sonar range contour and the extracted significant curvature points.

automated recognition process is the best matched model or the list of models registered in the order of closeness of matching. The operations, such as shift, rotation, enlargement and reduction, required in the matching process can be conducted easily due to the relatively small number of the feature points.

Now we consider the contour shown in Fig. 4.10 which is of the shape of indoor sonar range data taken from a mobile robot [63], [64]. In Fig. 4.10, the dotted contour is for the sonar range data, the solid lines are for the outline of the room which has two doors opening to the north and the east, and the cross mark indicates the position of the mobile robot. For the robot to perform autonomous navigation, the significant curvature points, which are extracted with our scheme and are shown in Fig. 4.11, of the range data are sufficient, since from these feature points the outline of the room and the sizes and the locations of the doorways can be determined.

4.5 Summary and Conclusions

In this chapter, we developed a new scheme to extract significant curvature points on a planar curve. The idea for this new scheme is similar to those used for generating the three-component image model. We have introduced the concept of the stressed curve which is analogous to the stressed image. The stressed curve is generated by solving a minimization problem recursively; the extraction of the significant curvature points is combined with the generation of the stressed curve to simplify the algorithm. The minimization problem has an interesting interpretation in which the objective function is the total potential energy of a mechanical structure; this interpretation motivates the development of the updating strategy of the parameter set. The minimization problem is solved by using Gauss-Seidel iteration; the resulting algorithm is rather simple and is local since the iteration for each point of the curve uses at most four of its nearest neighbors.

Examples and some of the applications of our scheme are described along with a comparison with the curvature primal sketch scheme. In this comparison, our scheme offers efficient and accurate extraction of significant curvature points. For the curvature primal sketch scheme, we have demonstrated the behavior of LGO and the necessity of multiple LGOs and a coarse-to-fine tracking procedure. We have also investigated the computational complexities incurred in this case by our scheme and the curvature primal sketch scheme; our scheme gives rise to much less complexity.

The concept of the scheme developed in this chapter can be also applied to 1-D signals such as LPC (or LSP) parameters of speech signals to extract the significant changes of the parameters in the presence of noise. Such extraction

of the parameter changes may lead to performance improvement for low bit rate speech coders [70], [71].

Chapter 5

Summary and Future Work

5.1 Summary and Conclusions

In this dissertation, we have introduced an approach for image coding based on the ideas of two categories of image compression techniques, namely, waveform coding and feature-based coding techniques. This approach takes the properties of the HVS into the consideration by decomposing the image signal into components; these components are significant for the formation of the visual perception. No distortion is introduced in the decomposition process. The quantization and coding of the image components are performed based on the information-theoretic principles such that the distortions are incurred at the minimum level for a given design bit rate. The advantages of this approach are demonstrated by the superior subjective and objective performances of the various image coding systems designed following this approach. The above basically summarizes the essence of this dissertation. We have also summarized every portions of this dissertation at the ends of the previous chapters. Nevertheless, a number of observations are appropriate to be presented here in the following.

(i) The binocular nature of the HVS provides us with an important apparatus for the understanding of the relationship of the properties of image signals and the corresponding visual perception.

(ii) To specify an edge in the image coding situations, we need the location of the edge plus the intensity variation across the edge; this is in contrast to the practice in computer vision where only the location of the edge is normally required. Therefore, some of the traditional edge extraction scheme (like LGO) can not directly applied in the image coding practice.

(iii) Although the ADCT coding schemes in this dissertation are developed primarily for the coding of the smooth and texture components and for the constructions of the corresponding overall coding systems, these schemes (1C-ADCT-HC and -AC) can be applied directly to images and offer comparable and higher PSNR performances at much less complexity than that of several state-of-the-art image coding systems.

(iv) For image coding, the three-component decomposition not only leads to subjective performance improvement, but also provides us with a better image model in an objective sense, since the analytical and simulation PSNRs match more closely for 3C-ADCT-HC than for 1C-ADCT-HC.

(v) The concept of the stressed image and the corresponding minimization problem and mechanical structure is quite important in its own right. This concept is proved to be useful in other image processing

applications by the development of the new planar curve representation scheme. The generation of the stressed image (or the stressed curve) is basically a non-uniform smoothing process.

5.2 Future Work

Since the algorithms for generating the stressed image and the primary image are local at each pixel of the image and the generation of the resultant three-components are pixel-wise subtractions, the procedure for the three-component image model can be made very fast through parallel computations; this will be a direction for the future work. The quantization method used for 2D DCT coefficients and subbands are scalar and very low in complexity. At the cost of some additional complexity, performance improvements can be obtained by using more powerful quantization methods. Work is currently underway to use the entropy-coded trellis-coded quantization method for transform coefficients; this quantization method is developed based on the entropy-constrained trellis-coded quantization of [50].

Appendix A

An Algorithm For Solving EMM Problem

In this appendix, we develop an algorithm solving the EMM problem (2.14) or equivalently (2.15). A typical equation in (2.15) is

$$\frac{\partial}{\partial y_{i,j}} E(\mathcal{Y}, \mathcal{X}, \Lambda) = 0, \tag{A.1}$$

which is

$$\begin{aligned}
&(\lambda^1_{i,j} + 4\lambda^2_{i,j} + 4\lambda^3_{i,j} + \lambda^2_{i,j-1} + \lambda^2_{i,j+1} + \lambda^3_{i-1,j} + \lambda^3_{i+1,j}) y_{i,j} \\
&\qquad -2(\lambda^2_{i,j} + \lambda^2_{i,j-1}) y_{i,j-1} - 2(\lambda^2_{i,j} + \lambda^2_{i,j+1}) y_{i,j+1} \\
&\qquad -2(\lambda^3_{i,j} + \lambda^3_{i-1,j}) y_{i-1,j} - 2(\lambda^3_{i,j} + \lambda^3_{i+1,j}) y_{i+1,j} \\
&+\lambda^2_{i,j-1} y_{i,j-2} + \lambda^2_{i,j+1} y_{i,j+2} + \lambda^3_{i-1,j} y_{i-2,j} + \lambda^3_{i+1,j} y_{i+2,j} \quad = \lambda^1_{i,j} x_{i,j}.
\end{aligned} \tag{A.2}$$

Equations in (2.15) related to boundary pixels are of forms different from the above one, since $C^r_{i,j}$ and $C^r_{i,j}$ may assume zeros on boundary pixels (ref. (2.9), (2.10)). We will consider iterative way of solving (2.15) for the relatively simple

structure of iterative algorithms and the applicability of Multi-Grid method, which gives fast convergence rate.

The most convenient iteration method to solve systems of linear equations is Gauss-Seidel iteration which is often very effective [27]. We explain Gauss-Seidel iteration briefly in the following. Let us denote the coefficient of $y_{i,j}$ in equation (A.1) as $\ell_{i,j}$. Then, one step of Gauss-Seidel iteration is given by

$$y_{i,j} = \{\ell_{i,j} y_{i,j} - \frac{\partial}{\partial y_{i,j}} E(\mathcal{X}^s, \mathcal{X}, \Lambda)\}/\ell_{i,j}, \tag{A.3}$$

where the sign "=" means "is replaced by", and index (i,j) assumes each value once in the set $\{0,1,\ldots,M-1\} \times \{0,1,\ldots,M-1\}$ according to a certain order. Note that $\ell_{i,j}$ is greater than zero for our assumption that $\lambda^1_{i,j} > 0$, for all (i,j) (ref. (A.2)). The above iteration is consistent with (2.15), since that $\{y_{i,j}\}$ is the solution of (2.15) if and only if $\{y_{i,j}\}$ is a fixed point of the iteration. Moreover, if Gauss-Seidel iteration generates a sequence $\{y^0_{i,j}\}, \{y^1_{i,j}\}, \ldots$ which converges to $\{y^*_{i,j}\}$, then $\{y^*_{i,j}\}$ is the solution of (2.15). To show that this Gauss-Seidel iteration is convergent, we only need that the matrix, L, one typical row of which is expressed in (A.2), is positive definite based on Ostrowski's theorem [27, p. 125] [65, p. 109]. Thus, by Lemma 2.3.1, the above Gauss-Seidel iteration converges to the solution of (2.15) or (2.14).

We now describe the basic concept of Multi-Grid method [27], and then give the algorithm of the above Gauss-Seidel iteration in Multi-Grid structure. Let us write the systems of linear equations (2.15) using matrix L:

$$L[y_{i,j}] = [\lambda^1_{i,j} x_{i,j}], \tag{A.4}$$

which is also Equation (2.17). We denote the solution of this systems of linear equations by $[y^*_{i,j}]$, and the result after kth Gauss-Seidel iteration by $[y^k_{i,j}]$. We

then define the *error* after kth Gauss-Seidel iteration:

$$[e^k_{i,j}] = [y^k_{i,j}] - [y^*_{i,j}], \tag{A.5}$$

and the *defect* after kth Gauss-Seidel iteration:

$$[d^k_{i,j}] = L[y^k_{i,j}] - [\lambda^1_{i,j} x_{i,j}] = L[y^k_{i,j}] - L[y^*_{i,j}] = L[e^k_{i,j}]. \tag{A.6}$$

Note that error $[e^k_{i,j}]$ is the solution of a linear system similar to (A.4) with the right-hand-side changed to $[d^k_{i,j}]$, and then is the solution of the following problem,

$$\min_{\{e_{i,j}\}} E(\mathcal{E}, \mathcal{D}, \Lambda), \tag{A.7}$$

where $\mathcal{E} = \{(i, j, e_{i,j})\}$ and $\mathcal{D} = \{(i, j, d^k_{i,j}/\lambda^1_{i,j})\}$. Thus, instead of continuing Gauss-Seidel iterations after kth to reduce the magnitude of error vectors, we may solve the problem (A.6) for error $[e^k_{i,j}]$, and then obtain the solution $[y^*_{i,j}]$ by subtracting $[e^k_{i,j}]$ from $[y^k_{i,j}]$:

$$[y^*_{i,j}] = [y^k_{i,j}] - [e^k_{i,j}]. \tag{A.8}$$

This is the basic idea of Multi-Grid method, which is based on the observation that the high-frequency component of $\{e^k_{i,j}\}$ dies away rather fast in the early iterations, i.e., error $\{e^k_{i,j}\}$ are effectively smoothed by iterations, and the convergence is prolonged basically by the low-frequency component of $\{e^k_{i,j}\}$ [27]. Now we suppose that k is large enough such that $\{e^k_{i,j}\}$ is relatively smooth, and thus we can subsample $\{e^k_{i,j}\}$ without loss of information, i.e., we can solve the system of linear equations (A.6) on a coarse grid, e.g., $(2i, 2j)$ for $i, j = 0, 1, \ldots, (M/2) - 1$ as compared with the original (i, j) for $i, j = 0, 1, \ldots, M - 1$. This process of subsampling is called *restriction*. We denote the error on a coarse grid, say, $(2i, 2j)$ for $i, j = 0, 1, \ldots, (M/2) - 1$, by

$\{\tilde{e}^k_{2i,2j}\}$, and the corresponding defect by $\{\tilde{d}^k_{2i,2j}\}$. We then solve the following system of linear equations, which is of smaller size than before,

$$L'[\tilde{e}^k_{2i,2j}] = [\tilde{d}^k_{2i,2j}], \tag{A.9}$$

where L' is same as L except of smaller size. To get $\{e^k_{i,j}\}$ from $\{\tilde{e}^k_{2i,2j}\}$ is an interpolation process which is inverse to the restriction, and is called *prolongation.* Notice that we can apply the above process: Gauss-Seidel iterations, restriction, solving a linear system of smaller size, and prolongation, again to solve (A.9). Actually, this process can be repeated hierarchically, and the resulting hierarchical process is called a Multi-Grid process of Gauss-Seidel iteration.

We summarize this subsection by outlining the algorithm of this Multi-Grid process of Gauss-Seidel iteration for solving (2.14) as follows.

Multi-Grid iteration for $L[y_{i,j}] = [\lambda^1_{i,j} x_{i,j}]$:

Start: $[y^0_{i,j}]$;

Gauss-Seidel iteration k times: $[y^k_{i,j}]$;

Compute defect $[d^k_{i,j}] = L[y^k_{i,j}] - [\lambda^1_{i,j} x_{i,j}]$;

Restriction: $[\tilde{d}^k_{i,j}]$;

Multi-Grid iteration for $L'[\tilde{e}^s_{i,j}] = [\tilde{d}^k_{i,j}]$ j times: $[\overline{e}^k_{i,j}]$;

Prolongation: $[e^k_{i,j}]$;

Updating: $[y_{i,j}] = [y^k_{i,j}] - [e^k_{i,j}]$;

Gauss-Seidel iteration additional ℓ times on $[y_{i,j}]$;

End.

Appendix B

An Algorithm For Generating Primary Images

We generate the primary image $\mathcal{P}$ based on the observations on the HVS. More precisely, we use the mechanism described by (2.3) to do the formation of $\mathcal{P}$ from $\mathcal{B}_T(\mathcal{X}^s)$, i.e., we define image $\mathcal{P}$ to be the solution of the following minimization problem,

$$\min_{\{x_{i,j}\}} V_{\mathcal{X}}, \quad \text{subject to } \mathcal{X} \cap \mathcal{B}_T(\mathcal{X}^s) = \mathcal{B}_T(\mathcal{X}^s), \tag{B.1}$$

with some modifications described later. Similarly as in Appendix A and in Subsection 2.3.2, we can conclude that there is a unique solution of (B.1) which is also the unique solution of the system of linear equations,

$$\nabla_{\{x_{i,j}\}} V_{\mathcal{X}} \equiv 0, \tag{B.2}$$

where the partial derivations are taken only with respect to those $x_{i,j}$ such that $(i, j, x_{i,j}) \notin \mathcal{B}_T(\mathcal{X}^s)$, for $\mathcal{B}_T(\mathcal{X}^s) \neq \emptyset$. A typical equation in the above linear system is

$$\frac{\partial}{\partial x_{i,j}} V_{\mathcal{X}} = 0, \tag{B.3}$$

where $(i, j, x_{i,j}) \notin \mathcal{B}_T(\mathcal{X}^s)$, and thus the generic form for one step of Gauss-Seidel iteration is

$$x_{i,j} = \frac{1}{4}(x_{i,j-1} + x_{i,j+1} + x_{i-1,j} + x_{i+1,j}), \tag{B.4}$$

which is simply the average of four neighboring pixels' intensity values. It is straightforward to derive the formulas for boundary pixels, which are slightly different from the above one. This Gauss-Seidel iteration is convergent, and converges to the solution of (B.1) (ref. Appendix A).

The Multi-Grid method is also applied to speed up the convergence, while the Multi-Grid method used here is different from the one described in Appendix A where we solve for error (ref. (A.5), (A.6)). We can not apply here the structure of the Multi-Grid method in Appendix A simply because that the form of the linear system (B.2) is image-dependent, and thus it is very difficult to determine the forms of linear systems at every level of grid, or equivalently, to determine the forms of L' (ref. (A.9)). Instead of solving for error, we solve the system (B.2) at every level of grid. Note that the formula of Gauss-Seidel iteration for each pixel's intensity value is the average of four neighboring pixels' intensity values as pointed out above, and the intensity values of the pixels in $\mathcal{B}_T(\mathcal{X}^s)$ are fixed in all iterations. This algorithm can be easily implemented at each level of grid. Let us consider a coarser grid $(2i, 2j)$, for $i, j = 0, 1, \ldots, (M/2) - 1$ on the grid (i, j), for $i, j = 0, 1, \ldots, M - 1$, a portion of which is shown in Figure B.1, where the pixels of the coarser grid are indicated by small circles. The initial intensity values, $x'_{2i,2j}$, on the coarser grid are the weighted average of the intensity value at the same location and the intensity values of all the neighboring pixels on the denser grid. The typical formula for this weighted average at $(2i, 2j)$ is

$$x'_{2i,2j} = \frac{1}{16}(4x_{2i,2j} + 2x_{2i,2j-1} + 2x_{2i,2j+1} + 2x_{2i-1,2j} + 2x_{2i+1,2j}$$

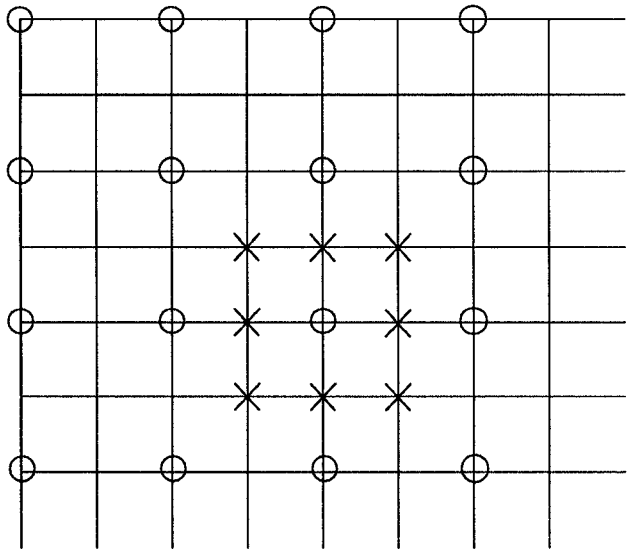


Figure B.1: Multi-Grid method.

$$+x_{2i-1,2j-1} + x_{2i-1,2j+1} + x_{2i+1,2j-1} + x_{2i+1,2j+1}). \tag{B.5}$$

We then define the set of fixed pixels at the coarser grid by

$$\mathcal{B}_T(\mathcal{X}^s)' \equiv \{(2i, 2j, x'_{2i,2j}) : (2i, 2j, .)$$
$$\text{or one of its eight neighboring pixels} \in \mathcal{B}_T(\mathcal{X}^s)\}. \tag{B.6}$$

For example, in Figure 12, the eight neighboring pixels of one pixel on the coarser grid are marked out with crosses. If one of these neighboring pixels or the pixel itself is in $\mathcal{B}_T(\mathcal{X}^s)$, then this pixel is in $\mathcal{B}_T(\mathcal{X}^s)'$. On the coarser grid we do similar averaging (B.4) with the new fixed pixel set $\mathcal{B}_T(\mathcal{X}^s)'$. The above process can be applied to even coarser grids in the same fashion as above. To recover the intensity values on a denser grid from a coarse grid is straightforward. Using the above notation, $x_{2i,2j} = x'_{2i,2j}$, for $i, j = 0, 1, \ldots, (M/2) - 1$; while for the other pixels not in $\mathcal{B}_T(\mathcal{X}^s)$, their intensity values are equal to the average of the intensity values on their neighboring pixels on coarse grid.

The algorithm starts from an initial image which has constant intensity values for all pixels. This constant intensity value is equal to the average value of the intensity values of the original image. After several times of averaging (B.4), we go to the coarser grid as above, and do averaging on this coarser grid. We repeat this process again until all the pixels in the current grid are fixed ones. We then do recovery all the way back to the original grid. After every recovering from a coarser grid to a denser grid, we also apply the averaging several times. This whole cycle, from the original grid to a very coarse grid and then back, can be repeated until we obtain satisfactory result.

Note that we are not required to solve the problem (B.1) accurately for generating primary image $\mathcal{P}$ as long as that $\mathcal{P}$ contains strong edges represented by $\mathcal{B}_T(\mathcal{X}^s)$, and $\mathcal{S} = \mathcal{X}^s \ominus \mathcal{P}$ is smooth.

Appendix C

EMM – a Space-Variant Filter

In this Appendix, we show that the EMM problem can be described as a space-variant filtering process. The input-output relation of this filter is determined by the minimization problem (2.14), and will be discussed in the following.

For simplicity, we consider the continuous form of (2.14). Let $x(u,v)$ be a continuous function of two variables u, v defined on a simply-connected closed domain [66] $G \subset \mathbb{R}^2$ with a piecewise smooth boundary C. We want to solve the following minimization problem,

$$\min_y J_x(y) \equiv \int\int_G \{\lambda_1(u,v)[x(u,v)-y(u,v)]^2 + \lambda_2(u,v)[y_{uu}(u,v)]^2 + \lambda_3(u,v)[y_{vv}(u,v)]^2\}\, dudv, \qquad \text{(C.1)}$$

where $\lambda_i(u,v) \geq 0$, $i = 1,2,3$, are functions of u, v on G having continuous second-order partial derivatives, $y(u,v)$ is an admissible function defined on G having continuous fourth-order partial derivatives with y, $\partial y/\partial u$, $\partial y/\partial v$ vanishing on the boundary C and $y_{uu}(u,v)$ and $y_{vv}(u,v)$ are second-order partial derivatives of y with respect to u and v, respectively. In (C.1), $J_x(y)$ is used to denote the objective function. All admissible functions form a linear space,

denoted by $\mathcal{D}(G)$ endowed by a norm $||y||$ defined as

$$||y|| \equiv \max_{(u,v)\in G} |y(u,v)| + \max_{(u,v)\in G} |y_{uu}(u,v)| + \max_{(u,v)\in G} |y_{vv}(u,v)|. \tag{C.2}$$

Thus $\mathcal{D}(G)$ becomes a normed linear space [66].

We say J_x has a minimum at $y^* \in \mathcal{D}(G)$ if and only if $J(y) - J(y^*) \geq 0$ for all $y \in \mathcal{D}(G)$. Then, the necessary and sufficient condition for y^* to be a minimum of J_x is that [66]

$$\int\int_G \{\lambda_1(y^* - x)h + \lambda_2 y^*_{uu} h_{uu} + \lambda_3 y^*_{vv} h_{vv}\}\, dudv = 0, \tag{C.3}$$

for all $h \in \mathcal{D}(G)$. By the Green's theorem [67], (C.3) reduces to [66]

$$\lambda_1(u,v)y^*(u,v) + (\lambda_2 y^*_{uu})_{uu}(u,v) + (\lambda_3 y^*_{vv})_{vv}(u,v) = \lambda_1(u,v)x(u,v), \tag{C.4}$$

for $(u,v) \in G$. Equation (C.4) is known as an *Euler's equation* [66].

Now we show that at all $(u,v) \in G$ satisfying $\lambda_1(u,v) > 0$, the solution of the Euler's equation is unique. Let y_1 and y_2 be two solutions of the Euler's equation, then $y_1 - y_2$ is a solution of the Euler's equation with $x \equiv 0$, and thus $y_1 - y_2$ is a minimum point of J_0. Then $y_1(u,v) - y_2(u,v) = 0$, for every (u,v) such that $\lambda_1(u,v) > 0$, for otherwise $J_0(\epsilon(y_1 - y_2)) < J_0(y_1 - y_2)$ for any $0 < \epsilon < 1$.

Now we investigate the frequency domain behavior of the input-output relation governed by the above Euler's equation. Let (u_1, v_1) be an interior point in G, and A be an open neighborhood containing the point (u_1, v_1). To further simplify the problem, we assume that the parameter functions, $\lambda_1 > 0$, λ_2, λ_3, are constants on A (certainly when A is a small neighborhood, this is a reasonable assumption). Then the above Euler's equation becomes

$$\lambda_1(u_1,v_1)y^*(u,v) + \lambda_2(u_1,v_1)y^*_{uuuu}(u,v) + \lambda_3(u_1,v_1)y^*_{vvvv}(u,v) = \lambda_1(u_1,v_1)x(u,v) \tag{C.5}$$

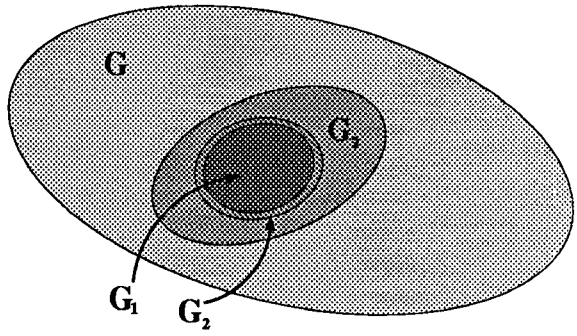


Figure C.1: Relationship of the subdomains $G_1 \subset G_2 \subset G_3 \subset G$.

on A. Equation (C.5) is a linear partial differential equation which describes the input-output relationship of a linear system with input $x(u,v)$, output $y^*(u,v)$ and frequency response $H(w_u, w_v)$ given by

$$H(w_u, w_v) = \frac{1}{1 + \frac{\lambda_2(u_1,v_1)}{\lambda_1(u_1,v_1)}\omega_u^4 + \frac{\lambda_3(u_1,v_1)}{\lambda_1(u_1,v_1)}\omega_v^4} \tag{C.6}$$

which clearly describes a 2-D low-pass filtering operation. The cutoff frequencies of $H(w_u, w_v)$ are controlled by the ratios $\frac{\lambda_2(u_1,v_1)}{\lambda_1(u_1,v_1)}$ and $\frac{\lambda_3(u_1,v_1)}{\lambda_1(u_1,v_1)}$. Larger values of $\lambda_1(u_1, v_1)$ give rise to higher cutoff frequencies in u and v, while larger values of $\lambda_2(u_1, v_1)$ and $\lambda_3(u_1, v_1)$ lead to lower cutoff frequencies in u and v directions, respectively.

Note that we have assumed that all admissible functions and their first-order partial derivatives vanish on the boundary C, while in the original case (2.14) we do not have any such constraint. This difficulty can be avoided by solving a slightly modified problem in which the boundary constraints are removed. Let us consider some subdomains of G: $G_1 \subset G_2 \subset G_3 \subset G$. We illustrate the relations of these submains and G in Fig. C.1.

Let $\lambda_1^{G_2} > 0, \lambda_2^{G_2} \geq 0, \lambda_3^{G_2} \geq 0$ be functions having continuous second order

partial derivatives defined on G_2. We derive parameter functions λ_i^*, $i = 1, 2, 3$, on G from $\lambda_i^{G_2}$, $i = 1, 2, 3$, such that

$$\lambda_i^*(u, v) = \lambda_i^{G_2}(u, v), \text{ for } (u, v) \in G_1$$
$$\lambda_i^*(u, v) = 0, \text{ for } (u, v) \in G \backslash G_2. \tag{C.7}$$

This can be done by using a so-called bump function f [68], which has the following property:

1. All mixed partial derivatives, of all orders, of f exist and are continuous at every interior point of G.
2. $0 \leq f \leq 1$ on G.
3. $f = 1$ on G_1.
4. $f = 0$ on $\overline{G \backslash G_2}$.

Then the functions $\lambda_i^* \equiv f\lambda_i^{G_2}$, $i = 1, 2, 3$, on G_2; $\lambda_i^* \equiv 0$ on $G \backslash G_2$, satisfy the requirements in (C.7).

Similarly, we define a function x^* on G, such that $x^*(u, v) = x^{G_2}(u, v)$ on G_1, and $x^*(u, v) = 0$ on $G \backslash G_2$, where x^{G_2} is a given data function on G_2.

We consider the following two problems similar to (C.1) with the above parameter functions λ_i^*, $i = 1, 2, 3$, and the data function x^*. The first one is on G:

$$\begin{aligned} \min_y J_{x^*}^1(y) &\equiv \int\int_G \{\lambda_1^*[y - x^*]^2 + \lambda_2^*[y_{uu}]^2 + \lambda_3^*[y_{vv}]^2\}\, dudv \\ &= \int\int_{G_2} \{\lambda_1^*[y - x^*]^2 + \lambda_2^*[y_{uu}]^2 + \lambda_3^*[y_{vv}]^2\}\, dudv, \end{aligned} \tag{C.8}$$

with the set of admissible functions as before, i.e., $\mathcal{D}(G)$; the other one on G_3:

$$\min_y J_{x^*}^2(y) \equiv \int\int_{G_3} \{\lambda_1^*[y - x^*]^2 + \lambda_2^*[y_{uu}]^2 + \lambda_3^*[y_{vv}]^2\}\, dudv$$

$$= \int\int_{G_2} \{\lambda_1^*[y - x^*]^2 + \lambda_2^*[y_{uu}]^2 + \lambda_3^*[y_{vv}]^2\}\, dudv, \tag{C.9}$$

with the set of admissible functions being $\mathcal{D}(G_3)$, except that no constraint on the boundary.

We claim that if $y^* \in \mathcal{D}(G)$ is a minimum function of $J^1_{x^*}$, then there is no function in $\mathcal{D}(G_3)$ not identical with y^* on the set: $G_s \equiv \{(u,v) : \lambda_1^*(u,v) > 0\}$ and giving smaller or equal value of the functional $J^2_{x^*}$ than $J^2_{x^*}(y^* \mid G_3)$, where $y^* \mid G_3$ denotes the restriction of y^* on G_3. (Note that $G_1 \subset G_s \subset G_2$.) Indeed, for otherwise we have a function y_1 on $\mathcal{D}(G_3)$, and $y_1 \not\equiv y^*$ on G_s, such that

$$J^2_{x^*}(y_1) \leq J^2_{x^*}(y^* \mid G_3). \tag{C.10}$$

We construct a function $y_2 \in \mathcal{D}(G)$ such that $y_2 \equiv y_1$ on G_2 and $y_2 \equiv 0$ on $G \backslash G_3$ by using a proper bump function. More specifically, let f be a bump function: $f = 1$ on G_2, and $f = 0$ on $\overline{G \backslash G_3}$, and $y_2 = f y_1$ on G_3, $y_2 = 0$ on $G \backslash G_3$. Then y_2, $\partial y_2/\partial u$, $\partial y_2/\partial v$, vanish on the boundary of G, and thus $y_2 \in \mathcal{D}(G)$. We now have

$$J^1_{x^*}(y_2) = J^2_{x^*}(y_1) \leq J^2_{x^*}(y^* \mid G_3) = J^1_{x^*}(y^*), \tag{C.11}$$

i.e. $J^1_{x^*}(y_2) - J^1_{x^*}(y^*) \leq 0$, a contradiction. Therefore, in the above sense, problem (C.8) can be substituted by problem (C.9).

Appendix D

Contour Prediction

The first-order prediction works in the following way. After the basic algorithm (with back-tracing) stops, we locate an additional pixel in the neighborhood of the end-pixel of the traced contour in the direction determined by the last two pixels. More precisely, for the traced contour, $(i^0, j^0, x^s_{i^0,j^0})$, $(i^1, j^1, x^s_{i^1,j^1})$, ..., $(i^{\ell-1}, j^{\ell-1}, x^s_{i^{\ell-1},j^{\ell-1}})$, the additional pixel, $(i^\ell, j^\ell, x^s_{i^\ell,j^\ell})$, is defined as:

$$i^\ell = i^{\ell-1} + (i^{\ell-1} - i^{\ell-2}); \;\; j^\ell = j^{\ell-1} + (j^{\ell-1} - j^{\ell-2}). \tag{D.1}$$

If this new contour, $(i^1, j^1, x^s_{i^1,j^1})$, ..., $(i^\ell, j^\ell, x^s_{i^\ell,j^\ell})$, satisfies condition (2.24), and by the above basic search procedure we can locate in the neighborhood of $(i^\ell, j^\ell, x^s_{i^\ell,j^\ell})$ a pixel other than $(i^{\ell-1}, j^{\ell-1}, x^s_{i^{\ell-1},j^{\ell-1}})$ in $\mathcal{B}_T(\mathcal{X}^s)$, we move the search center to this new pixel in $\mathcal{B}_T(\mathcal{X}^s)$; otherwise, the contour terminates at $(i^{\ell-1}, j^{\ell-1}, x^s_{i^{\ell-1},j^{\ell-1}})$. The first-order prediction is illustrated in Fig. D.1, where the end pixel and its previous pixel are indicated by a circle and a cross, respectively, and the predicted pixel is indicated by a square box.

If the first-order prediction fails, a second-order prediction can be used. Using the same notation as before, we define pixels $(i^\ell, j^\ell, x^s_{i^\ell,j^\ell})$ and $(i^{\ell+1}, j^{\ell+1}, x^s_{i^{\ell+1},j^{\ell+1}})$

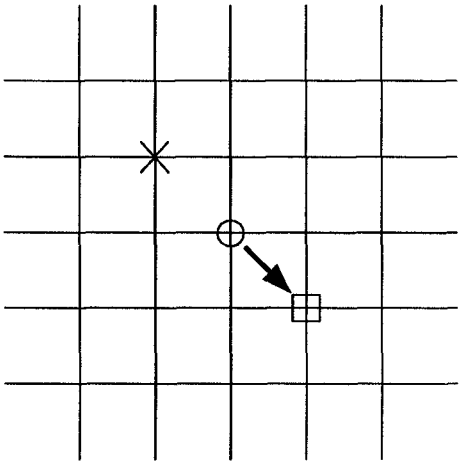


Figure D.1: The first-order prediction.

such that

$$i^{\ell} = i^{\ell-1} + (i^{\ell-2} - i^{\ell-3}); \;\; j^{\ell} = j^{\ell-1} + (j^{\ell-2} - j^{\ell-3});$$
$$i^{\ell+1} = i^{\ell} + (i^{\ell-1} - i^{\ell-2}); \;\; j^{\ell+1} = j^{\ell} + (j^{\ell-1} - j^{\ell-2}). \tag{D.2}$$

If this extended contour does not satisfy condition (2.24), we stop and end the contour at pixel $(i^{\ell-1}, j^{\ell-1}, x^s_{i^{\ell-1},j^{\ell-1}})$; otherwise, we continue as follows. If pixel $(i^{\ell+1}, j^{\ell+1}, x^s_{i^{\ell+1},j^{\ell+1}})$ is in $\mathcal{B}_T(\mathcal{X}^s)$, we move the search center to this pixel and continue; if not, we perform the basic search step at $(i^{\ell+1}, j^{\ell+1}, x^s_{i^{\ell+1},j^{\ell+1}})$, and if we successfully locate a pixel in $\mathcal{B}_T(\mathcal{X}^s)$, we then move the search center to this new pixel and continue. We illustrate the second-order prediction in Fig. D.2 with a similar legend as in Fig. D.1. This procedure can be extended to higher order predictions in a similar manner.

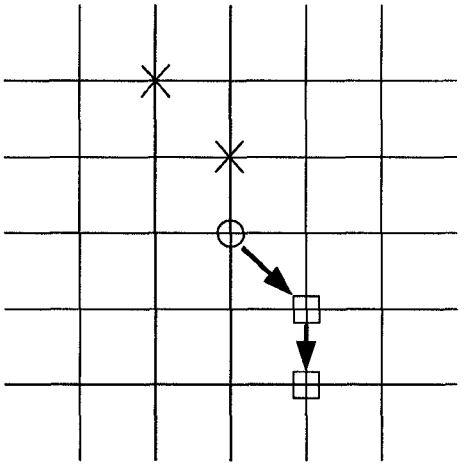


Figure D.2: The second-order prediction.

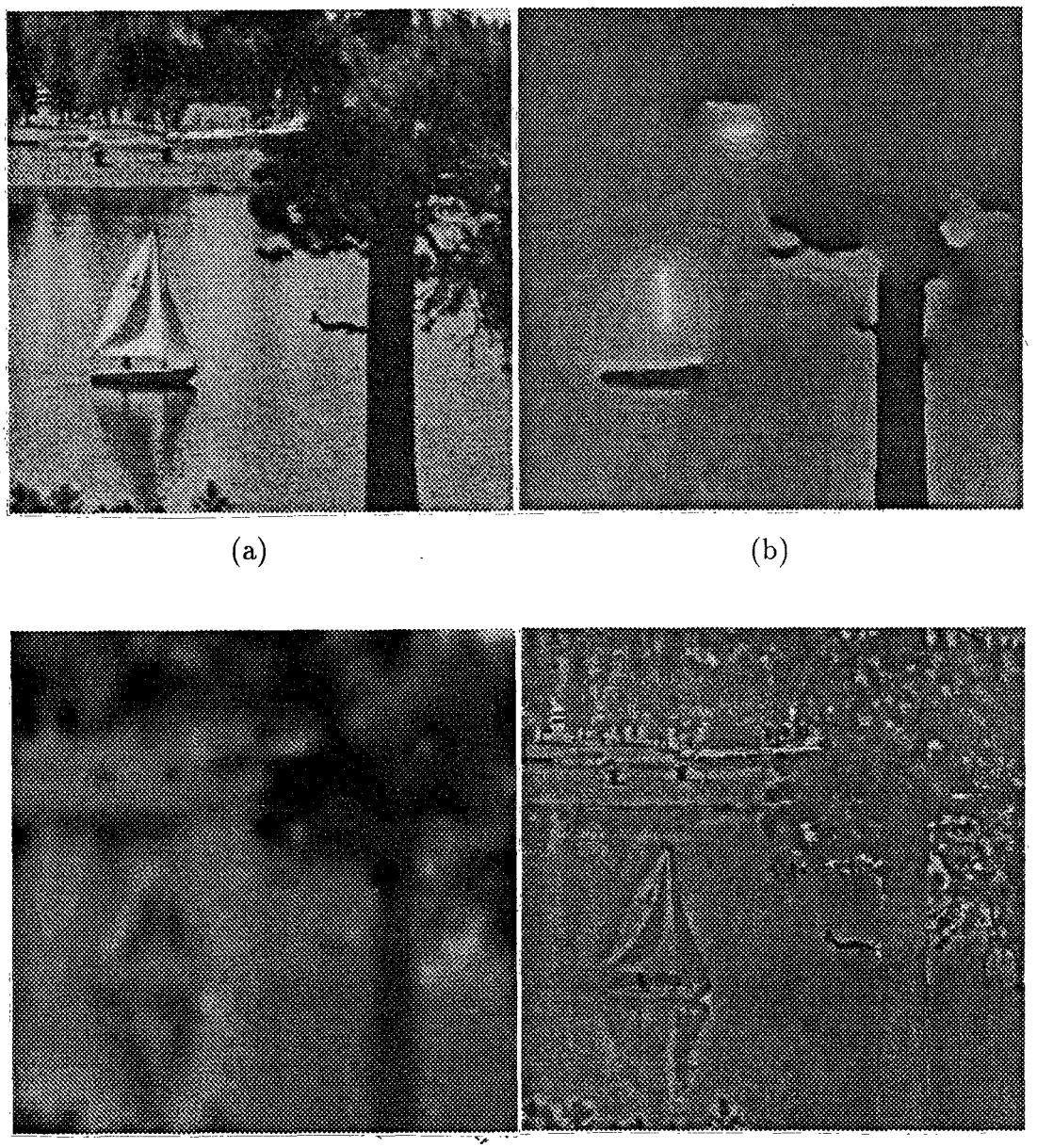
Appendix E

Examples of the three-component image model

Several examples of the three-component image model are included in this Appendix. The parameters for these examples are tabulated in the following.

Image name	N_ν	T	T_c	T_ℓ	d
Sailboat	50	16	32	16	3
Peppers	50	256	32	8	3
House	50	64	32	8	3
Airplane	50	128	32	16	3
Swanmaster	50	128	32	8	3
Kingfisher	50	32	32	8	3
Bison	50	16	64	8	3

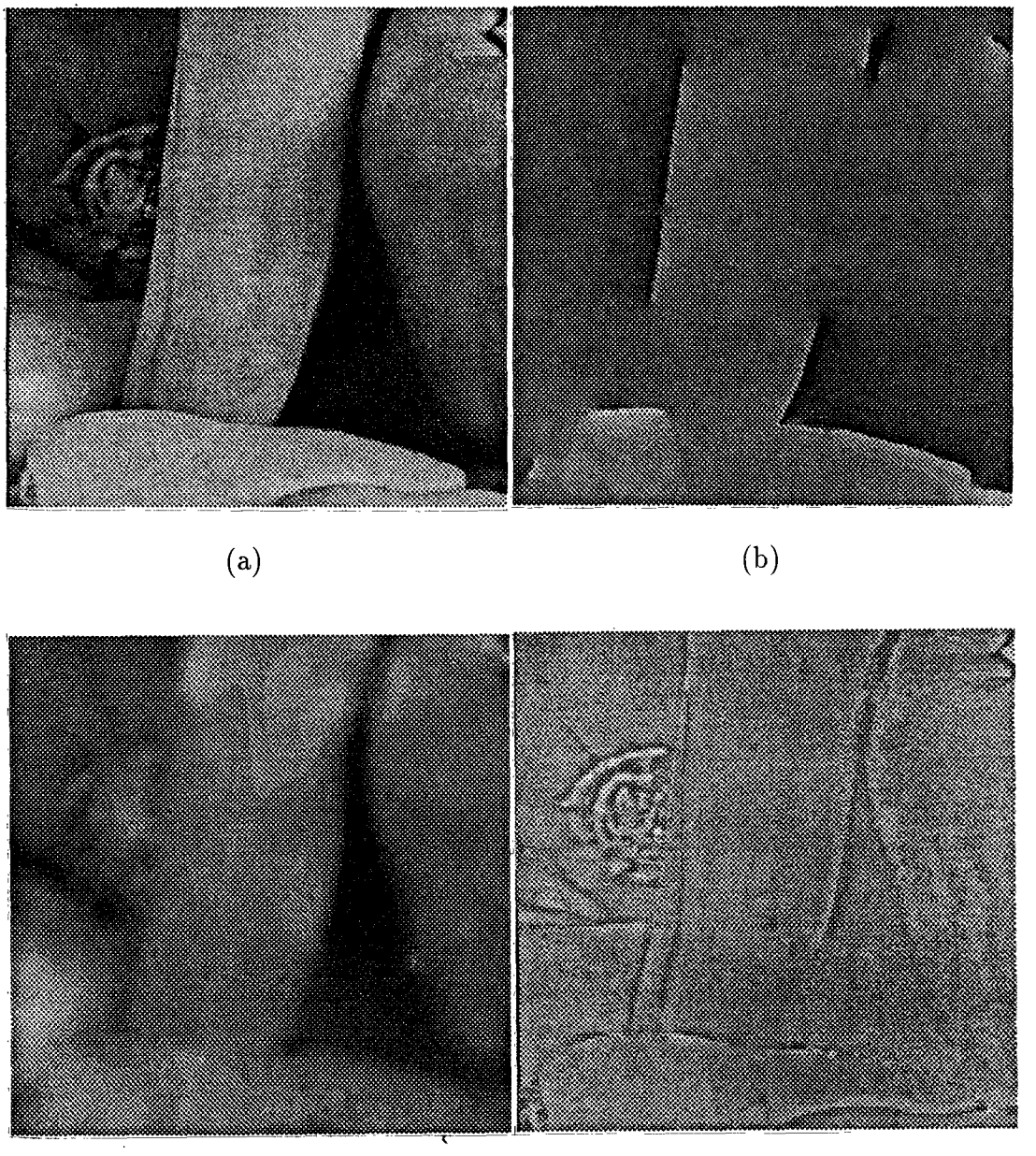
Table E.1: Parameters for the three-component image model examples.



(a) (b)

(c) (d)

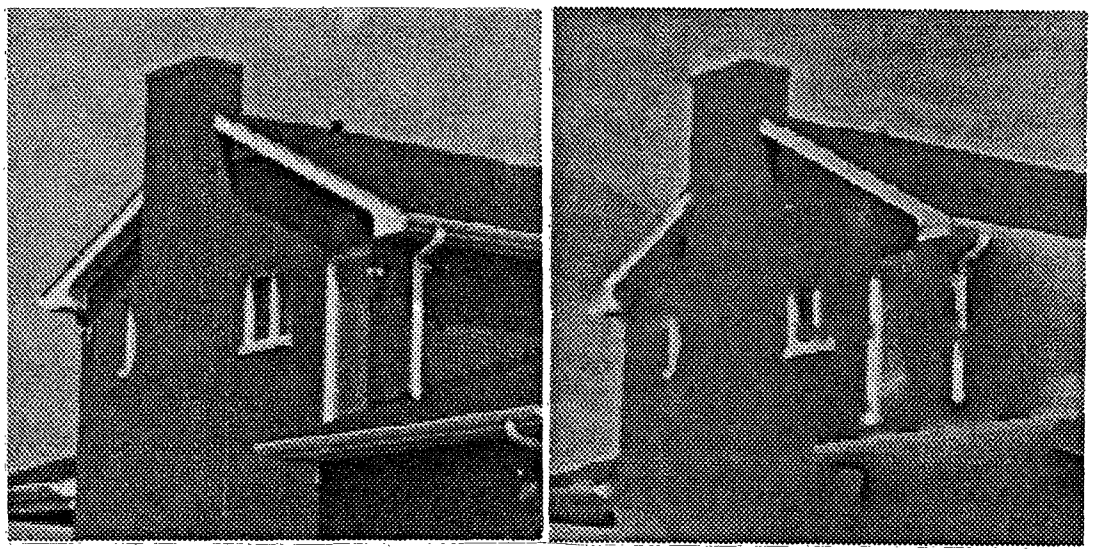
Figure E.1: (a) The original, (b) the primary component, (c) the smooth component and (d) the texture component associated with image Sailboat.



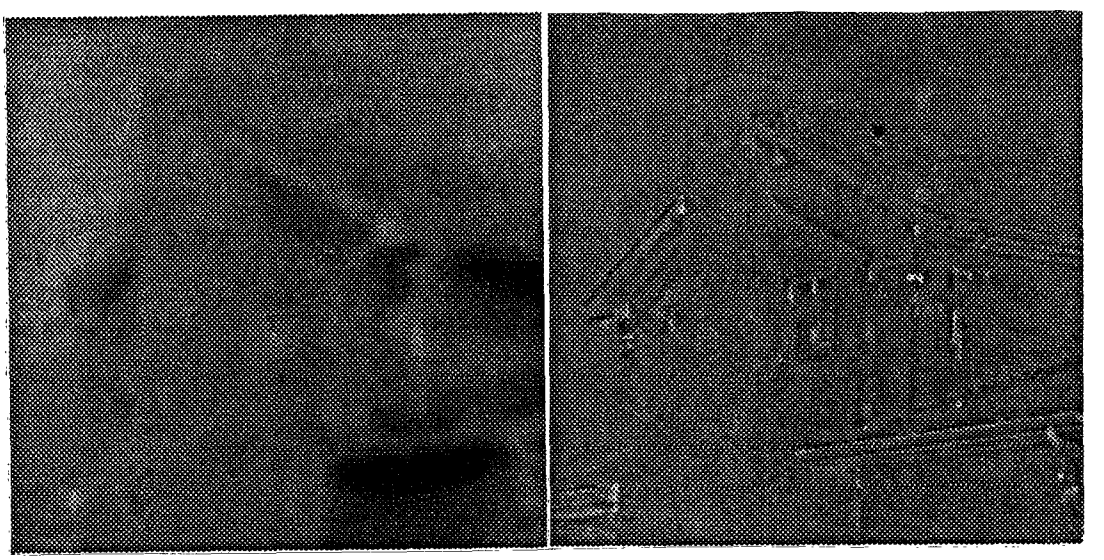
(a) (b)

(c) (d)

Figure E.2: (a) The original, (b) the primary component, (c) the smooth component and (d) the texture component associated with image Peppers.

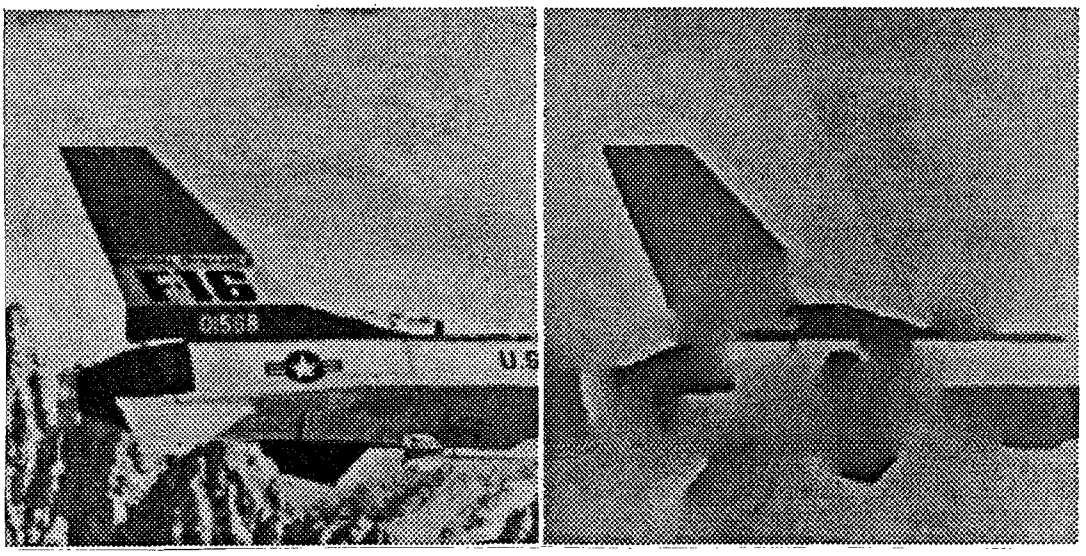


(a) (b)

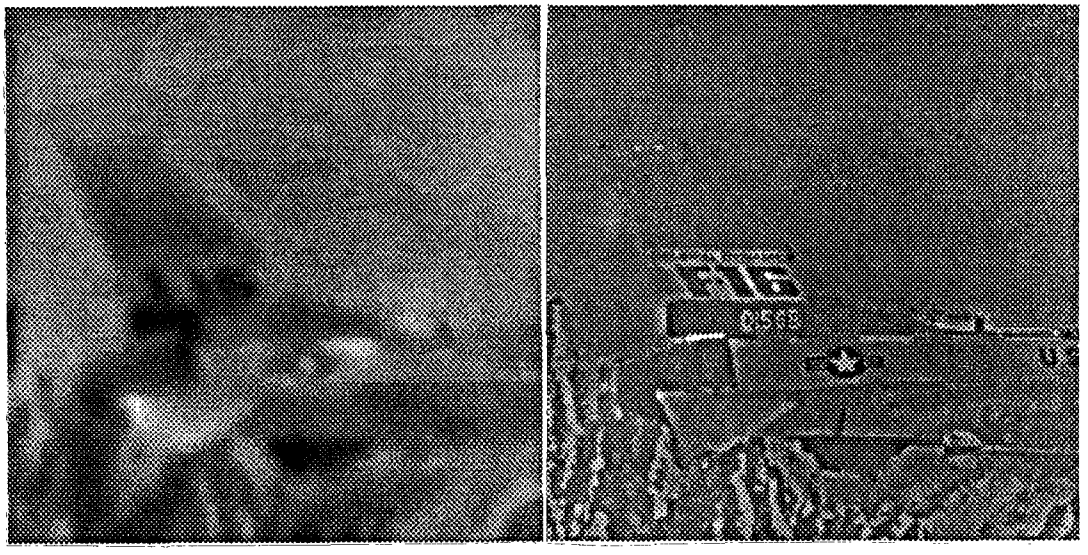


(c) (d)

Figure E.3: (a) The original, (b) the primary component, (c) the smooth component and (d) the texture component associated with image House.

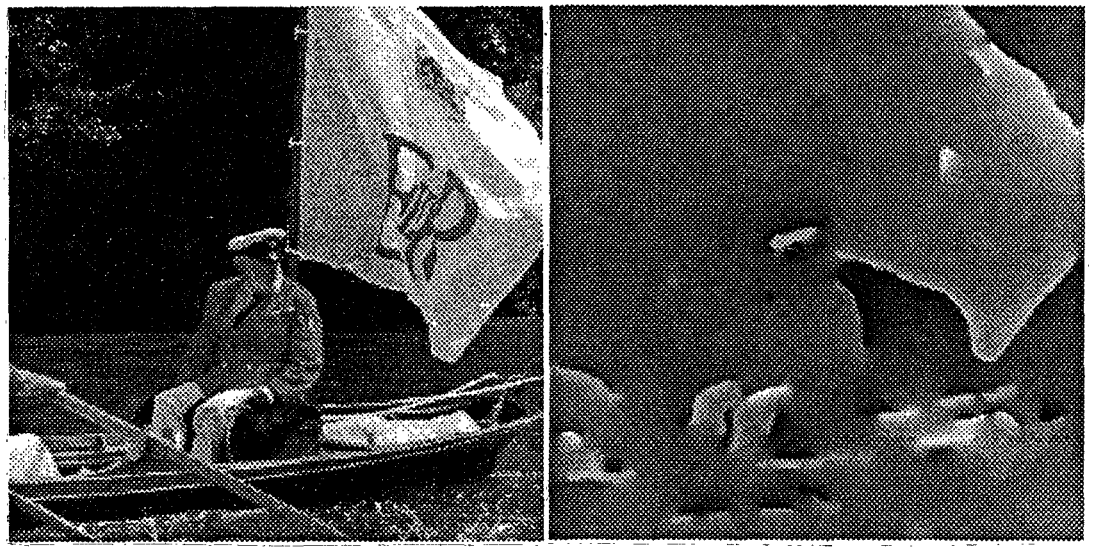


(a) (b)

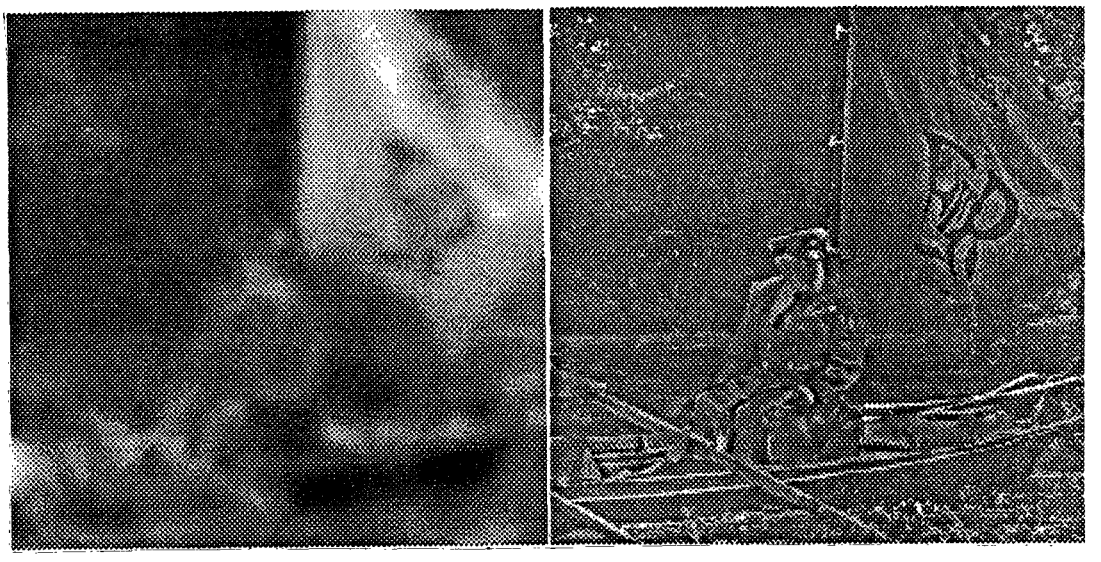


(c) (d)

Figure E.4: (a) The original, (b) the primary component, (c) the smooth component and (d) the texture component associated with image Airplane.



(a) (b)



(c) (d)

Figure E.5: (a) The original, (b) the primary component, (c) the smooth component and (d) the texture component associated with image Swanmaster.

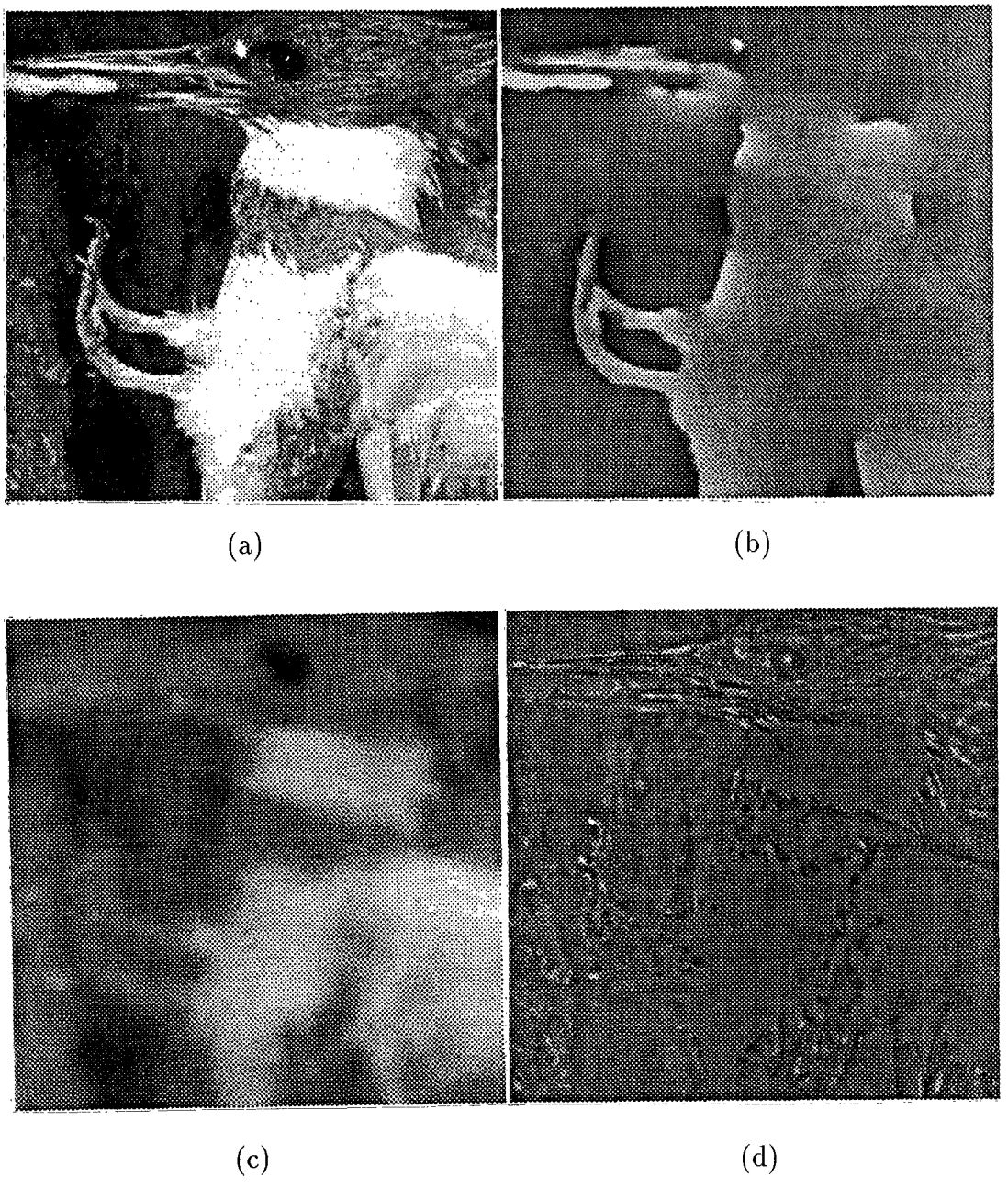
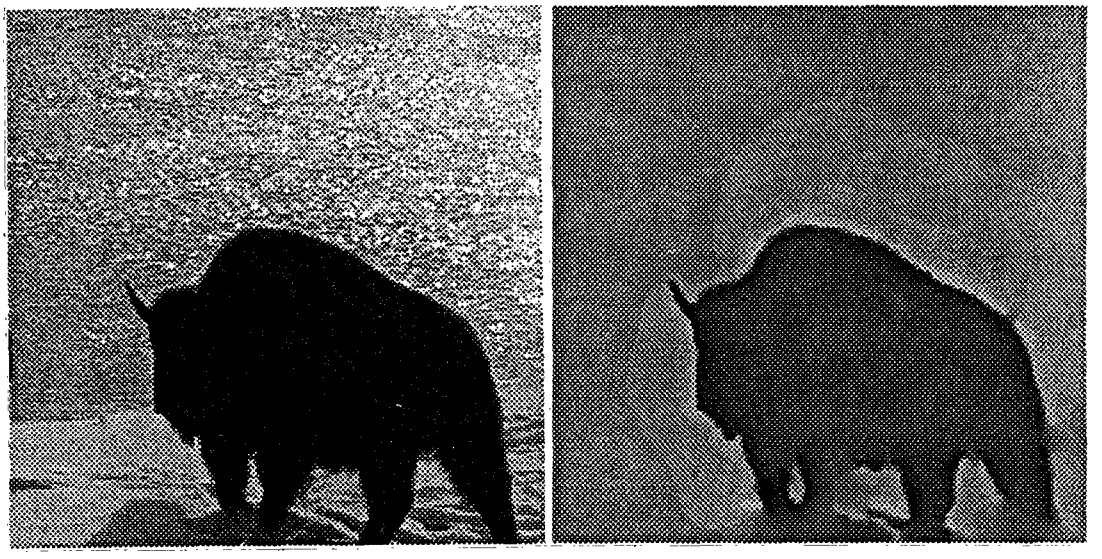
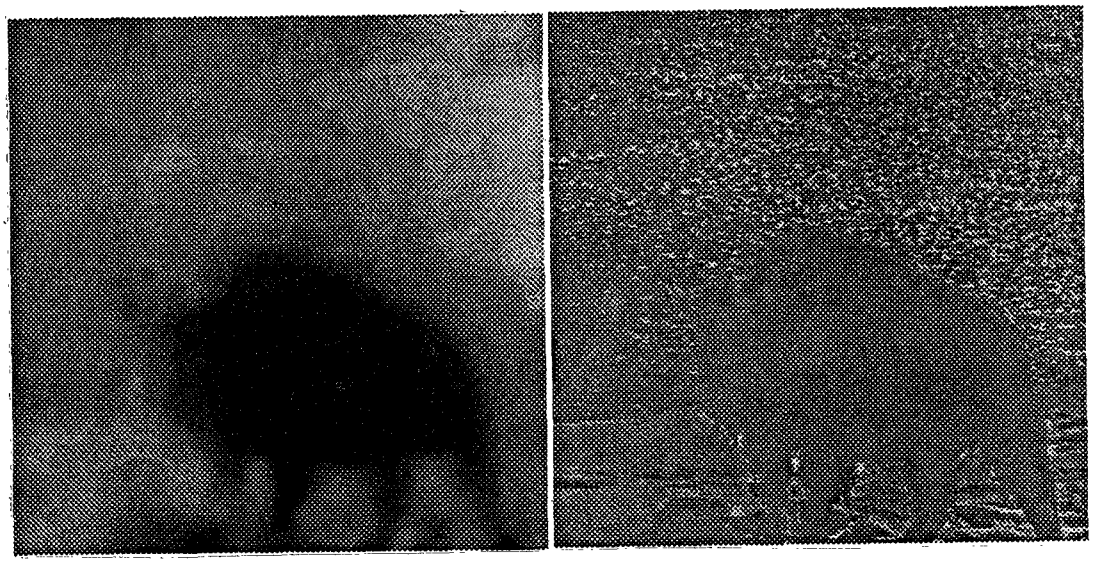


Figure E.6: (a) The original, (b) the primary component, (c) the smooth component and (d) the texture component associated with image Kingfisher.



(a) (b)



(c) (d)

Figure E.7: (a) The original, (b) the primary component, (c) the smooth component and (d) the texture component associated with image Bison.

Appendix F

Constant Variances of Quantization Errors

Consider the following function of bit rates r_o, $r_{0,0}$ and $r_{u,v}^k$, $k = 0, 1, \cdots, K-1$, $(u,v) \neq (0,0)$,

$$F(\mathbf{r}) \equiv D(\mathbf{r}) + \lambda R(\mathbf{r}), \tag{F.1}$$

where $D(\mathbf{r})$ is defined as D in (3.4) with $d_G(r)$ and $d_L(r)$ substituted by the approximations given in (3.8), $R(\mathbf{r})$ is defined as in (3.5), λ is the Lagrange multiplier introduced, $\mathbf{r}$ denotes the bit rates: $r_{0,0}$ and $r_{u,v}^k$, $k = 0, 1, \cdots, K-1$, $(u,v) \neq (0,0)$, collectively, and r_o is assumed to be a constant. Denote the optimal bit rates for problem (F.1) by $\mathbf{r}^*$: ${r_{0,0}}^*$ and ${r_{u,v}^k}^*$, $k = 0, 1, \cdots, K-1, (u,v) \neq (0,0)$. Then by the first-order necessary condition [26], we have

$$\nabla_{\mathbf{r}} F(\mathbf{r}^*) = 0; \qquad R(\mathbf{r}^*) = R^*, \tag{F.2}$$

where $\nabla_{\mathbf{r}} F(\mathbf{r}^*)$ is the gradient of F with respect to $\mathbf{r}$ at $\mathbf{r}^*$. The above conditions give a total of $K(L^2-1)+2$ equations in $K(L^2-1)+2$ variables comprising $\mathbf{r}^*$

and λ; by solving these equations, we have

$$
\begin{aligned}
{r_{0,0}}^* &= \frac{1}{2\ln 2}\ln\frac{\frac{\pi e \ln 2}{3}\sigma^2(0,0)}{\lambda^*}; \\
{r_{u,v}^k}^* &= \frac{1}{2\ln 2}\ln\frac{\frac{\pi e \ln 2}{3}\sigma_k^2(u,u)}{\lambda^*}, \qquad (u,v) = (0,1),(1,0); \\
{r_{u,v}^k}^* &= \frac{1}{2\ln 2}\ln\frac{\frac{e \ln 2}{3}\sigma_k^2(u,u)}{\lambda^*}, \qquad (u,v) \neq (0,0),(0,1),(1,0),
\end{aligned}
\tag{F.3}
$$

$$
\begin{aligned}
\lambda^* = \exp\{2\ln 2(-r_{st}^* + r_o + \frac{1}{2\ln 2 L^2}\{\ln(\frac{\pi e \ln 2}{3}\sigma^2(0,0)) + \\
\frac{1}{K}\sum_{k=0}^{K-1}[2\ln(\frac{\pi e \ln 2}{3}\sigma_k(0,1)\sigma_k(1,0)) + \\
\sum_{(u,v)\neq(0,0),(0,1),(1,0)} \ln(\frac{e \ln 2}{3}\sigma_k^2(u,v))]\})\},
\end{aligned}
\tag{F.4}
$$

and

$$
\epsilon_k^2(u,v) = \frac{\lambda^*}{2\ln 2}, \tag{F.5}
$$

for $k = 0,1,\cdots,K-1, u,v = 0,1,\cdots,L-1$, where r_{st}^* is the design rate for smooth and texture components.

References

[1] A. N. Netravali and B. G. Haskell, *Digital pictures, representation and compression*, Plenum Press, New York, 1988.

[2] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, pp. 30-44, April 1991.

[3] D. Le Gall, "MPEG: a video compression standard for multimedia applications," *Commun. ACM*, vol. 34, pp. 46-58, April 1991.

[4] A. Habibi and P. A. Wintz, "Image coding by linear transformation and block quantization," *IEEE Trans. Commun. Tech.*, vol. COM-19, pp. 50-62, February 1971.

[5] W. H. Chen and C. H. Smith, "Adaptive coding of monochrome and color images," *IEEE Trans. Commun.*, vol. COM-25, pp. 1285-1292, November 1977.

[6] W. H. Chen and W. K. Pratt, "Scene adaptive coder," *IEEE Trans. Commun.*, vol. COM-32, pp. 225-232, March 1984.

[7] W. A. Pearlman, "Adaptive cosine transform image coding with constant block distortion," *IEEE Trans. Commun.*, vol. COM-38, pp. 698-703, May 1990.

[8] J. W. Woods and S. D. O'Neil, "Subband coding of images," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-34, pp. 1278-1288, October 1986.

[9] H. Gharavi and A. Tabatabai, "Sub-band coding of monochrome and color images," *IEEE Trans. Circuits and Systems*, vol. 35, pp. 207-214, February 1988.

[10] P. H. Westerink, D. E. Boekee, J. Biemond and J. W. Woods, "Subband coding of images using vector quantization," *IEEE Trans. Commun.*, vol. 36, pp. 713-719, June 1988.

[11] N. Tanabe and N. Farvardin, "Subband image coding using entropy-coded quantization over noisy channels," *IEEE Journal of Selected Areas in Communications,* vol. 10, pp. 926-943, June 1992.

[12] Y. H. Kim and J. W. Modestino, "Adaptive entropy coded subband coding of images," *IEEE Trans. Image Processing*, vol. 1, pp. 31-48, January 1992.

[13] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, vol. 1, pp. 205-220, April 1992.

[14] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84-95, January 1980.

[15] N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: a review," *IEEE Trans. Commun.*, vol. COM-36, pp. 957-971, August 1988.

[16] N. S. Jayant and P. Noll, *Digital coding of waveforms, principles and applications to speech and video*, Englewood Cliff, NJ: Prentice-Hall, 1984.

[17] A. Gersho and R. M. Gray, *Vector quantization and signal compression*, Kluwer Academic Publishers, Boston, 1992.

[18] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second-generation image-coding techniques," *Proc. IEEE,* vol. 73, pp. 549-574, April 1985.

[19] M. Kunt, M. Benard, and R. Leonardi, "Recent results in high-compression image coding," *IEEE Trans. Circuit and Systems*, vol. CAS-34, pp. 1306-1336, November 1987.

[20] H. Helmholtz, *Treatise on physiological optics*, edited by J. Southall, vol. III, *The perceptions of vision*, the Optical Society of America, George Bonta Publishing Company, Menasha, Wisconsin, 1925.

[21] T. Cornsweet, *Visual perception*, Academic Press, New York and London, 1970.

[22] S. Carlsson, "Sketch based coding of grey level images," *Signal Processing*, vol. 15, pp. 57–83, July 1988.

[23] S. Carlsson, C. Reillo, and L. H. Zetterberg, "Sketch based representation of grey value and motion information," in *From Pixels to Features* by J. C. Simon (ed.) Elsevier Science Publishers B.V. (North-Holland), 1989.

[24] W. Grimson, "Surface consistency constraints in vision," *Computer Vision, Graphics, and Image Processing* 24, pp. 28-51, 1983.

[25] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Roy. Soc. London*, Ser. B, vol. 207, pp. 187-217, 1980.

[26] D. G. Luenberger, *Linear and nonlinear programming*, Addison-Wesley Publishing Company, Menlo Park, California, etc., 1984.

[27] W. Hackbusch, *Multi-grid methods and applications*, Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1985.

[28] P. Chou and N. Pagano, *Elasticity, tensor, dyadic, and engineering approaches*, D. Van Nostrand Company, Inc., Princeton, New Jersdy, Toronto, London, 1967.

[29] T. S. Huang, "Coding of two-tone images," *IEEE Trans. Commun.*, vol. COM-25, pp. 1406-1424, November 1977.

[30] D. N. Graham, "Image transmission by two-dimensional contour coding," *Proceedings of IEEE*, Vol. 55, pp. 336-346, March, 1967.

[31] H. Freeman, "On the encoding of arbitrary geometric configuration," *IRE Trans. Electron. Comput.*, EC-10, pp. 260-268, June 1961.

[32] J. K. Yan and D. J. Sakrison, "Encoding of images based on a two-component source model," *IEEE Trans. Commun.*, vol. COM-25, pp. 1315-1322, November 1977.

[33] V. S. Nalwa and T. O. Binford, "On detecting edges," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-8, pp. 699–714, November 1986.

[34] F. W. Campbell and J. G. Robson, "Application of Fourier analysis to the visibility of gratings," *J. Physiol.*, 197, pp. 551–566, 1968.

[35] J. L. Mannos and D. J. Sakrison, "The effects of a visual fidelity criterion on the encoding of images," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 525-536, July 1974.

[36] D. L. Neuhoff and K. G. Castor, "A rate and distortion analysis of chain codes for line drawings," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 53-67, January 1985.

[37] J. J. Rissanen, "Generalized Kraft inequality and arithmetic coding," *IBM J. Res. Develop.*, pp. 198-203, May 1976.

[38] I. H. Witten, R. M. Neal and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM,* vol. 30, pp. 520-540, June 1987.

[39] O. R. Mitchell and A. Tabatabai, "Adaptive transform image coding for human analysis," *Proc. ICC,* pp. 23.2.1-23.2.5, 1979.

[40] R. C. Reininger and J. D. Gibson, "Distributions of the two-dimensional DCT coefficients for images," *IEEE Trans. Commun.*, vol. COM-31, pp. 835-839, June 1983.

[41] N. Farvardin and J. W. Modestino, "Optimum quantizer performance for a class of non-Gaussian memoryless sources," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 485-497, May 1984.

[42] A. V. Trushkin, "Optimal bit allocation algorithm for quantizing a random vector," *Problems of Inform. Transmission,* vol. 17, pp. 156-161, 1981.

[43] H. Gish and J. N. Pierce, "Asymptotically efficient quantizing," *IEEE Trans. Inform. Theory,* vol. IT-14, pp. 676-683, September 1968.

[44] T. Berger, *Rate distortion theory*, Englewood Cliff, NJ: Prentice-Hall, 1971.

[45] M. H. Woehrmann, A. N. Hessenflow and D. Kettmann, *Image alchemy*^{tm}, *user's manual*, Version 1.4, Handmade Software, Inc., March 18, 1991.

[46] J. M. Shapiro,"An embedded wavelet hierarchical image coder," *Proc. IEEE ICASSP*, pp. IV.657–IV.660, March 1992.

[47] F. Kessentini, M. J. T. Smith and C. F. Barnes, "Image coding with variable rate RVQ," *Proc. IEEE ICASSP*, pp. III.369-III.372, March 1992.

[48] M. G. Perkins and T. Lookabaugh, "A psychophysically justified bit allocation algorithm for subband image coding systems," *Proc. IEEE ICASSP*, pp. 1815-1818, May 1989.

[49] M.K.H. Fan, A. L. Tits, J. Zhou, L.-S. Wang and J. Koninckx, "CONSOLE, User's Manual, version 1.1" *Technical Research Report*, TR 87-212r2, SRC, University of Maryland, College Park, MD, June 1990.

[50] T. R. Fischer and M. Wang, "Entropy-constrained trellis-coded quantization," *IEEE Trans. Inform. Theory,* vol. IT-38, pp. 415-426, March 1992.

[51] R. Laroia and N. Farvardin, "Trellis-based scalar-vector quantizer for memoryless sources," submitted to *IEEE Trans. Inform. Theory* for publication, July 1992.

[52] H. Asada and M. Brady, "The curvature primal sketch," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-8, pp. 2–14, January 1986.

[53] A. L. Yuille and T. A. Poggio, "Scaling theorems for zero crossings," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-8, pp. 15-25, January 1986.

[54] J. W. Mckee and J. K. Aggarwal, "Computer recognition of partial views of curved objects," *IEEE Trans. Computers*, vol. C-26, pp. 790-800, August 1977.

[55] T. Pavlidis, "Curve fitting as a pattern recognition problem," in *Proc. 6th Int. Conf. Pattern Recognition*, pp. 853-859, 1982.

[56] W. A. Perkins, "A model-based vision system for industrial parts," *IEEE Trans. Computers*, vol. C-27, pp. 126-143, February 1978.

[57] N. Ayache and O. D. Fangeras, "HYPER, a new approach for the recognition and positioning of two-dimensional objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-8, pp. 44-54, January 1986.

[58] F. Attneave, "Some informational aspects of visual perception," *Psychol. Rev.,* vol. 61, pp. 183-193, 1954.

[59] I. D. Faux and M. J. Pratt, *Computational geometry for design and manufacture*, Chichester, England: Ellis Horwood, 1979.

[60] M. Brady, "Criteria for representations of shape," in *Human and Machine Vision*, I. Beck, B. Hope and A. Rosenfeld, Eds., New York: Academic, 1983.

[61] M. Brady and H. Asada, "Smoothed local symmetries and their implementation," in *Proc. 1st Int. Symp. Robotics Res.*, M. Brady and R. P. Paul, Eds., Cambridge, MA: M.I.T. Press, 1984.

[62] F. Mokhtarian and A. Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-8, pp. 34-43, January 1986.

[63] H. R. Everett, "A second-generation autonomous sentry robot," *Robotics Age*, pp. 29-32, April 1985.

[64] H. R. Everett, "A multielement ultrasonic ranging array," *Robotics Age*, pp. 13-20, July 1985.

[65] D. M. Young, *Iterative solution of large linear systems*, Academic Press, New York and London, 1971.

[66] I. M. Gelfand and S. V. Fomin, *Calculus of variations*, translated and edited by R.A. Silverman, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1963.

[67] W. Kaplan, *Advanced calculus*, Addison-Wesley Publishing Company, Inc., MA., 1952.

[68] B. O'Neill, *Semi-Riemannian geometry, with applications to relativity*, Academic Press, New York, London, etc., 1983.

[69] O. Axelsson, *Finite element solution of boundary value problems : theory and computation*, Academic Press, Orlando, Fla., 1984.

[70] D. P. Kemp, J. S. Collura and T. E. Tremain, "Multi-frame coding of LPC parameters at 600-800 bps," *Proc. IEEE ICASSP,* pp. 609-612, May 1991.

[71] Y. Shiraki and M. Honda, "LPC speech coding based on variable-length segment quantization," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-36, pp. 1437-1444, September 1988.

[72] X. Ran and N. Farvardin, "Combined VQ-DCT coding of images using interblock noiseless coding," *Proc. IEEE ICASSP,* pp. 2281-2284, April 1990.

[73] X. Ran and N. Farvardin,"A three-component image model for coding with a subjective criterion," presented in the Seventh Workshop on Multidimensional Signal Processing, Lake Placid, NY, p. 7.2, September 1991.

[74] X. Ran and N. Farvardin, "Adaptive DCT image coding based on a three-component image model," *Proc. of IEEE ICASSP*, pp. III-201–III-204, March 1992.

[75] X. Ran and N. Farvardin, "A perceptually motivated three-component image model," submitted to *IEEE Trans. Image Processing*, March 1992. Also, SRC Technical Report TR 92-32, University of Maryland, College Park, MD, March 1992.

[76] X. Ran and N. Farvardin, "Low bit-rate image coding using a three-component image model," submitted to *IEEE Trans. Image Processing*, July 1992. Also, SRC Technical Report TR 92-75, University of Maryland, College Park, MD, July 1992.